

Lösungen zur Hausaufgabe zur Vorlesung  
Digitale Systeme

WS 2004/2005  
Prof. Dr. rer. nat. Manfred Schimmler  
Dipl.-Inform. Viktor Bunimov  
Dipl.-Ing. Gerd Pfeiffer  
Dipl.-Ing. Heinz Kreft

### Serie 1

#### Aufgabe 1

a) Konvertieren Sie die Zahl  $(368)_{10}$  in eine Dualzahl

1. Möglichkeit: Wiederholte Division durch 2 mit Rest:

$$\begin{aligned} 368 &= 184 \cdot 2 + 0 \\ 184 &= 92 \cdot 2 + 0 \\ 92 &= 46 \cdot 2 + 0 \\ 46 &= 23 \cdot 2 + 0 \\ 23 &= 11 \cdot 2 + 1 \\ 11 &= 5 \cdot 2 + 1 \\ 5 &= 2 \cdot 2 + 1 \\ 2 &= 1 \cdot 2 + 0 \\ 1 &= 0 \cdot 2 + 1 \end{aligned}$$

Es gilt also

$$\begin{aligned} (368)_{10} &= (((((((1 \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 0) \cdot 2 + 0) \cdot 2 + 0) \cdot 2 + 0 \\ &= 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 \\ &= (101110000)_2 \end{aligned}$$

Die Berechnung erfolgte hierbei im Dezimalsystem.

2. Möglichkeit: Jede Ziffer der Zahl  $(368)_{10}$  als 4-stellige Binärzahl darstellen und Hornerschema für  $B = 10$  ausrechnen:

$$\begin{aligned}
 368 &= (3 \cdot 10 + 6) \cdot 10 + 8 \\
 &= (0011_2 \cdot 1010_2 + 0110_2) \cdot 1010_2 + 1000_2 \\
 &= (11110_2 + 0110_2) \cdot 1010_2 + 1000_2 \\
 &= 10110100_2 + 1000_2 \\
 &= 101110000_2
 \end{aligned}$$

Nebenrechnungen im Dualsystem:

$$\begin{array}{r} 0011 \cdot 1010 \\ \hline \end{array}$$

0011

0000

0011

0000

0011110

101101000

1000

101110000

$$\begin{array}{r} 11110 \\ + 0110 \\ \hline \end{array}$$

100100

$$\begin{array}{r} 100100 \cdot 1010 \\ \hline \end{array}$$

100100

000000

100100

000000

10110100

Die Berechnung findet hier vollständig im Dualsystem statt, daher ist das Verfahren zur Konvertierung von Dezimal- in Dualzahlen durch Rechner mit Binärarithmetik geeignet.

b) Multiplizieren Sie die Dualzahl aus a) mit der Zahl  $(101)_2$  im Dualsystem . Es ist

$$(101110000)_2 \cdot (101)_2 = 368 \cdot 5 = 1840.$$

$$\begin{array}{r} 101110000 \cdot 101 \\ \hline \end{array}$$

101110000

000000000

101110000

11100110000

c) Konvertieren Sie das Ergebnis aus b) in eine Dezimalzahl.

1. Möglichkeit: Division durch  $(1010)_2 = (10)_{10}$  mit Rest. Die Berechnung erfolgt dabei im Dualsystem:

$$11100110000 : 1010 = 10111000$$

$$\begin{array}{r} 1010 \\ \hline 10001 \\ \underline{1010} \\ 1111 \\ \underline{1010} \\ 1010 \\ \underline{1010} \\ 0000 \end{array}$$

$$10111000 : 1010 = 10010$$

$$\begin{array}{r} 1010 \\ \hline 1100 \\ \underline{1010} \\ 100 \end{array}$$

$$10010 : 1010 = 1$$

$$\begin{array}{r} 1010 \\ \hline 1000 \end{array}$$

$$\begin{aligned} 11100110000_2 &= ((1_2 \cdot 1010_2 + 1000_2) \cdot 1010_2 + 100_2) \cdot 1010_2 + 0_2 \\ &= ((1 \cdot 10 + 8) \cdot 10 + 4) \cdot 10 + 0 \\ &= 1840 \end{aligned}$$

2. Möglichkeit: Berechnung im Dezimalsystem mit Hornerschema

$$\begin{aligned} (11100110000)_2 &= 0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 0 \cdot 2^7 + 1 \cdot 2^8 + 1 \cdot 2^9 + 1 \cdot 2^{10} \\ &= 16 + 32 + 256 + 512 + 1024 \\ &= 1840 \end{aligned}$$

## Aufgabe 2

Multiplizieren Sie die Zahl  $(321)_4$  mit der Zahl  $(213)_4$  im 4-adischen Zahlensystem.

$$321 \cdot 213$$

$$\begin{array}{r} 1302 \\ 321 \\ 2223 \\ \hline 111 \\ \hline 202233 \end{array}$$

### Aufgabe 3

Führen Sie die Subtraktion  $(2354)_{10} - (1234)_{10}$  im Dualsystem aus. Benutzen Sie dazu das 2-Komplement.

Zunächst die Zahlenkonvertierung in das Dualsystem:

$$(2354)_{10} = ((10_2 \cdot 1010_2 + 11_2) \cdot 1010_2 + 101_2) \cdot 1010_2 + 100_2 :$$

|                             |     |   | Ergebnis im Zielsystem |
|-----------------------------|-----|---|------------------------|
|                             | 10  | = | 10                     |
| $10 \cdot 1010 +$           | 11  | = | 10111                  |
| $10\ 111 \cdot 1010 +$      | 101 | = | 11 101 011             |
| $11\ 101\ 011 \cdot 1010 +$ | 100 | = | 100 100 110 010        |

$$(1234)_{10} = ((1010_2 + 10_2) \cdot 1010_2 + 11_2) \cdot 1010_2 + 100_2 :$$

|                            |     |   | Ergebnis im Zielsystem |
|----------------------------|-----|---|------------------------|
|                            | 1   | = | 1                      |
| $1 \cdot 1010 +$           | 10  | = | 1100                   |
| $1100 \cdot 1010 +$        | 11  | = | 1 111 011              |
| $1\ 111\ 011 \cdot 1010 +$ | 100 | = | 10 011 010 010         |

2-Komplement:

| Dezimalzahl                    | Schutzstelle | Vorzeichen | Zahlenwert      |
|--------------------------------|--------------|------------|-----------------|
| $A = 2354_{10}$                |              | 0          | 100 100 110 010 |
| $B = 1234_{10}$                |              | 0          | 010 011 010 010 |
| Komplement zu $B = -1234_{10}$ |              | 1          | 101 100 101 110 |
| Ergebnis                       |              | 0          | 010 001 100 000 |

#### Aufgabe 4

Zeigen Sie: In jedem B-adischen Zahlensystem mit  $B \geq 4$  ist  $(1331)_B$  eine Kubikzahl.

Es gilt:

$$\begin{aligned}(1331)_B &= 1 \cdot B^3 + 3 \cdot B^2 + 3 \cdot B^1 + 1 \cdot B^0 \\ &= (B+1)^3 \\ &= ((11)_B)^3.\end{aligned}$$

Lösungen zur Hausaufgabe zur Vorlesung  
Digitale Systeme

WS 2004/2005  
Prof. Dr. rer. nat. Manfred Schimpler  
Dipl.-Inform. Viktor Bunimov  
Dipl.-Ing. Gerd Pfeiffer  
Dipl.-Ing. Heinz Kreft

## Serie 2

### Aufgabe 1

Konvertieren Sie die Zahl  $(13303790)_{10}$  in eine Dual-, Oktal- und Hexadezimalzahl.

$13303790 = 11001010111111111101110$

6651895

3325947

1662973

831486

415743

207871

103935

51967

25983

12991

6495

3247

1623

811

405

202

101

50

25

12

6

3

1

0

Dual  $\rightarrow$  Oktalzahl:

Algorithmus: Wir unterteilen die Dualzahl in drei Bit Blöcke von rechts nach links. Jeder Block entspricht einer Oktalziffer, da  $2^3=8$ .

|            |     |     |     |     |     |     |     |     |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Dualzahl:  | 110 | 010 | 101 | 111 | 111 | 111 | 101 | 110 |
| Oktalzahl: | 6   | 2   | 5   | 7   | 7   | 7   | 5   | 6   |

Dual  $\rightarrow$  Hexadezimalzahl:

Algorithmus: Wir unterteilen die Dualzahl in vier Bit Blöcke von rechts nach links. Jeder Block entspricht einer Hexadezimalzahl, da  $2^4=16$ .

|              |      |      |      |      |      |      |
|--------------|------|------|------|------|------|------|
| Dualzahl:    | 1100 | 1010 | 1111 | 1111 | 1110 | 1110 |
| Hexadezimal: | C    | A    | F    | F    | E    | E    |

## Aufgabe 2

Konvertieren Sie die Zahl 468,2 in die Festkommadarstellung zur Basis 2 mit 9 Vorkomma- und 6 Nachkommastellen.

$$468,2 = (468)_{10} + (0,2)_{10} = \sum_{i=-6}^9 b_i \cdot 2^i$$

$$0,2 = \sum_{i=-6}^{-1} b_i \cdot 2^i$$

$$= ((((((2^{-1}b_{-6} + b_{-5})2^{-1} + b_{-4})2^{-1} + b_{-3})2^{-1} + b_{-2})2^{-1} + b_{-1})2^{-1}$$

Berechnung der  $b_i$  für  $-6 \leq i \leq -1$ :

$$\begin{aligned} 0,2 \cdot 2 &= 0,4 \Rightarrow b_{-1} = 0 \\ 0,4 \cdot 2 &= 0,8 \Rightarrow b_{-2} = 0 \\ 0,8 \cdot 2 &= 1,6 \Rightarrow b_{-3} = 1 \\ 0,6 \cdot 2 &= 1,2 \Rightarrow b_{-4} = 1 \\ 0,2 \cdot 2 &= 0,4 \Rightarrow b_{-5} = 0 \\ 0,4 \cdot 2 &= 0,8 \Rightarrow b_{-6} = 0 \\ 0,8 \cdot 2 &= 1,6 \Rightarrow b_{-7} = 1 \end{aligned}$$

Berechnung der  $b_i$  für  $0 \leq i \leq 9$ :

$$\begin{array}{r} 468:16 = 29 \text{ Rest } 4 \\ \underline{32} \\ 148 \\ \underline{144} \end{array}$$

$$\begin{array}{r} 29:16 = 1 \text{ Rest } 13 \\ \underline{16} \\ 13 \end{array}$$

4

$$\Rightarrow 468 = (1 \cdot 16 + 13) \cdot 16 + 4$$

$$(468)_{10} = (1D4)_{16}$$

$$= (111010100)_2$$

Damit ergibt sich nach Rundung ( $b_{-6}$  wird auf 1 gesetzt, da die siebte Nachkommastelle eine 1 ist) das Ergebnis

$$(b_9 b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0, b_{-1} b_{-2} b_{-3} b_{-4} b_{-5} b_{-6})_2 = (111010100, 001101)_2$$

### Aufgabe 3

a) Bestimmen Sie die Gleitkommastellung der Zahlen 10 und 0,3 im IEEE 32-Bit-Format.

$$10 = (1010)_2$$

$$= (-1)^0 \cdot (1,010)_2 \cdot 2^3$$

$$= (-1)^0 \cdot (1,010\,000\,000\,000\,000\,000\,000\,00)_2 \cdot 2^{130-127}$$

Die Gleitkommadarstellung der Zahl 10 im IEEE 32-Bit-Format besteht also aus dem Vorzeichenbit  $V = 0$ , der Mantisse  $M = 0100000000000000000000$  und dem Exponenten  $E = (130)_{10} = 10\,000\,010$ .

Für die Zahl 0,3 gilt

$$0,3 \cdot 2 = 0,6 \Rightarrow b_{-1} = 0$$

$$0,6 \cdot 2 = 1,2 \Rightarrow b_{-2} = 1$$

$$0,2 \cdot 2 = 0,4 \Rightarrow b_{-3} = 0$$

$$0,4 \cdot 2 = 0,8 \Rightarrow b_{-4} = 0$$

$$0,8 \cdot 2 = 1,6 \Rightarrow b_{-5} = 1$$

$$0,6 \cdot 2 = 1,2 \Rightarrow b_{-6} = 1$$

$$0,2 \cdot 2 = 0,4 \Rightarrow b_{-7} = 0$$

$$0,4 \cdot 2 = 0,8 \Rightarrow b_{-8} = 0$$

$$0,8 \cdot 2 = 1,6 \Rightarrow b_{-9} = 1$$

$$\Rightarrow 0,3 = (0,01\overline{001})_2$$

Nach Rundung auf 23 Nachkommastellen ist

$$0,3 = (1,00110011001100110011010)_2 \cdot 2^{-2}.$$

Die Gleitkommadarstellung der Zahl 0,3 im IEEE 32-Bit-Format besteht also aus dem Vorzeichen  $V = 0$ , der Mantisse  $M = 0011001100110011010$  und dem Exponenten  $E = 125 = 01111101$ .

b) addieren Sie die beiden Gleitkommazahlen.

$$(1,010\,000\,000\,000\,000\,000\,000\,00)_2 \cdot 2^3 + (1,00110011001100110011010)_2 \cdot 2^{-2}.$$

Exponentendifferenz:  $3 - (-2) = 5$

Die zweite Binärzahl muß also um 5 Stellen nach rechts geschoben werden, wenn der Exponent der Zweierpotenz von  $-2$  auf  $3$  geändert wird.

$$\begin{array}{r} 1,010\ 000\ 000\ 000\ 000\ 000\ 000\ 00 \\ +\ 0,000\ 010\ 011\ 001\ 100\ 110\ 011\ 01 \\ \hline 1,010\ 010\ 011\ 001\ 100\ 110\ 011\ 01 \end{array}$$

Das Ergebnis der Addition lautet hiermit

$$(1,01001001100110011001101)_2 \cdot 2^3.$$

c) Multiplizieren Sie die beiden Gleitkommazahlen. Wir multiplizieren

$(1,00110011001100110011010)_2 \cdot 2^{-2}$  mit  $(1,010\ 000\ 000\ 000\ 000\ 000\ 000\ 00)_2 \cdot 2^3$ . Es werden die Mantissen ergänzt um die führenden Einsen miteinander multipliziert. Die Multiplikation von  $2^{-2}$  mit  $2^3$  wird durch Addition der Exponenten ausgeführt:  $2^{-2} \cdot 2^3 = 2^{-2+3} = 2^1$ .

Die Multiplikation der 24-Bit Dualzahlen ergibt

$$\begin{array}{r} 1,00110011001\ 1001\ 10011010 \cdot \quad 1,010000000000000000000000 \\ \hline 100110011001100110011010 \\ 000000000000000000000000 \\ 100110011001100110011010 \\ 000000000000000000000000 \\ \quad \cdot \\ \quad \cdot \\ \quad \cdot \\ \hline 1,100000000000000000000010 \quad \dots 0 \end{array}$$

Als Ergebnis erhält man also

$$(1,1000000000000000000000010)_2 \cdot 2^1 \approx 3.$$

#### Aufgabe 4

Bestimmen Sie die kleinste und die größte Gleitkommazahl im IEEE 32-Bit-Format.

Die kleinste Gleitkommazahl  $w_k$  im IEEE 32-Bit-Format ergibt sich wie folgt:

$$\begin{aligned} w_k &= (-1)^0 \cdot (0, \underbrace{0\dots 0,1}_{23\text{ Bits}})_2 \cdot 2^{-126} \\ &= 2^{-23} \cdot 2^{-126} \\ &= 2^{-149} \end{aligned}$$

Die größte positive Gleitkommazahl  $w_g$  im IEEE 32-Bit-Format ergibt sich wie folgt:

$$\begin{aligned}w_g &= (-1)^0 \cdot (1, \underbrace{1 \dots 1}_{23 \text{ Bits}})_2 2^{254-127} \\&= 1 \cdot (1, 1 \dots 1)_2 \cdot 2^{127} \\&= 1 \cdot (\underbrace{1 \dots 1}_{24 \text{ Bits}})_2 \cdot 2^{127-23} \\&= (2^{24} - 1) \cdot 2^{104} \\&2^{128} - 2^{104}\end{aligned}$$

Lösungen zur Hausaufgabe zur Vorlesung  
Digitale Systeme

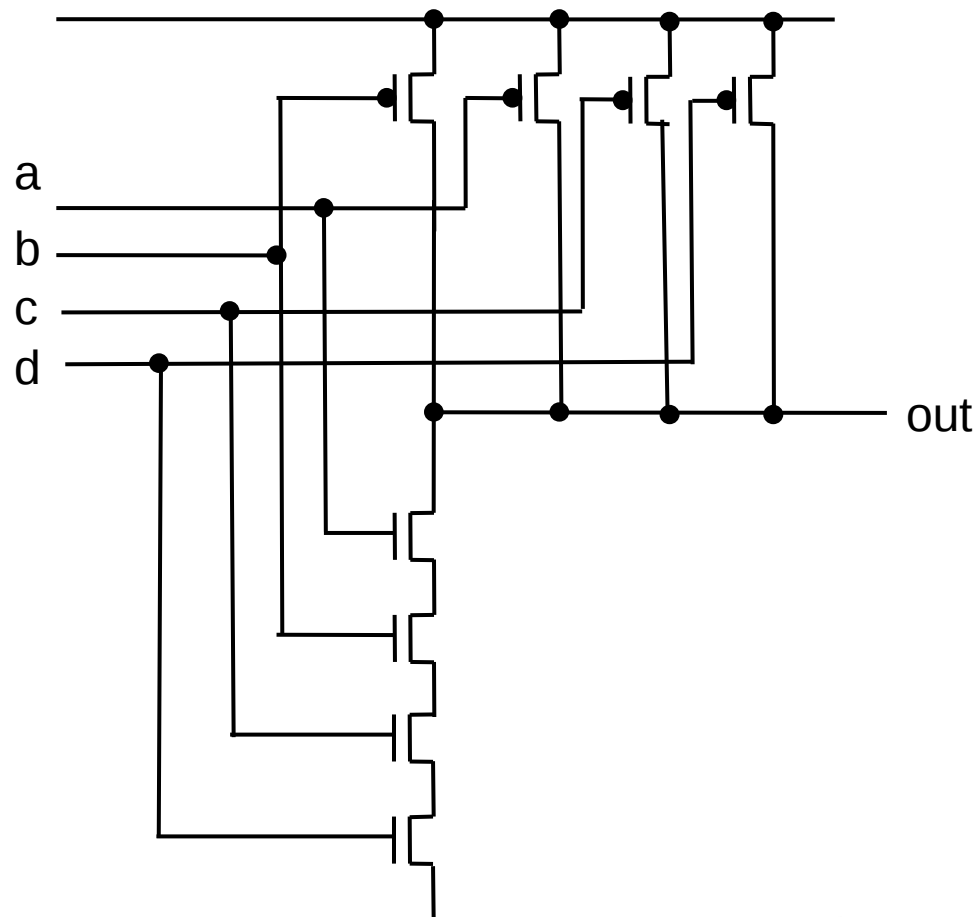
WS 2004/2005  
Prof. Dr. rer. nat. Manfred Schimmeler  
Dipl.-Inform. Viktor Bunimov  
Dipl.-Ing. Gerd Pfeiffer  
Dipl.-Ing. Heinz Kreft

**Serie 3**

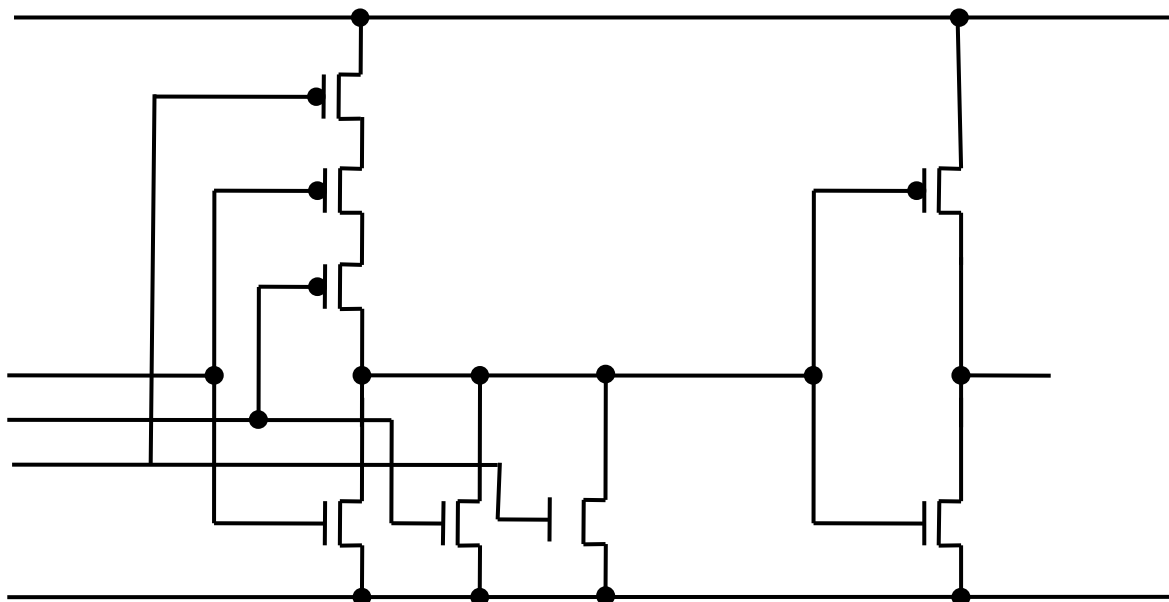
**Aufgabe 1**

Konstruieren Sie in CMOS ein

- a) NAND-Gatter mit vier Eingängen

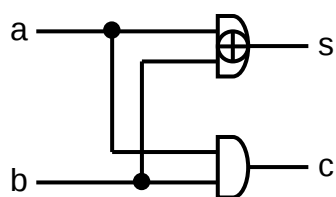


b) OR-Gatter mit drei Eingängen

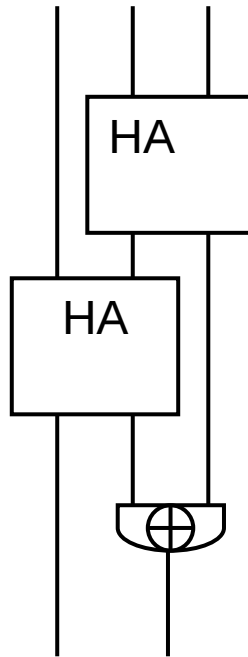


## Aufgabe 2

a) Bauen Sie einen Halbaddierer aus den Boole'schen Elementarfunktionen mit zwei Eingängen aus der Vorlesung.



b) Bauen Sie einen Volladdierer aus Halbaddierern und den Boole'schen Elementarfunktionen aus der Vorlesung.



### Aufgabe 3

a) Bestimmen Sie die Kanonische disjunktive Normalform (KDNF) von

$$f_1 = (x_1 + x_2 + \overline{x_3}) \cdot (x_1 x_2 + \overline{x_1} x_3)$$

| $x_1$ | $x_2$ | $x_3$ | $\overline{x_1}$ | $\overline{x_3}$ | $x_1 x_2$ | $\overline{x_1} x_3$ | $x_1 + x_2 + \overline{x_3}$ | $x_1 x_2 + \overline{x_1} x_3$ | $f_1$ |
|-------|-------|-------|------------------|------------------|-----------|----------------------|------------------------------|--------------------------------|-------|
| 0     | 0     | 0     | 1                | 1                | 0         | 0                    | 1                            | 0                              | 0     |
| 0     | 0     | 1     | 1                | 0                | 0         | 1                    | 0                            | 1                              | 0     |
| 0     | 1     | 0     | 1                | 1                | 0         | 0                    | 1                            | 0                              | 0     |
| 0     | 1     | 1     | 1                | 0                | 0         | 1                    | 1                            | 1                              | 1     |
| 1     | 0     | 0     | 0                | 1                | 0         | 0                    | 1                            | 0                              | 0     |
| 1     | 0     | 1     | 0                | 0                | 0         | 0                    | 1                            | 0                              | 0     |
| 1     | 1     | 0     | 0                | 1                | 1         | 0                    | 1                            | 1                              | 1     |
| 1     | 1     | 1     | 0                | 0                | 1         | 0                    | 1                            | 1                              | 1     |

$$f_1 = \overline{x_1} x_2 x_3 + x_1 x_2 \overline{x_3} + x_1 x_2 x_3$$

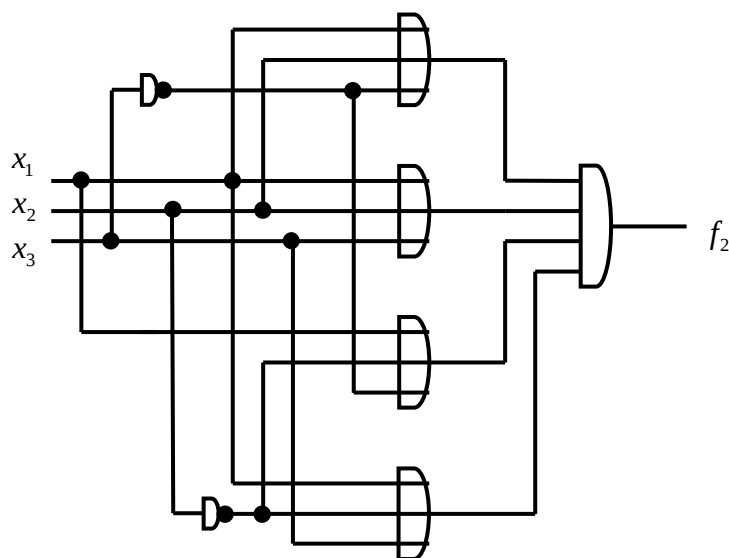
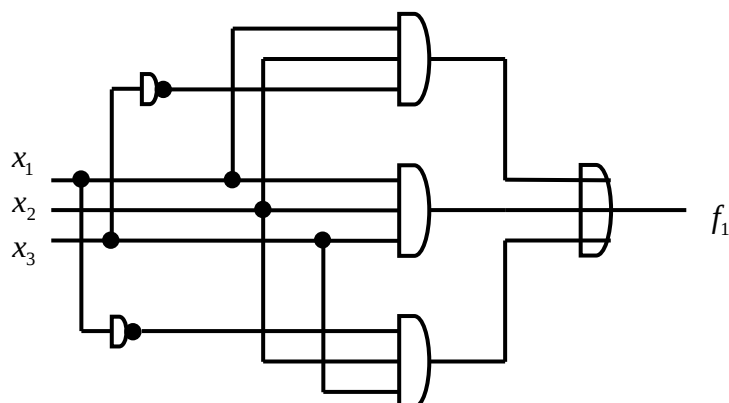
b) Bestimmen Sie die Kanonische konjunktive Normalform (KKNF) von

$$f_2 = x_1 \bar{x}_2 x_3 + x_1 (x_2 + \bar{x}_3)$$

| $x_1$ | $x_2$ | $x_3$ | $\bar{x}_2$ | $\bar{x}_3$ | $x_1 \bar{x}_2 x_3$ | $x_2 + \bar{x}_3$ | $x_1 (x_2 + \bar{x}_3)$ | $f_1$ |
|-------|-------|-------|-------------|-------------|---------------------|-------------------|-------------------------|-------|
| 0     | 0     | 0     | 1           | 1           | 0                   | 1                 | 0                       | 0     |
| 0     | 0     | 1     | 1           | 0           | 0                   | 0                 | 0                       | 0     |
| 0     | 1     | 0     | 0           | 1           | 0                   | 1                 | 0                       | 0     |
| 0     | 1     | 1     | 0           | 0           | 0                   | 1                 | 0                       | 0     |
| 1     | 0     | 0     | 1           | 1           | 0                   | 1                 | 1                       | 1     |
| 1     | 0     | 1     | 1           | 0           | 1                   | 0                 | 0                       | 1     |
| 1     | 1     | 0     | 0           | 1           | 0                   | 1                 | 1                       | 1     |
| 1     | 1     | 1     | 0           | 0           | 0                   | 1                 | 1                       | 1     |

$$f_2 = (x_1 + x_2 + x_3)(x_1 + x_2 + \bar{x}_3)(x_1 + \bar{x}_2 + x_3)(x_1 + \bar{x}_2 + \bar{x}_3)$$

c) Stellen Sie  $f_1$  und  $f_2$  als Schaltnetz dar.



**Serie 4**

**Aufgabe 1**

Beweisen Sie:

c) Für jede Eingabevariable  $x_0, x_1$  gilt:  $(x_0 + x_1)(\overline{x_0 + x_1}) = x_0$

$$(x_0 + x_1)(\overline{x_0 + x_1}) \stackrel{\text{Kommutativgesetz}}{=} (\overline{x_0 + x_1})(x_0 + x_1) =$$

$$\stackrel{\text{Distributivgesetz}}{=} x_0(x_0 + x_1) + \overline{x_1}(x_0 + x_1) \stackrel{\text{Kommutativgesetz}}{=} (x_0 + x_1)x_0 + (x_0 + x_1)\overline{x_1} =$$

$$\stackrel{\text{Distributivgesetz}}{=} x_0x_0 + x_1x_0 + x_0\overline{x_1} + x_1\overline{x_1} \stackrel{\text{Satz 4 + Inv. Element}}{=} x_0 + x_1x_0 + x_0\overline{x_1} + 0 =$$

$$\stackrel{\text{Kommutativgesetz}}{=} x_0 + x_1x_0 + \overline{x_1}x_0 + 0 \stackrel{\text{Distributivgesetz}}{=} x_0 + (x_1 + \overline{x_1}) \cdot x_0 + 0 =$$

$$\stackrel{\text{Kommutativgesetz}}{=} x_0 + 0 + (x_1 + \overline{x_1}) \cdot x_0 \stackrel{\text{Neutrales Element}}{=} x_0 + 0 + (x_1 + \overline{x_1}) \cdot x_0 =$$

$$\stackrel{\text{Inverses Element}}{=} x_0 + 1 \cdot x_0 \stackrel{\text{Neutrales Element}}{=} x_0 + x_0 \stackrel{\text{Satz 4}}{=} x_0$$

d) Für jede Eingabevariable  $x_0, x_1$  gilt:  $\overline{x_0} + \overline{x_1} = \overline{x_0 \cdot x_1}$  (Das Gesetz von De Morgan)

Beweis: Die Inverse von  $\overline{x_0x_1}$  ist  $x_0x_1$  ( $(x_0x_1) \cdot \overline{x_0x_1} = 0$  und  $(x_0x_1) + \overline{x_0x_1} = 1$ )

Wir beweisen, dass  $x_0x_1$  die Inverse von  $\overline{x_0} + \overline{x_1}$  ist. Um das zu beweisen, müssen wir zeigen, dass  $(\overline{x_0} + \overline{x_1}) \cdot x_0x_1 = 0$  und  $(\overline{x_0} + \overline{x_1}) + x_0x_1 = 1$

$$\begin{array}{ccccc} & \text{Distributivgesetz} & & \text{Kommutativgesetz} & \text{Satz 2} \\ (\overline{x_0} + \overline{x_1}) \cdot x_0 x_1 & = & \overline{x_0}(x_0 x_1) + \overline{x_1}(x_0 x_1) & = & \overline{x_0}(x_0 x_1) + \overline{x_1}(x_1 x_0) = \end{array}$$

$$\begin{array}{ccccc} & \text{Inverses Element} & & \text{Kommutativgesetz} & \\ = (\overline{x_0} x_0) x_1 + (\overline{x_1} x_1) x_0 & = & 0 \cdot x_1 + 0 \cdot x_0 & = & x_1 \cdot 0 + x_0 \cdot 0 = \end{array}$$

$$\begin{array}{ccccc} \text{Satz 1} & \text{Neutrales Element} & & & \\ = 0 + 0 & = & 0 & & \end{array}$$

$$\begin{array}{ccccc} & \text{Assoziativgesetz} & & \text{Kommutativgesetz} & \\ (\overline{x_0} + \overline{x_1}) + x_0 x_1 & = & \overline{x_0} + (\overline{x_1} + x_0 x_1) & = & \overline{x_0} + (x_0 x_1 + \overline{x_1}) = \end{array}$$

$$\begin{array}{ccccc} \text{Assoziativgesetz} & & \text{Kommutativgesetz} & & \\ = & (\overline{x_0} + x_0 x_1) + \overline{x_1} & = & (x_0 x_1 + \overline{x_0}) + \overline{x_1} = \end{array}$$

$$\begin{array}{ccccc} \text{Kommutativgesetz} & & \text{Distributivgesetz} & & \\ = & (x_0 x_1 + \overline{x_0}) + \overline{x_1} & = & (x_0 + \overline{x_0}) \cdot (x_1 + \overline{x_0}) + \overline{x_1} = \end{array}$$

$$\begin{array}{ccccc} \text{Inverses Element} & & \text{Neutrales Element} & & \\ = & 1 \cdot (x_1 + \overline{x_0}) + \overline{x_1} & = & (x_1 + \overline{x_0}) + \overline{x_1} = \end{array}$$

$$\begin{array}{ccccc} \text{Assoziativgesetz} & & \text{Kommutativgesetz} & & \text{Satz 2} \\ = & x_1 + (\overline{x_0} + \overline{x_1}) & = & x_1 + (\overline{x_1} + \overline{x_0}) & = (x_1 + \overline{x_1}) + \overline{x_0} = \end{array}$$

$$\begin{array}{ccccc} \text{Inverses Element} & & \text{Kommutativgesetz} & & \text{Satz 1} \\ = & 1 + \overline{x_0} & = & \overline{x_0} + 1 & = 1 \end{array}$$

## Aufgabe 2

Minimieren Sie jeweils die DNF und die KNF der durch die folgenden Karnaughdiagramme gegebenen Booleschen Funktionen.

|   |   |   |   |   |   |
|---|---|---|---|---|---|
|   | b |   |   |   |   |
| a | 1 | 0 | 0 | 0 | c |
|   | 0 | 1 | 0 | 0 |   |
|   | 1 | 1 | 1 | 1 |   |
|   | 1 | 0 | 0 | 0 |   |
|   | d |   |   |   |   |

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
|   | b |   |   |   |   |   |   |
| a | { | 1 | 0 | 0 | 1 | } | c |
|   |   | 0 | 1 | 1 | 0 |   |   |
|   |   | 0 | 1 | 1 | 0 |   |   |
|   |   | 1 | 0 | 0 | 1 |   |   |
|   |   | d |   |   |   |   |   |

|   |   |   |   |   |   |
|---|---|---|---|---|---|
|   | b |   |   |   |   |
| a | 1 | 1 | 0 | 1 | c |
|   | 1 | 0 | 0 | 1 |   |
|   | 1 | 0 | 0 | 1 |   |
|   | 1 | 1 | 0 | 1 |   |
|   | d |   |   |   |   |

|     |   |   |   |   |
|-----|---|---|---|---|
| b   |   |   |   |   |
| a { | 0 | 1 | 0 | 1 |
|     | 0 | 1 | 0 | 0 |
|     | 0 | 0 | 1 | 0 |
|     | 1 | 0 | 1 | 0 |
| d   |   |   |   |   |

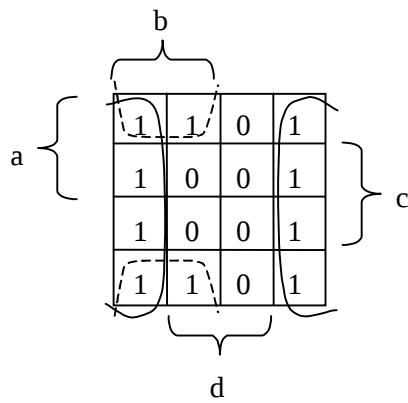
Minimierung der DNF:

|     |   |   |   |   |
|-----|---|---|---|---|
| b   |   |   |   |   |
| a { | 1 | 0 | 0 | 0 |
|     | 0 | 1 | 0 | 0 |
|     | 1 | 1 | 1 | 1 |
|     | 1 | 0 | 0 | 0 |
| d   |   |   |   |   |

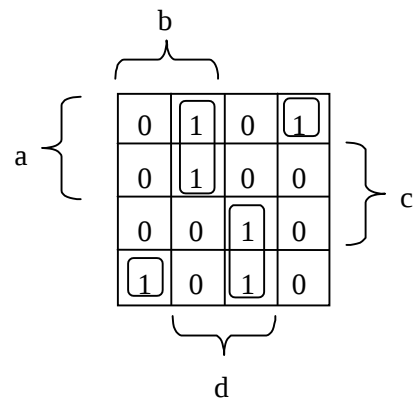
$$b\bar{c}\bar{d} + bcd + \bar{a}c$$

|     |   |   |   |   |
|-----|---|---|---|---|
| b   |   |   |   |   |
| a { | 1 | 0 | 0 | 1 |
|     | 0 | 1 | 1 | 0 |
|     | 0 | 1 | 1 | 0 |
|     | 1 | 0 | 0 | 1 |
| d   |   |   |   |   |

$$\bar{c}\bar{d} + cd$$

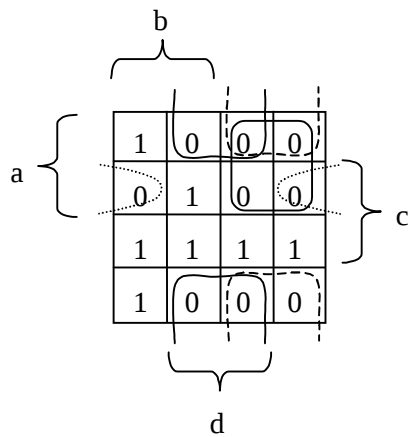


$$b\bar{c} + \bar{d}$$

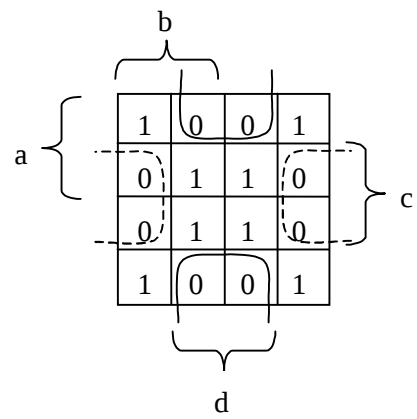


$$\bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + abd + \bar{a}\bar{b}d$$

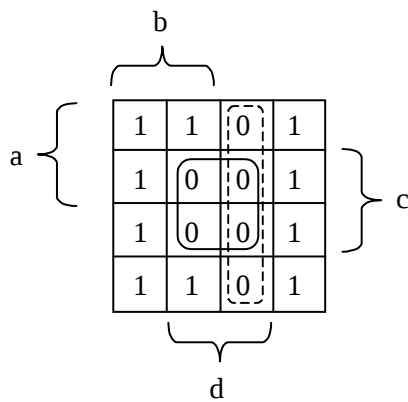
Minimierung der KNF:



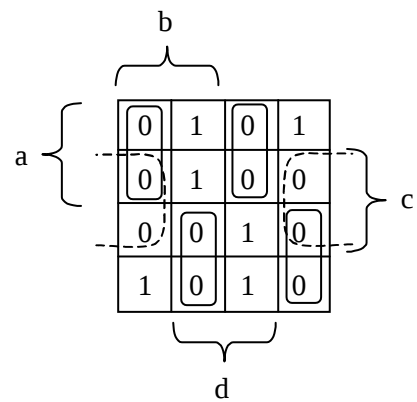
$$(\bar{a} + \bar{c} + d) \cdot (c + \bar{d}) \cdot (b + c) \cdot (\bar{a} + b)$$



$$(\bar{c} + d) \cdot (c + \bar{d})$$



$$(\bar{c} + \bar{d}) \cdot (b + \bar{d})$$



$$(\bar{a} + \bar{b} + d) \cdot (a + \bar{b} + \bar{d}) \cdot (\bar{a} + b + \bar{d}) \cdot (a + b + d) \cdot (\bar{c} + d)$$

### Aufgabe 3

Minimieren Sie die disjunktive Normalform der durch das Karnaughdiagramm aus Bild 1 gegebenen Booleschen Funktion  $h$ .

|       |  |       |   |       |   |   |   |   |   |       |       |
|-------|--|-------|---|-------|---|---|---|---|---|-------|-------|
|       |  |       |   | $x_1$ |   |   |   |   |   |       |       |
|       |  |       |   | $x_0$ |   |   |   |   |   |       |       |
| $x_3$ |  | $x_2$ | 1 | 1     | 1 | 1 | 1 | 1 | 1 | 1     | $x_5$ |
|       |  |       | 0 | 0     | 1 | 0 | 0 | 1 | 1 | 1     |       |
|       |  |       | 0 | 0     | 1 | 0 | 0 | 1 | 0 | 0     |       |
|       |  |       | 0 | 1     | 1 | 0 | 0 | 1 | 1 | 0     |       |
|       |  | 0     | 1 | 1     | 0 | 0 | 1 | 1 | 0 | $x_5$ |       |
|       |  | 0     | 0 | 1     | 0 | 0 | 1 | 0 | 0 |       |       |
|       |  | 0     | 0 | 1     | 0 | 0 | 1 | 1 | 1 |       |       |
|       |  | 0     | 0 | 0     | 1 | 1 | 1 | 1 | 1 |       |       |
|       |  |       |   | $x_4$ |   |   |   |   |   |       |       |
|       |  |       |   | $x_4$ |   |   |   |   |   |       |       |

Bild 1: Karnaughdiagramm von  $h$ .

$x_1$

$x_0$

$x_2$

$x_3$

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

$x_5$

$x_5$

$x_4$

$x_4$

|  |  |  |  |  |  |  |  |       |  |
|--|--|--|--|--|--|--|--|-------|--|
|  |  |  |  |  |  |  |  | $x_1$ |  |
|  |  |  |  |  |  |  |  | $x_0$ |  |
|  |  |  |  |  |  |  |  | $x_2$ |  |
|  |  |  |  |  |  |  |  | $x_3$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  | $x_5$ |  |
|  |  |  |  |  |  |  |  | $x_4$ |  |
|  |  |  |  |  |  |  |  |       |  |

Lösungen zur Hausaufgabe zur Vorlesung  
Digitale Systeme

WS 2004/2005  
Prof. Dr. rer. nat. Manfred Schimmmler  
Dipl.-Inform. Viktor Bunimov  
Dipl.-Ing. Gerd Pfeiffer  
Dipl.-Ing. Heinz Kreft

Serie 5

**Aufgabe 1**

Es soll eine Schaltung entworfen werden, welche für eine ganze Zahl  $x$  mit  $0 \leq x \leq 21$  die Zahl  $f(x) = (63 - 3 \cdot x) \bmod 8$  als Ausgabe liefert. Die Eingabe von  $x$  sowie die Ausgabe von  $f(x)$  soll in Binärdarstellung erfolgen.

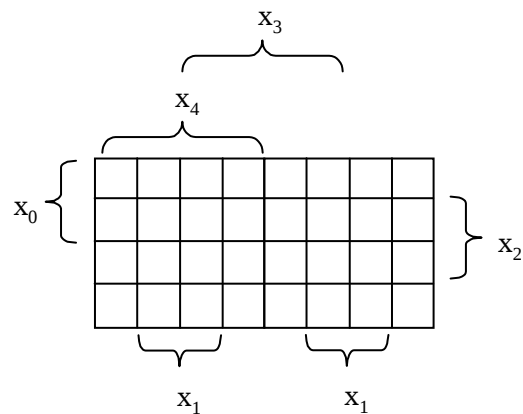
- a) Erstellen Sie die vollständige Funktionstabelle unter Berücksichtigung der beliebig wählbaren Funktionswerte.

Mit den Bezeichnungen  $x = (x_4 x_3 x_2 x_1 x_0)_2$  und  $y = f(x) = (y_2 y_1 y_0)_2$  ergibt sich die folgende Wertetabelle:

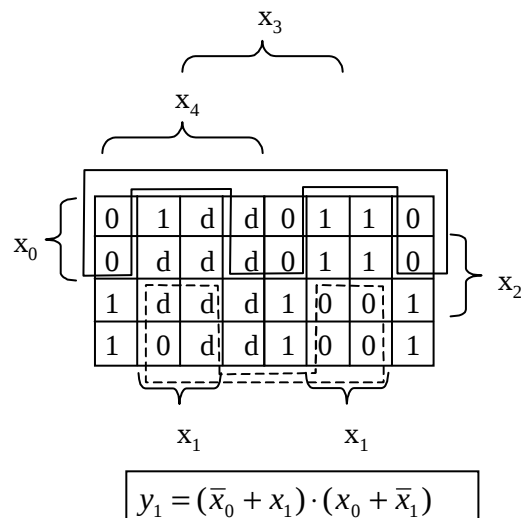
| $x_4$ | $x_3$ | $x_2$ | $x_1$ | $x_0$ | $63-3x$ | $y = (63-3x) \bmod 8$ | $y_2$ | $y_1$ | $y_0$ |
|-------|-------|-------|-------|-------|---------|-----------------------|-------|-------|-------|
| 0     | 0     | 0     | 0     | 0     | 63      | 7                     | 1     | 1     | 1     |
| 0     | 0     | 0     | 0     | 1     | 60      | 4                     | 1     | 0     | 0     |
| 0     | 0     | 0     | 1     | 0     | 57      | 1                     | 0     | 0     | 1     |
| 0     | 0     | 0     | 1     | 1     | 54      | 6                     | 1     | 1     | 0     |
| 0     | 0     | 1     | 0     | 0     | 51      | 3                     | 0     | 1     | 1     |
| 0     | 0     | 1     | 0     | 1     | 48      | 0                     | 0     | 0     | 0     |
| 0     | 0     | 1     | 1     | 0     | 45      | 5                     | 1     | 0     | 1     |
| 0     | 0     | 1     | 1     | 1     | 42      | 2                     | 0     | 1     | 0     |
| 0     | 1     | 0     | 0     | 0     | 39      | 7                     | 1     | 1     | 1     |
| 0     | 1     | 0     | 0     | 1     | 36      | 4                     | 1     | 0     | 0     |
| 0     | 1     | 0     | 1     | 0     | 33      | 1                     | 0     | 0     | 1     |
| 0     | 1     | 0     | 1     | 1     | 30      | 6                     | 1     | 1     | 0     |
| 0     | 1     | 1     | 0     | 0     | 27      | 3                     | 0     | 1     | 1     |
| 0     | 1     | 1     | 0     | 1     | 24      | 0                     | 0     | 0     | 0     |
| 0     | 1     | 1     | 1     | 0     | 21      | 5                     | 1     | 0     | 1     |
| 0     | 1     | 1     | 1     | 1     | 18      | 2                     | 0     | 1     | 0     |
| 1     | 0     | 0     | 0     | 0     | 15      | 7                     | 1     | 1     | 1     |
| 1     | 0     | 0     | 0     | 1     | 12      | 4                     | 1     | 0     | 0     |
| 1     | 0     | 0     | 1     | 0     | 9       | 1                     | 0     | 0     | 1     |
| 1     | 0     | 0     | 1     | 1     | 6       | 6                     | 1     | 1     | 0     |
| 1     | 0     | 1     | 0     | 0     | 3       | 3                     | 0     | 1     | 1     |

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | d | d | d | d | d |
| 1 | 0 | 1 | 1 | 1 | d | d | d | d | d |
| 1 | 1 | 0 | 0 | 0 | d | d | d | d | d |
| 1 | 1 | 0 | 0 | 1 | d | d | d | d | d |
| 1 | 1 | 0 | 1 | 0 | d | d | d | d | d |
| 1 | 1 | 0 | 1 | 1 | d | d | d | d | d |
| 1 | 1 | 1 | 0 | 0 | d | d | d | d | d |
| 1 | 1 | 1 | 0 | 1 | d | d | d | d | d |
| 1 | 1 | 1 | 1 | 0 | d | d | d | d | d |
| 1 | 1 | 1 | 1 | 1 | d | d | d | d | d |

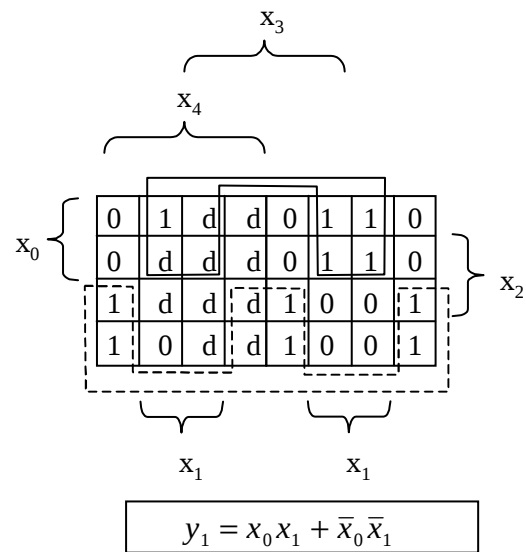
b) Minimieren Sie die disjunkten und die konjunktiven Normalformen der Ausgangsfunktionen mit Hilfe von Karnaughdiagrammen. Benutzen Sie dazu das folgende Karnaughdiagramm ( $X = (x_4x_3x_2x_1x_0)_2$ ).



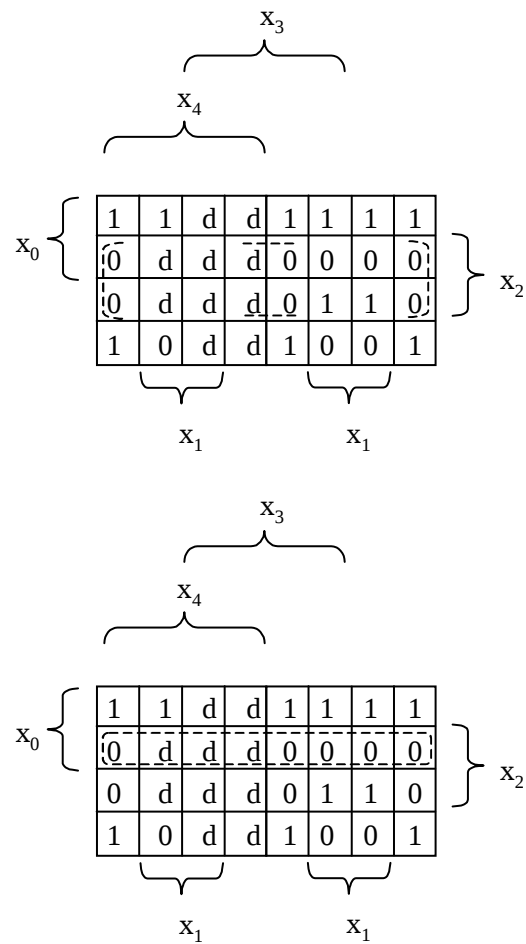
Minimierung der KNF von  $y_1$



Minimierung der DNF von  $y_1$ :



Minimierung der KNF von  $y_2$ :



|       |   |   |   |   |   |   |   |       |       |  |
|-------|---|---|---|---|---|---|---|-------|-------|--|
|       |   |   |   |   |   |   |   | $x_3$ |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   | $x_4$ |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
| $x_0$ | 1 | 1 | d | d | 1 | 1 | 1 | 1     | $x_2$ |  |
|       | 0 | d | d | d | 0 | 0 | 0 | 0     |       |  |
|       | 0 | d | d | d | 0 | 1 | 1 | 0     |       |  |
|       | 1 | 0 | d | d | 1 | 0 | 0 | 1     |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |

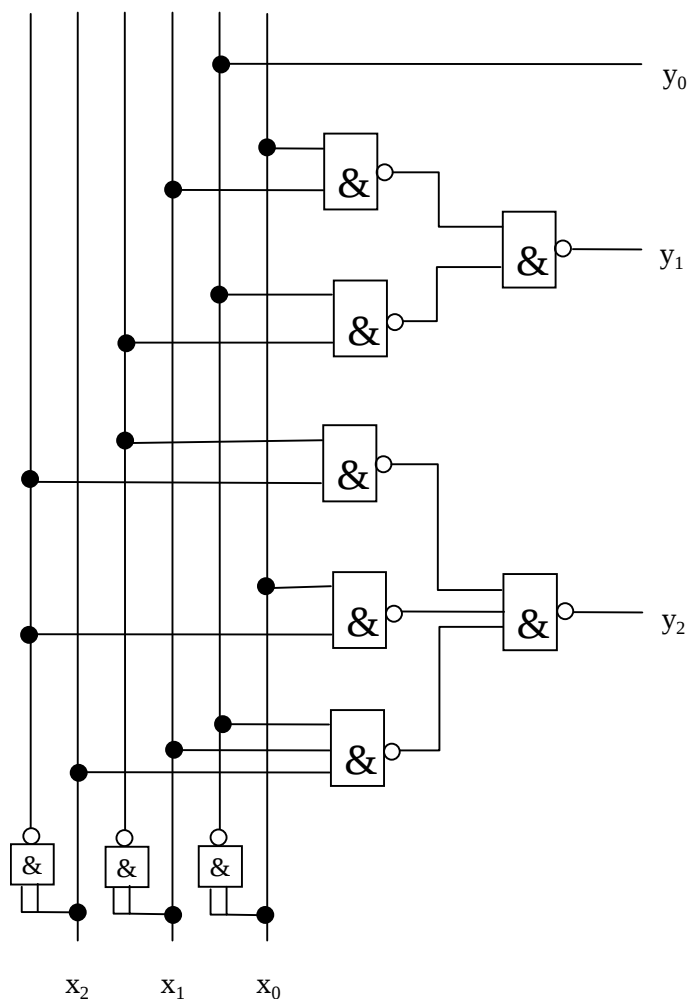
$$\begin{array}{c}
 \begin{array}{c} x_3 \\ \underbrace{\hspace{10em}} \end{array} \\
 \begin{array}{c} x_4 \\ \underbrace{\hspace{10em}} \end{array} \\
 \begin{array}{c} x_0 \left\{ \begin{array}{|c|c|c|c|c|c|c|c|} \hline \overline{1} & \overline{1} & \overline{d} & \overline{d} & \overline{1} & \overline{1} & \overline{1} & \overline{1} \\ \hline 0 & d & d & d & 0 & 0 & 0 & 0 \\ \hline 0 & d & d & d & 0 & 1 & 1 & 0 \\ \hline 1 & 0 & d & d & 1 & 0 & 0 & 1 \\ \hline \end{array} \right\} x_2 \\
 \begin{array}{c} \underbrace{\hspace{4em}} \quad \underbrace{\hspace{4em}} \\ x_1 \quad \quad x_1 \end{array}
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{c} x_3 \\ \underbrace{\hspace{10em}} \end{array} \\
 \begin{array}{c} x_4 \\ \underbrace{\hspace{10em}} \end{array} \\
 \begin{array}{c} x_0 \left\{ \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 1 & d & d & 1 & 1 & 1 & 1 \\ \hline 0 & d & d & d & 0 & 0 & 0 & 0 \\ \hline 0 & \overline{d} & \overline{d} & d & 0 & \overline{1} & \overline{1} & 0 \\ \hline 1 & 0 & d & d & 1 & 0 & 0 & 1 \\ \hline \end{array} \right\} x_2 \\
 \begin{array}{c} \underbrace{\hspace{4em}} \quad \underbrace{\hspace{4em}} \\ x_1 \quad \quad x_1 \end{array}
 \end{array}$$

$$y_2 = \bar{x}_1 \bar{x}_2 + x_0 \bar{x}_2 + \bar{x}_0 x_1 x_2$$

Minimierung der KNF von  $y_0$

|       |   |   |   |   |   |   |   |       |       |  |
|-------|---|---|---|---|---|---|---|-------|-------|--|
|       |   |   |   |   |   |   |   | $x_3$ |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   | $x_4$ |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
| $x_0$ | 0 | 0 | d | d | 0 | 0 | 0 | 0     | $x_2$ |  |
|       | 0 | d | d | d | 0 | 0 | 0 | 0     |       |  |
|       | 1 | d | d | d | 1 | 1 | 1 | 1     |       |  |
|       | 1 | 1 | d | d | 1 | 1 | 1 | 1     |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |
|       |   |   |   |   |   |   |   |       |       |  |



- d) Realisieren Sie das Schaltnetz für die minimierten Ausgangsfunktionen mit NOR-Gattern.

Aus den minimierten KNF von  $y_2$ ,  $y_1$  und  $y_0$  ergeben sich die Darstellungen:

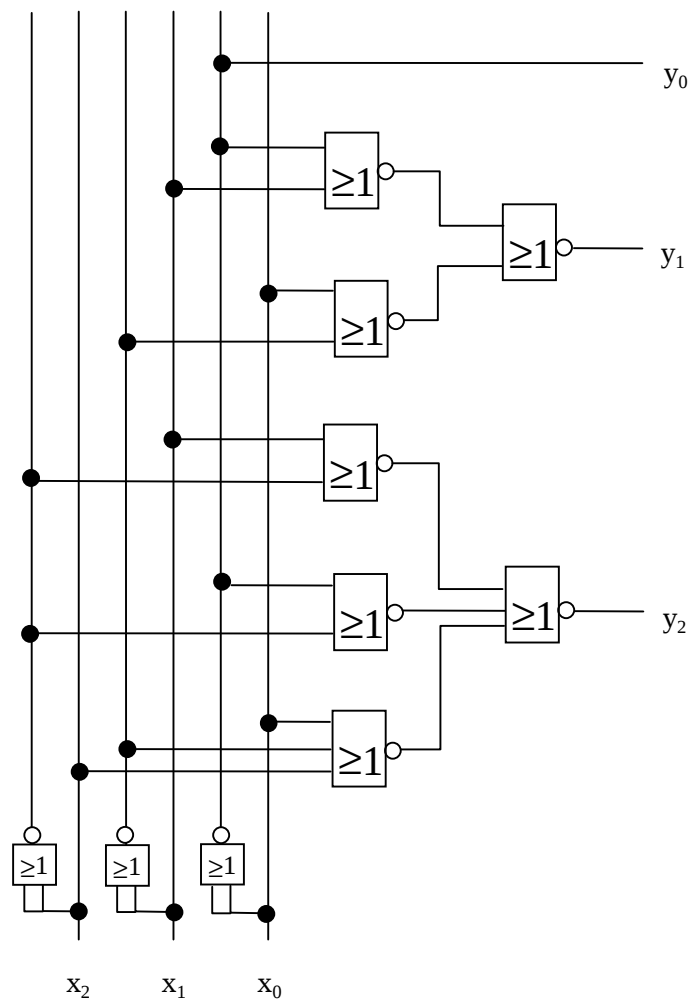
$$y_2 = (x_1 + \bar{x}_2) \cdot (\bar{x}_0 + \bar{x}_2) \cdot (x_0 + \bar{x}_1 + x_2)$$

$$= \overline{\overline{(x_1 + \bar{x}_2)} + \overline{(\bar{x}_0 + \bar{x}_2)} + \overline{(x_0 + \bar{x}_1 + x_2)}}$$

$$y_1 = (\bar{x}_0 + x_1) \cdot (x_0 + \bar{x}_1)$$

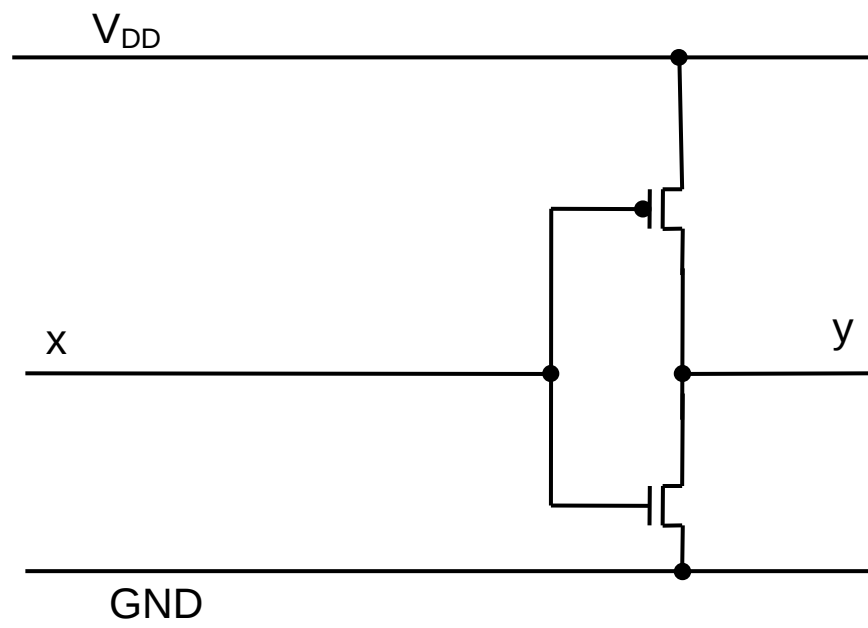
$$= \overline{\overline{(\bar{x}_0 + x_1)} + \overline{(x_0 + \bar{x}_1)}}$$

$$y_0 = \bar{x}_0$$



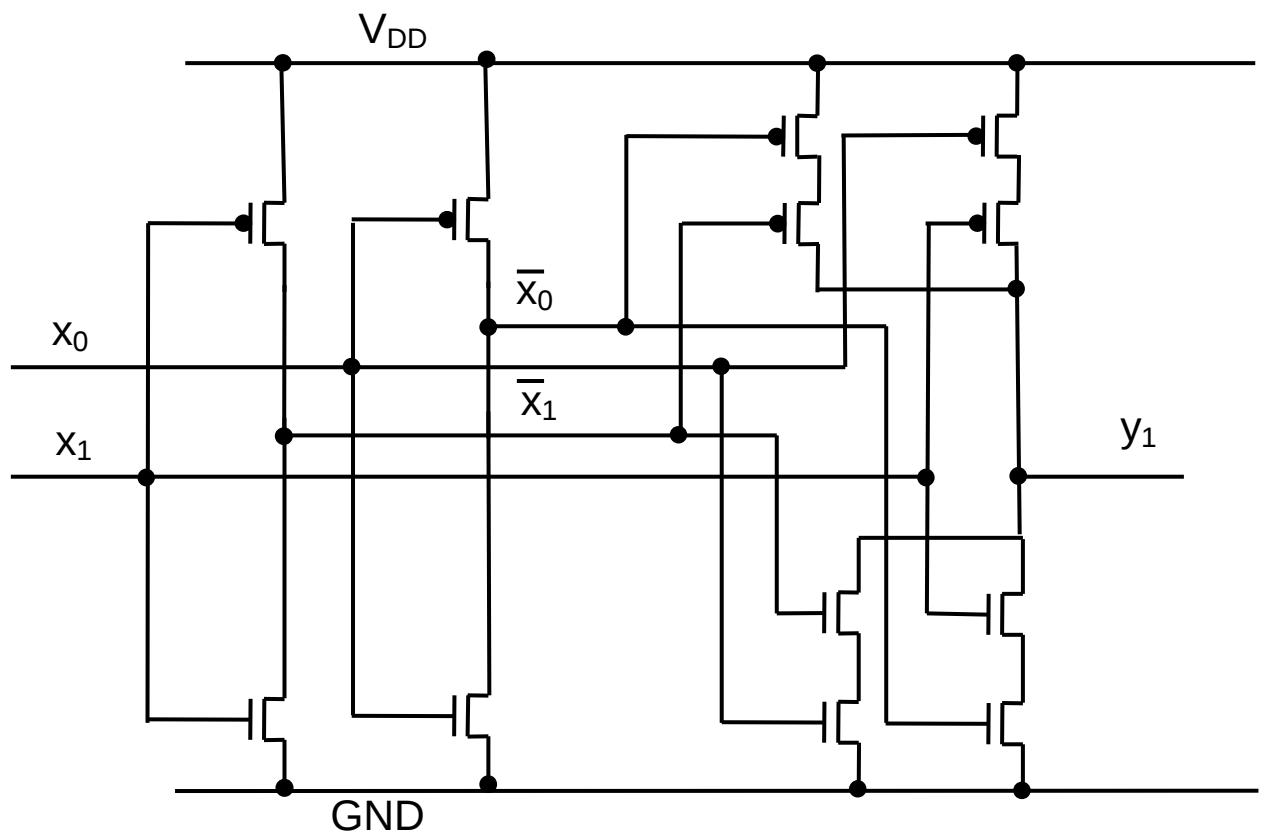
- e) Realisieren Sie das Schaltnetz in CMOS als jeweils ein Komplexgatter für jede Ausgangsvariable.

$$y_0 = x_0$$



$$y_1 = x_0 x_1 + \bar{x}_0 \bar{x}_1$$

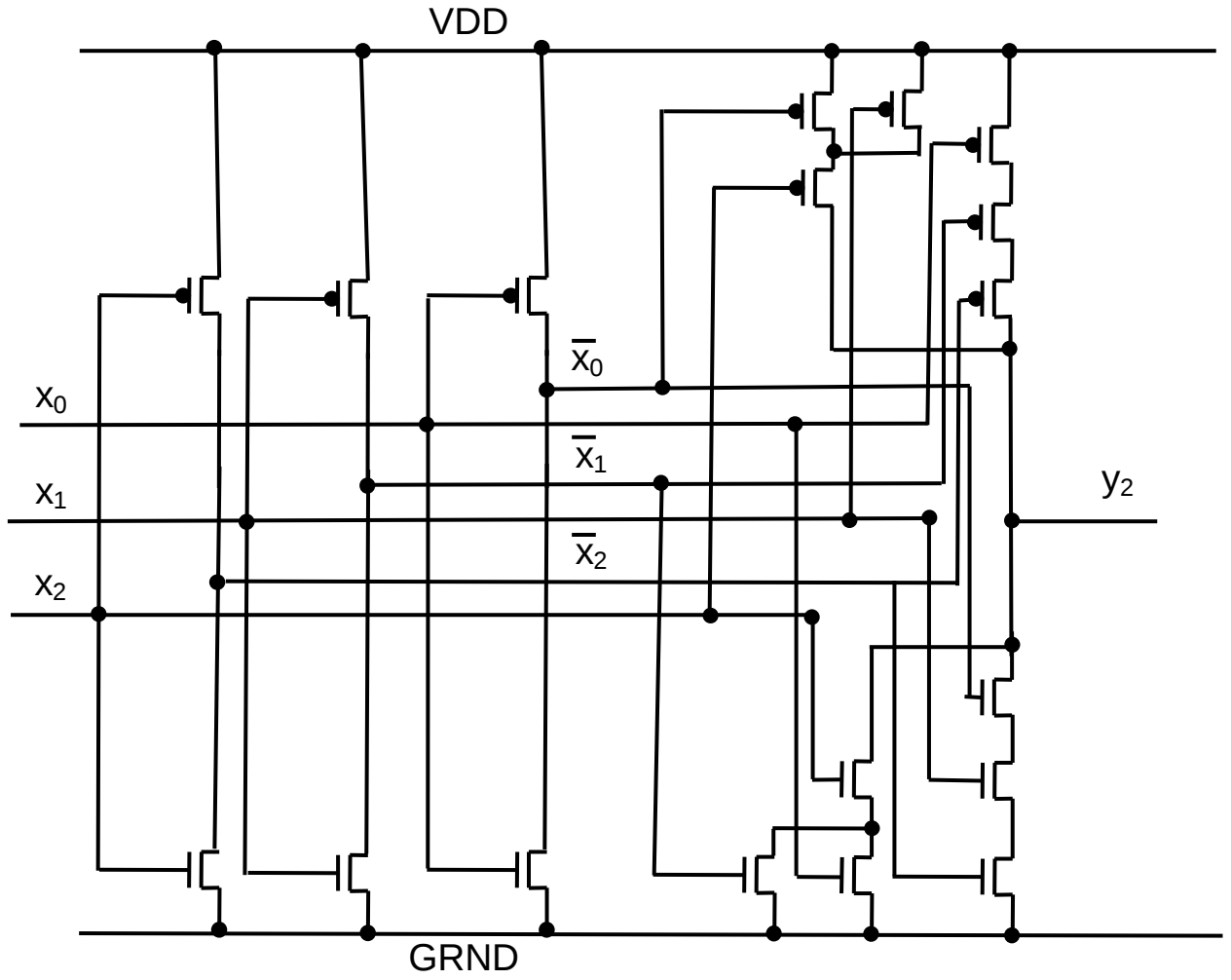
$$y_1 = (\bar{x}_0 + x_1) \cdot (x_0 + \bar{x}_1) = \overline{(\bar{x}_0 + x_1)} \cdot \overline{(x_0 + \bar{x}_1)} = \overline{(\bar{x}_0 + x_1)} + \overline{(x_0 + \bar{x}_1)} = x_0 \bar{x}_1 + \bar{x}_0 x_1$$



$$y_2 = \bar{x}_1\bar{x}_2 + x_0\bar{x}_2 + \bar{x}_0x_1x_2 = (\bar{x}_1 + x_0)\bar{x}_2 + \bar{x}_0x_1x_2$$

$$y_2 = (x_1 + \bar{x}_2) \cdot (\bar{x}_0 + \bar{x}_2) \cdot (x_0 + \bar{x}_1 + x_2) = \overline{(x_1 + \bar{x}_2) + (\bar{x}_0 + \bar{x}_2) + (x_0 + \bar{x}_1 + x_2)}$$

$$= \overline{\bar{x}_1x_2 + x_0x_2 + \bar{x}_0x_1\bar{x}_2} = \overline{(\bar{x}_1 + x_0)x_2 + \bar{x}_0x_1\bar{x}_2}$$



## Aufgabe 2

Minimieren Sie die folgende Funktion f mit dem Verfahren von Quine und McCluskey.

$$f = \overline{x_0}x_1\overline{x_2}x_3 + \overline{x_0}x_1x_2\overline{x_3} + \overline{x_0}x_1x_2x_3 + x_0\overline{x_1}\overline{x_2} \cdot \overline{x_3} + \overline{x_0}x_1\overline{x_2}x_3 + \overline{x_0}x_1x_2\overline{x_3}$$

### Algorithmus von McCluskey

$$\begin{array}{cccccc} x_0x_1x_2x_3 + & x_0x_1\overline{x_2}x_3 + & x_0x_1\overline{x_2}\overline{x_3} + & x_0x_1\overline{x_2}\overline{x_3} + & \overline{x_0}x_1x_2x_3 + & \overline{x_0}x_1\overline{x_2}\overline{x_3} \\ \text{I} & \text{II} & \text{III} & \text{IV} & \text{V} & \text{VI} \end{array}$$

#### 1. Schritt (zweite Vereinfachungsregel):

$$\begin{array}{ccccccc} x_0x_1x_3 + & x_0x_1x_2 + & x_1x_2x_3 + & x_0x_1\overline{x_2} + & x_0x_1\overline{x_3} + & x_1x_2\overline{x_3} + & \overline{x_0}x_1x_2 \\ \text{A} & \text{B} & \text{C} & \text{D} & \text{E} & \text{F} & \text{G} \end{array}$$

#### 2. Schritt (zweite Vereinfachungsregel):

$$\begin{array}{cccc} x_0x_1 + & x_0x_1 + & x_1x_2 + & x_1x_2 \\ \text{a} & \text{b} & \text{c} & \text{d} \end{array}$$

#### 3. Schritt (Zusammenfassen mit Regel (i)):

$$x_0x_1 + x_1x_2$$

### Algorithmus von Quine

#### 1. Schritt (erstellen der Tabelle)

|                                                 | $x_0x_1$ | $x_1x_2$ |
|-------------------------------------------------|----------|----------|
| $x_0x_1x_2x_3$                                  | 1        | 0        |
| $\overline{x_0}x_1\overline{x_2}x_3$            | 1        | 1        |
| $\overline{x_0}x_1x_2\overline{x_3}$            | 1        | 1        |
| $\overline{x_0}x_1\overline{x_2}\overline{x_3}$ | 1        | 0        |
| $\overline{x_0}\overline{x_1}x_2x_3$            | 0        | 1        |
| $\overline{x_0}\overline{x_1}\overline{x_2}x_3$ | 0        | 1        |

**2. Schritt (streichen der dominanten Zeilen)**

|                                                    | $x_0 x_1$ | $x_1 x_2$ |
|----------------------------------------------------|-----------|-----------|
| $x_0 x_1 x_2 x_3$                                  | 1         | 0         |
| $\overline{x_0} x_1 x_2 x_3$                       | 1         | 1         |
| $x_0 \overline{x_1} x_2 x_3$                       | 1         | 1         |
| $x_0 x_1 \overline{x_2} x_3$                       | 1         | 0         |
| $\overline{x_0} x_1 x_2 \overline{x_3}$            | 0         | 1         |
| $\overline{x_0} x_1 \overline{x_2} \overline{x_3}$ | 0         | 1         |

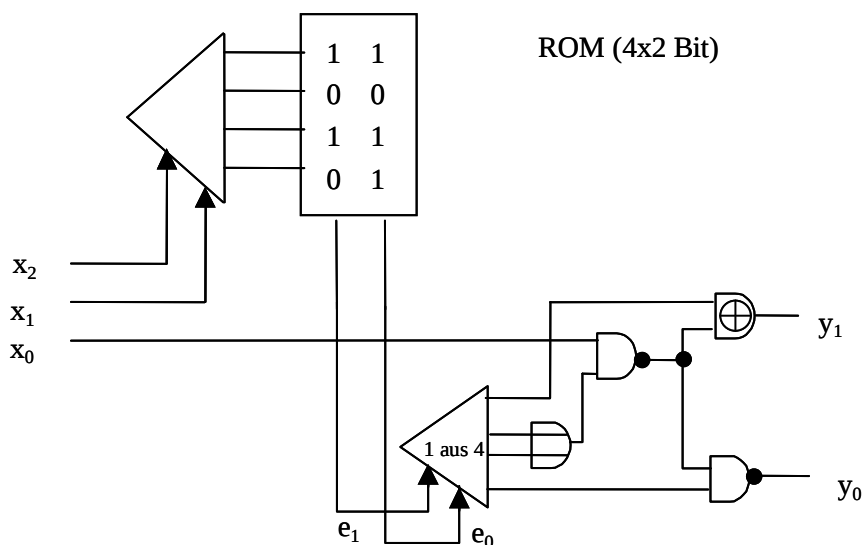
**$\Rightarrow$ DMF:**

$$f = x_0 x_1 + x_1 x_2$$

Serie 6

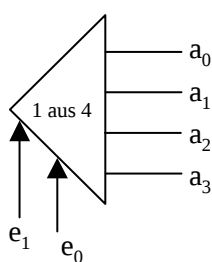
Aufgabe 1

Gegeben sei die folgende Schaltung:



a) Stellen Sie die vollständige Funktionstabelle für die dargestellte Schaltung auf.

Zur leichten Auswertung führen wir die Hilfsvariablen  $e_2$  und  $e_3$  ein, sie geben den jeweiligen Wert am Ausgang des OR-, bzw. NAND-Gatters an. Der 1 aus 4 Dekoder hat die folgende Funktion:



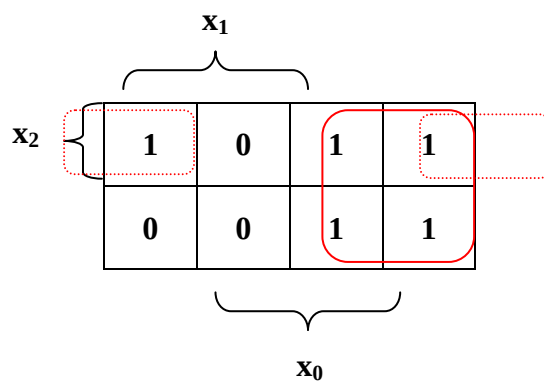
| $e_1$ | $e_0$ | $a_3$ | $a_2$ | $a_1$ | $a_0$ |
|-------|-------|-------|-------|-------|-------|
| 0     | 0     | 0     | 0     | 0     | 1     |
| 0     | 1     | 0     | 0     | 1     | 0     |
| 1     | 0     | 0     | 1     | 0     | 0     |
| 1     | 1     | 1     | 0     | 0     | 0     |

Damit ergibt sich folgende Funktionstabelle:

| $x_2$ | $x_1$ | $x_0$ | $e_0$ | $e_1$ | $e_2$ | $e_3$ | $y_1$ | $y_0$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0     | 0     | 0     | 1     | 1     | 0     | 1     | 1     | 0     |
| 0     | 0     | 1     | 1     | 1     | 0     | 1     | 1     | 0     |
| 0     | 1     | 0     | 0     | 0     | 0     | 1     | 0     | 1     |
| 0     | 1     | 1     | 0     | 0     | 0     | 1     | 0     | 1     |
| 1     | 0     | 0     | 1     | 1     | 0     | 1     | 1     | 0     |
| 1     | 0     | 1     | 1     | 1     | 0     | 1     | 1     | 0     |
| 1     | 1     | 0     | 1     | 0     | 1     | 1     | 1     | 1     |
| 1     | 1     | 1     | 1     | 0     | 1     | 0     | 0     | 1     |

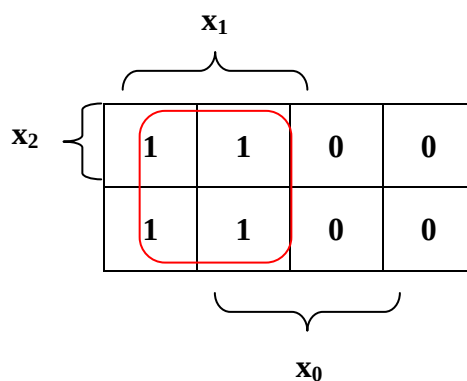
b) Minimieren Sie die disjunktiven Normalformen der Ausgangsfunktionen mit Karnaughdiagrammen.

$y_1$ :



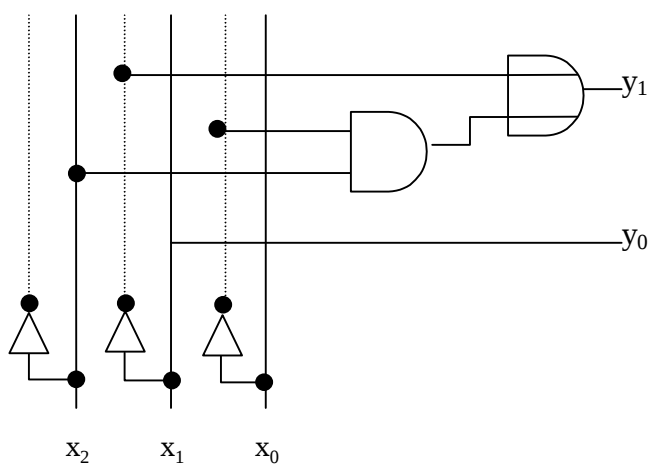
$$y_1 = \overline{x_1} + x_2 \overline{x_0}$$

$y_0$ :



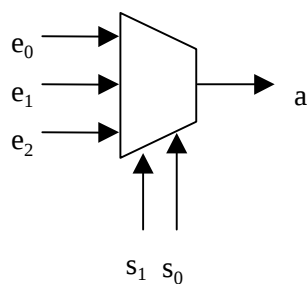
$$y_0 = x_1$$

c) Realisieren Sie die minimierten Ausgangsfunktionen mit möglichst wenigen Gattern.



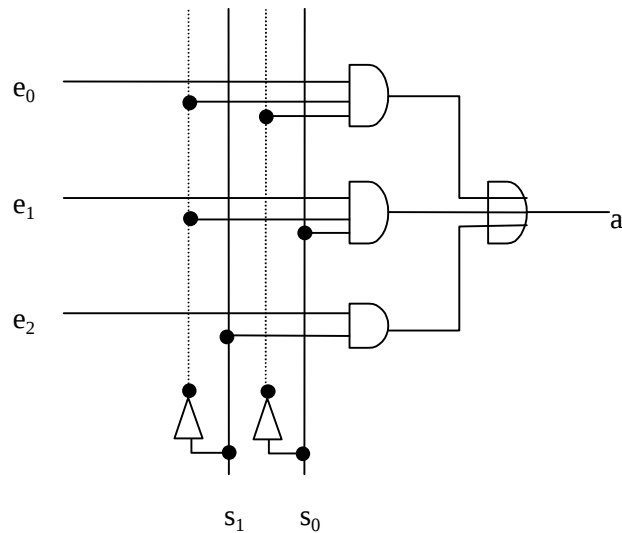
## Aufgabe 2

Entwerfen Sie mit Hilfe der DNF ein Schaltnetz für einen 3 auf 1 Multiplexer.



Über die Steuereingänge des Multiplexers wird festgelegt, welches der Eingangssignale  $e_0, e_1, e_2$  am Ausgang  $a$  ausgegeben wird. Da wir 3 Eingänge haben, brauchen wir 2 Steuereingänge (die Binärdarstellung der Zahlen 0,1,2,3 besteht aus 2 Bits). Der 3 auf 1 Multiplexer arbeitet nach der folgenden Funktionstabelle:

| $s_1$ | $s_0$ | $a$   |
|-------|-------|-------|
| 0     | 0     | $e_0$ |
| 0     | 1     | $e_1$ |
| 1     | x     | $e_2$ |



### Aufgabe 3

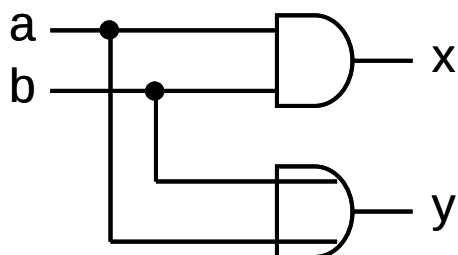
- a) Skizzieren Sie ein Sortierelement, das zwei Bits sortiert. Benutzen Sie dafür die in der Vorlesung vorgestellten Elementargatter mit zwei Eingängen.

Wir nehmen zwei Bits  $a$  und  $b$  als Eingabe. Die Ausgangsbits sind  $x$  und  $y$ . Wir erstellen eine Wahrheitstabelle:

| $a$ | $b$ | $x$ | $y$ |
|-----|-----|-----|-----|
| 0   | 0   | 0   | 0   |
| 0   | 1   | 0   | 1   |
| 1   | 0   | 0   | 1   |
| 1   | 1   | 1   | 1   |

Entsprechend sind  $x = ab$  und  $y = a + b$

Das Sortierelement sieht folgendermaßen aus:



- b) Bauen Sie unter Zuhilfenahme der Ergebnisse aus a) ein Sortiernetz, das vier 1-bit Zahlen sortiert. Benutzen Sie dafür Bubblesort.

Wir nehmen vier Bits  $a$ ,  $b$ ,  $c$  und  $d$  als Eingabe. Der Sortieralgorithmus macht folgendes:

$a \longleftrightarrow b$        $c$        $d$

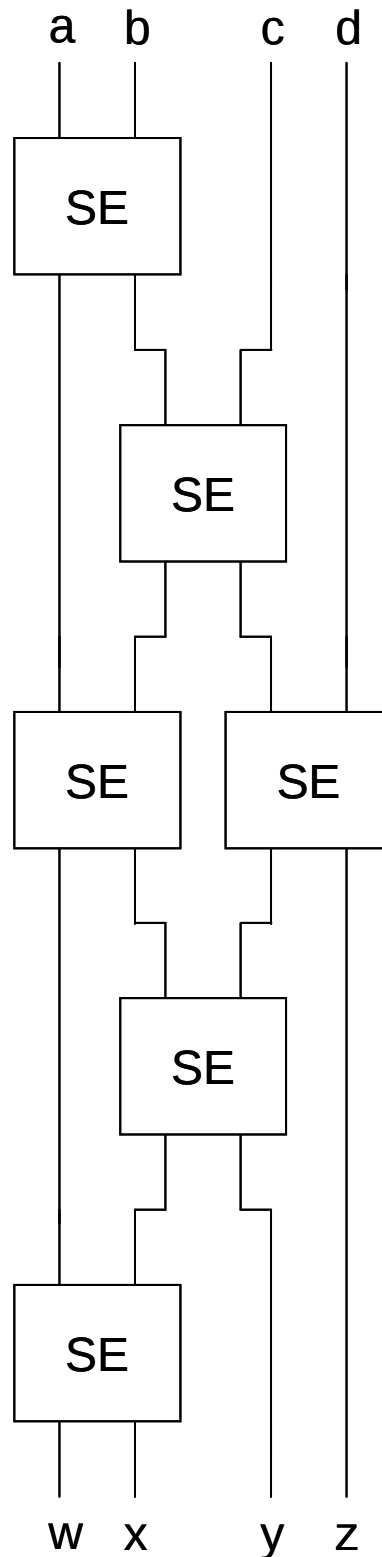
$a$        $b \longleftrightarrow c$        $d$

$a \longleftrightarrow b$        $c \longleftrightarrow d$

$a$        $b \longleftrightarrow c$        $d$

$a \longleftrightarrow b$        $c$        $d$

Das Sortiernetz sieht entsprechend so aus:



- c) Wie viele Zeiteinheiten braucht das Sortiernetz aus b), wenn ein Gatter 1 Zeiteinheit (1 ZE) braucht, um zu schalten.

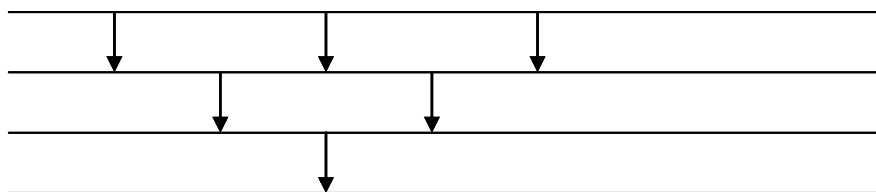
Ein Sortierelement hat die Zeitkomplexität von 1 ZE, da ein Sortierelement aus zwei parallel geschalteten Gattern besteht. Der längste Pfad verläuft über drei Sortierelemente. Entsprechend beträgt die Zeitkomplexität des Sortiernetzes 5 ZE.

**Serie 7**

**Aufgabe 1**

Ein Sortiernetz für 1-bit Daten ist ein auf Vergleichen und Vertauschen von jeweils zwei Elementen basierendes Schaltnetz. Eine mögliche Darstellung für ein Sortiernetz sind waagerechte Linien, die durch senkrechte Pfeile miteinander verbunden sind. Die zu sortierenden 1-bit Daten wandern von links nach rechts auf den waagerechten Linien durch das Diagramm und wenn sie an einen Pfeil gelangen, werden sie mit dem Datum am anderen Ende des Pfeils verglichen und, falls sie in falscher Reihenfolge stehen, vertauscht. Dabei wird immer das kleinere Element an das Pfeilende und das größere an die Pfeilspitze bewegt.

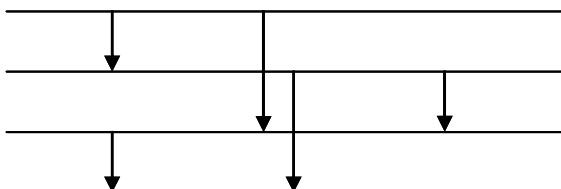
Beispiel: Sortiernetz für 4 Elemente (Bubble-Sort)



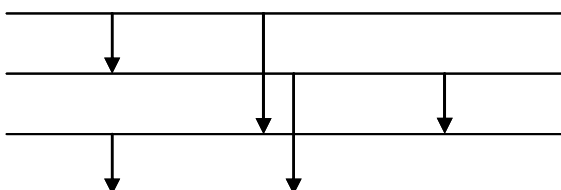
Die Schaltzeit eines Sortierelements (eines senkrechten Pfeils im Sortiernetz) sei 1 ZE (Zeiteinheit).

Aufgabe: Konstruieren Sie ein Sortiernetz für 4 Elemente, das

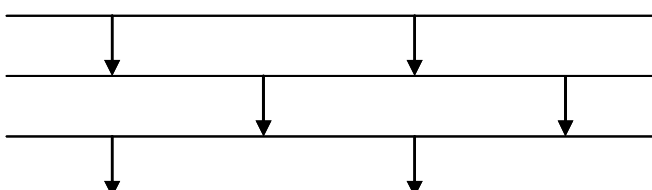
a) weniger als 4 ZE benötigt.



b) weniger als 6 Sortierelemente benötigt



c) höchstens 4 ZE benötigt, bei denen senkrechte Pfeile immer zwischen benachbarten Linien verlaufen.



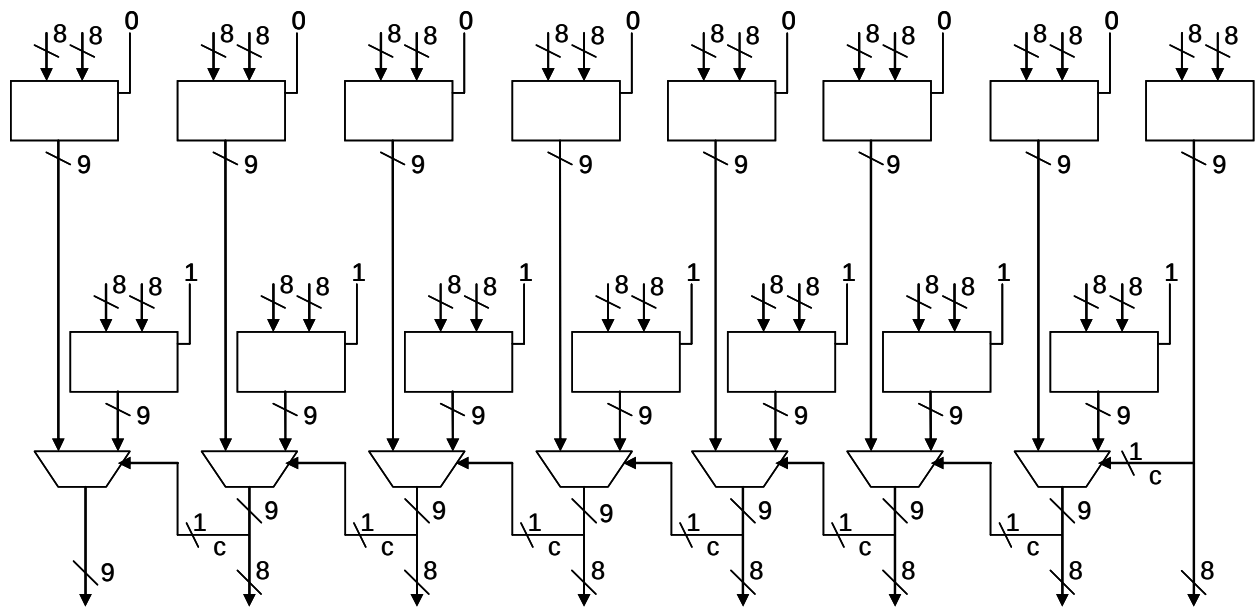
## Aufgabe 2

Betrachten Sie verschiedene Realisierungen von Carry-Select-Addierern.

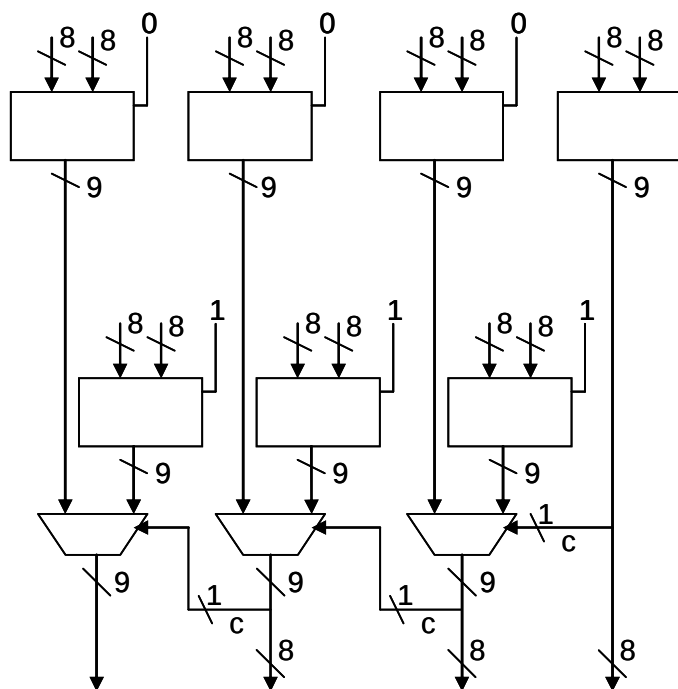
Gegeben sei folgendes Modell:

Ein Volladdierer belegt die Chipfläche 1 FE (Flächeneinheit) und benötigt als Schaltzeit 1 ZE (Zeiteinheit). Ein 1-bit-Multiplexer belegt ebenfalls 1 FE und braucht dieselbe Schaltzeit 1 ZE. A ist die Gesamtfläche, T ist die Gesamtzeit.

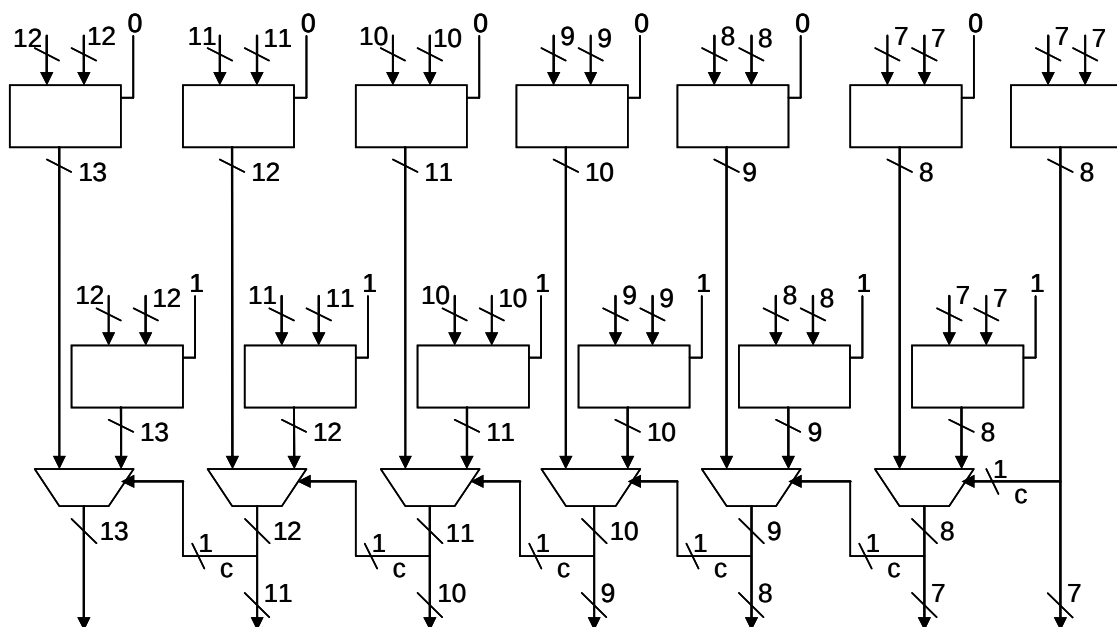
- a) Geben Sie das Blockschaltbild eines Carry-Select-Addierers an, der in der Zeit  $T=15$  ZE eine 64-bit Addition durchführt.



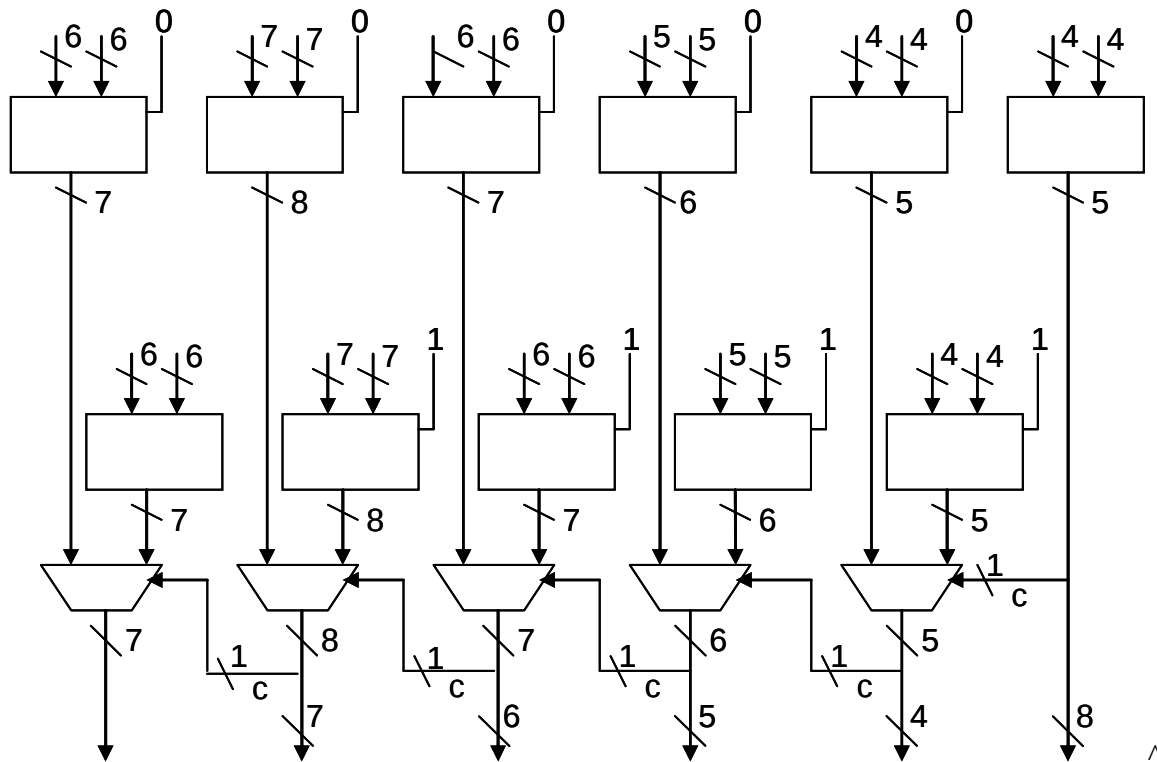
- b) Geben Sie verschiedene Carry-Select-Addierer an, die eine 32-bit Addition in  $T=11$  ZE durchführen (Skizzieren Sie einen davon, geben Sie bei dem anderen nur die relevante Daten an).



c) Geben Sie das Blockschaltbild eines 64-bit Carry-Select-Addierers an, mit  $T < 15$  ZE schaltet.



d) Konstruieren Sie einen 32-bit Carry-Select-Addierer, der minimal in Hinsicht auf Produkt von Fläche und Zeit ist. (Beweis ist nicht erforderlich, nur Berechnung des AT-Produktes).



Fläche:  $A = (4 + 2 \cdot 4 + 2 \cdot 5 + 2 \cdot 6 + 2 \cdot 7 + 2 \cdot 6) + (5 + 6 + 7 + 8 + 7) = 60 + 33 = 93$  FE

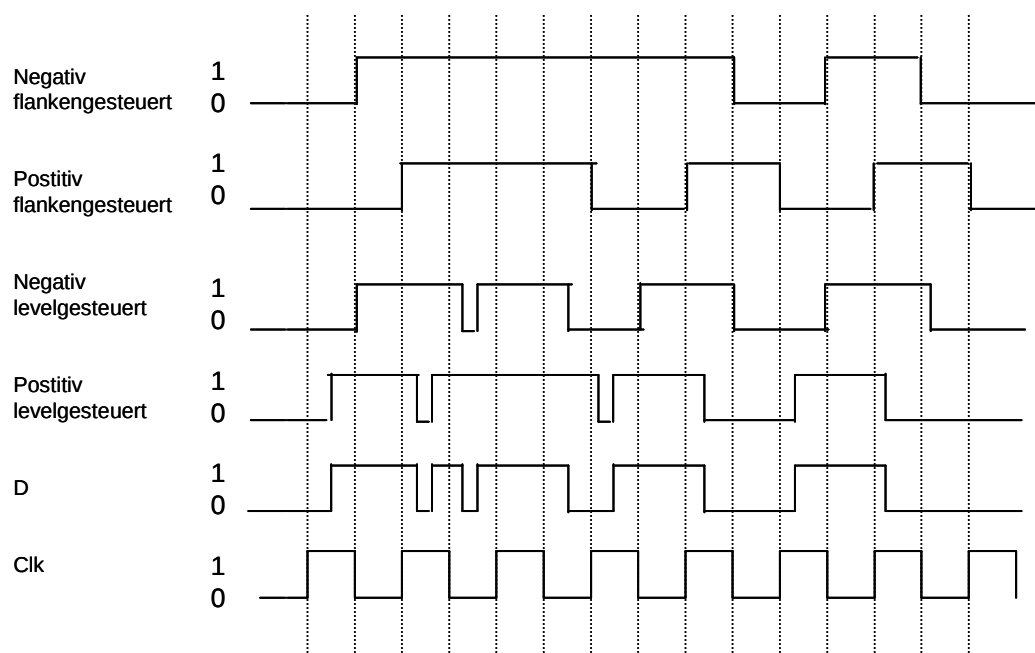
Zeit:  $T = 9$  ZE

Flächen-Zeit-Produkt:  $AT = 93 \cdot 9 = 837$

### Aufgabe 3

Wir betrachten das Ein-Ausgabe-Verhalten von vier verschiedenen D-Flipflop-Typen: Einen negativ flankengesteuerten, einen positiv flankengesteuerten, einen negativ levelgesteuerten und einen positiv levelgesteuerten D-Flipflop.

Vervollständigen Sie das folgende Impulsdiagramm:



**Serie 8**

**Aufgabe 1**

Es soll ein Schaltwerk entworfen werden, welches für zwei Binärzahlen  $a$  und  $b$ , welche bitseriell mit den MSB's (den höchstwertigsten Bits) zuerst dem Schaltwerk zugeführt werden, ob  $a = b$ ,  $a > b$  oder  $a < b$  gilt. Ein Startsignal  $s$  bestimmt, wann zwei neue Zahlen beginnen. Genau dann, wenn  $s = 1$  ist, liegen während des nächsten Taktes die MSB's der neuen Zahlen an.

a) Geben Sie die Ein- und Ausgangsvariablen des Schaltwerkes an.

Eingangsvariablen:

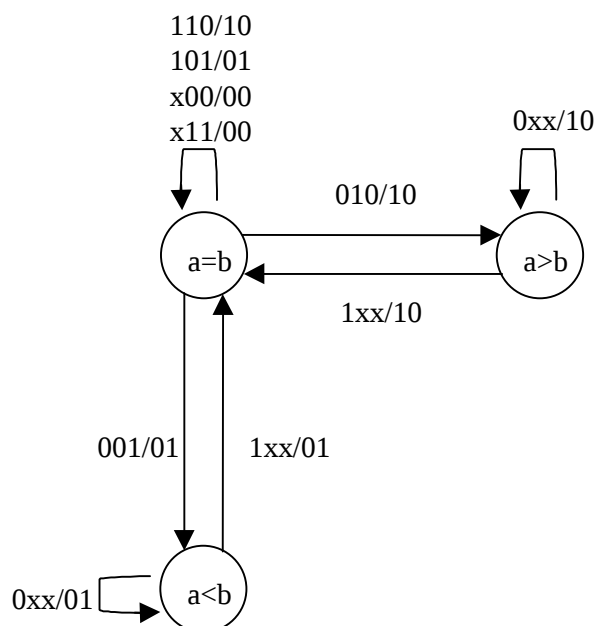
- $s$  Startsignal
- $a_i$  aktuelles Bit von  $a$
- $b_i$  aktuelles Bit von  $b$

Ausgangsvariablen:

- $e_0$  Ergebnisbit 1
- $e_1$  Ergebnisbit 2

| Ergebnis | $e_1$ | $e_0$ |
|----------|-------|-------|
| $a = b$  | 0     | 0     |
| $a < b$  | 0     | 1     |
| $a > b$  | 1     | 0     |

b) Zeichnen Sie den Mealyautomatengraphen des Schaltwerkes.

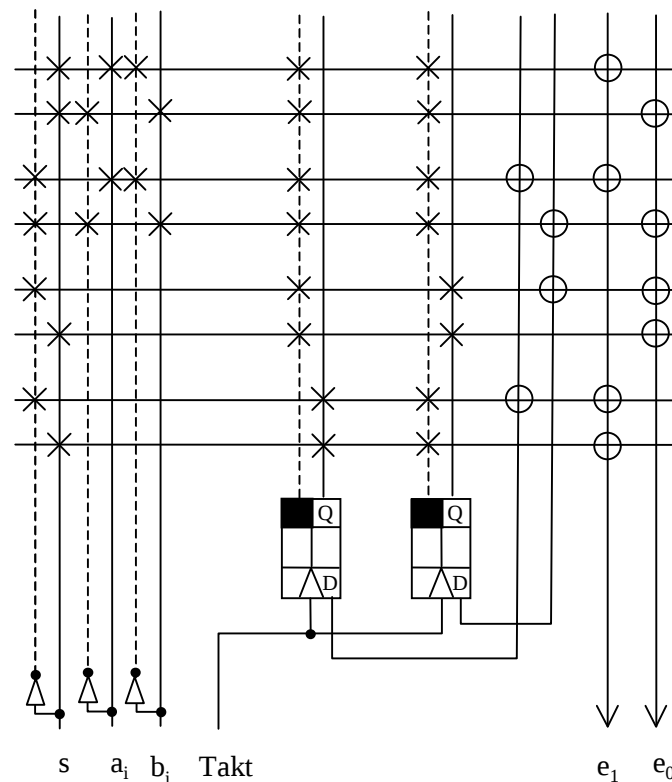


c) Zeichnen Sie die Realisierung des Schaltwerkes als FPLA mit D-Flipflops direkt aus der Funktionstabelle.

Zustandskodierung:

| Zustand | $q_1$ | $q_0$ |
|---------|-------|-------|
| $a = b$ | 0     | 0     |
| $a < b$ | 0     | 1     |
| $a > b$ | 1     | 0     |

| s | $a_i$ | $b_i$ | $q_1$ | $q_0$ | $e_1$ | $e_0$ | $q'_1$ | $q'_0$ |
|---|-------|-------|-------|-------|-------|-------|--------|--------|
| 1 | 1     | 0     | 0     | 0     | 1     | 0     | 0      | 0      |
| 1 | 0     | 1     | 0     | 0     | 0     | 1     | 0      | 0      |
| x | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      |
| x | 1     | 1     | 0     | 0     | 0     | 0     | 0      | 0      |
| 0 | 1     | 0     | 0     | 0     | 1     | 0     | 1      | 0      |
| 0 | 0     | 1     | 0     | 0     | 0     | 1     | 0      | 1      |
| 0 | x     | x     | 1     | 0     | 1     | 0     | 1      | 0      |
| 1 | x     | x     | 1     | 0     | 1     | 0     | 0      | 0      |
| 0 | x     | x     | 0     | 1     | 0     | 1     | 0      | 1      |
| 1 | x     | x     | 0     | 1     | 0     | 1     | 0      | 0      |
| x | x     | x     | 1     | 1     | x     | x     | x      | x      |



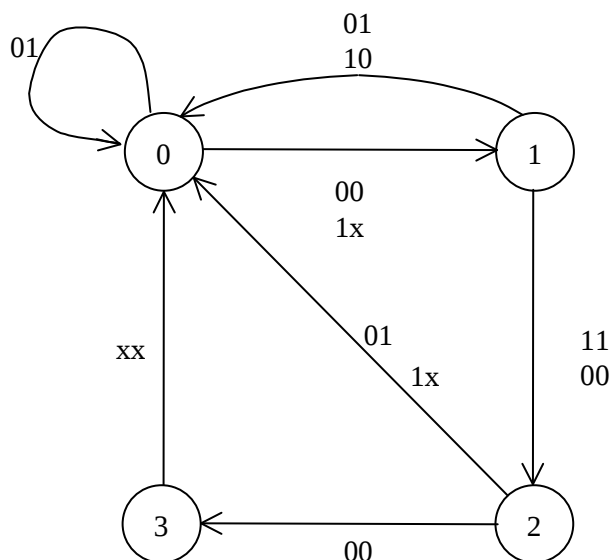
## Aufgabe 2

Entwerfen Sie ein Schaltwerk für einen Modulo- $n$ -Zähler. Das  $n$  soll dabei variabel sein und Werte von 1 bis 4 annehmen können. Der Wert von  $n$  soll über zwei Eingänge bestimmt werden. Der Zähler soll mit J-K-Flipflops aufgebaut werden. Bei der Umschaltung von einem  $n$  nach  $n'$  soll der Zähler normal weiter zählen, wenn sein Wert kleiner als  $n'$  ist und er soll auf 0 zählen, wenn sein aktueller Wert größer oder gleich  $n'$  ist.

Zur Bestimmung des Wertes  $n$  für den Modulo- $n$ -Zähler benutzen wir die beiden niederwertigsten Bits  $n_0$  und  $n_1$  der Binärdarstellung  $(n_2 n_1 n_0)_2$  der Dezimalzahl  $n$ :

| $n$ | $n_2$ | $n_1$ | $n_0$ |
|-----|-------|-------|-------|
| 1   | 0     | 0     | 1     |
| 2   | 0     | 1     | 0     |
| 3   | 0     | 1     | 1     |
| 4   | 1     | 0     | 0     |

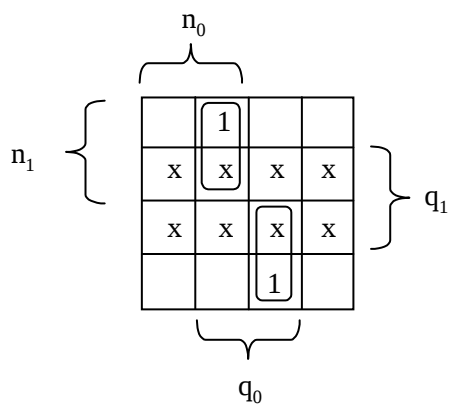
Der Zähler hat 4 Zustände, die Zustände repräsentieren die möglichen Reste modulo  $n$  (0,1,2,3). Der Zähler als Mooreautomat konstruiert, die Ausgabe für den jeweiligen Zustand ist dabei der Rest modulo  $n$ . Die Eingabe besteht aus den Bits  $n_1$ ,  $n_0$ . Daraus ergibt sich der folgende Automatengraph:



Damit ergibt sich für den Modulo- $n$ -Zähler folgende Funktionstabelle:

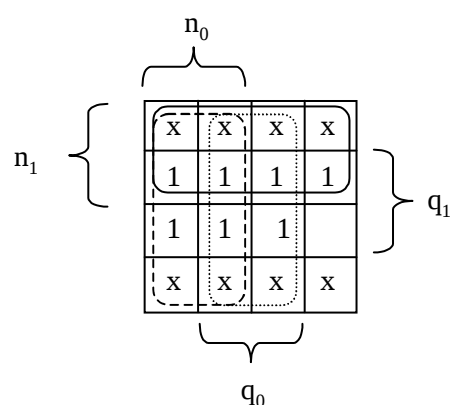
| $n_1$ | $n_0$ | $q_1$ | $q_0$ | $q'_1$ | $q'_0$ | $J_1$ | $K_1$ | $J_0$ | $K_0$ |
|-------|-------|-------|-------|--------|--------|-------|-------|-------|-------|
| 0     | 0     | 0     | 0     | 0      | 1      | 0     | x     | 1     | x     |
| 0     | 0     | 0     | 1     | 1      | 0      | 1     | x     | x     | 1     |
| 0     | 0     | 1     | 0     | 1      | 1      | x     | 0     | 1     | x     |
| 0     | 0     | 1     | 1     | 0      | 0      | x     | 1     | x     | 1     |
| 0     | 1     | 0     | 0     | 0      | 0      | 0     | x     | 0     | x     |
| 0     | 1     | 0     | 1     | 0      | 0      | 0     | x     | x     | 1     |
| 0     | 1     | 1     | 0     | 0      | 0      | x     | 1     | 0     | x     |
| 0     | 1     | 1     | 1     | 0      | 0      | x     | 1     | x     | 1     |
| 1     | 0     | 0     | 0     | 0      | 1      | 0     | x     | 1     | x     |
| 1     | 0     | 0     | 1     | 0      | 0      | 0     | x     | x     | 1     |
| 1     | 0     | 1     | 0     | 0      | 0      | x     | 1     | 0     | x     |
| 1     | 0     | 1     | 1     | 0      | 0      | x     | 1     | x     | 1     |
| 1     | 1     | 0     | 0     | 0      | 1      | 0     | x     | 1     | x     |
| 1     | 1     | 0     | 1     | 1      | 0      | 1     | x     | x     | 1     |
| 1     | 1     | 1     | 0     | 0      | 0      | x     | 1     | 0     | x     |
| 1     | 1     | 1     | 1     | 0      | 0      | x     | 1     | x     | 1     |

$J_1$ :



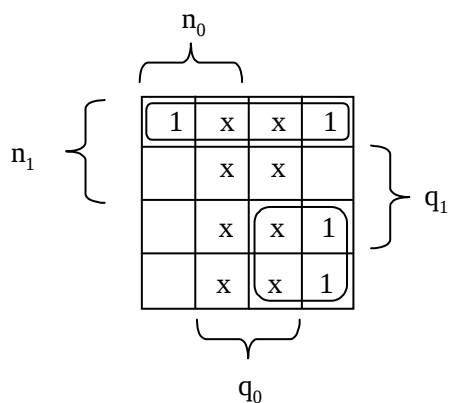
$$J_1 = n_0 n_1 q_0 + \bar{n}_0 \bar{n}_1 q_0$$

$K_1$ :



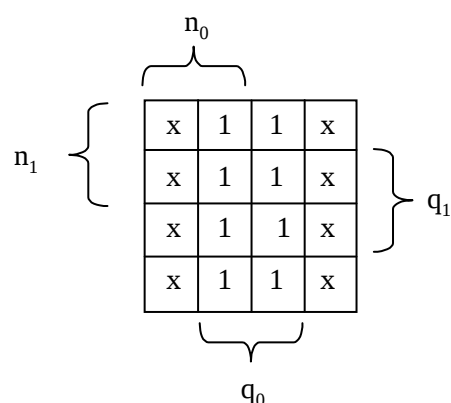
$$K_1 = n_0 + q_0 + n_1$$

$J_0 :$



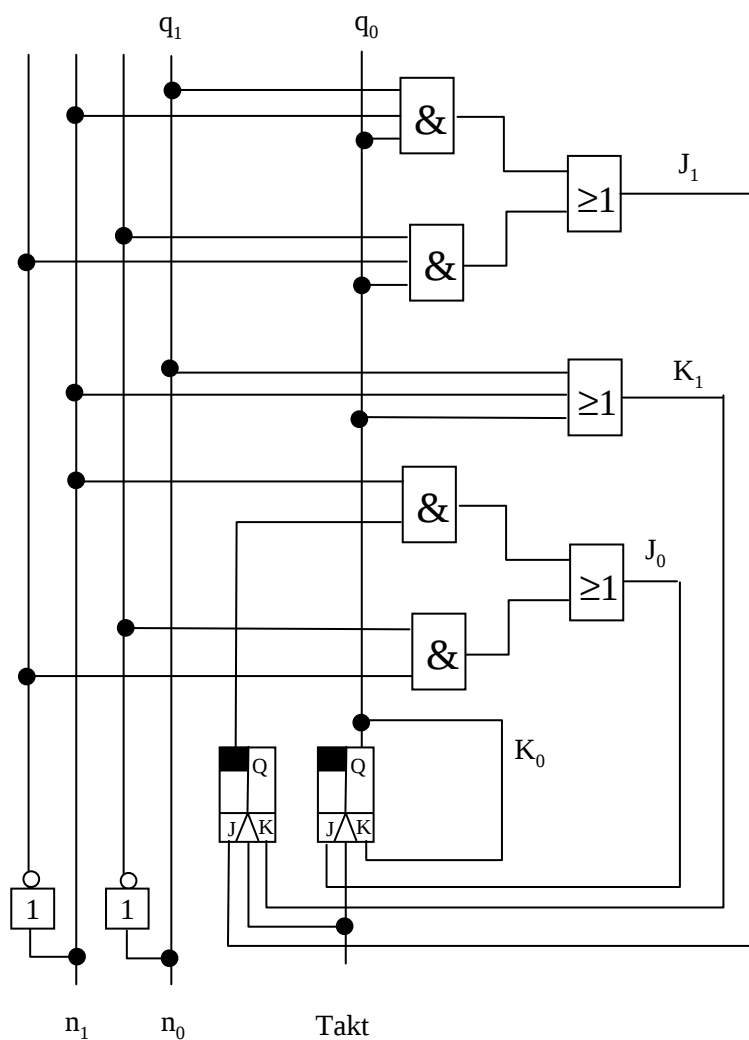
$$J_0 = n_1 \bar{q}_1 + \bar{n}_0 \bar{n}_1$$

$K_0 :$



$$K_0 = q_0 \text{ oder } K_0 = 1$$

Realisierung des Schaltwerkes mit Invertiern, AND- und OR-Gattern:

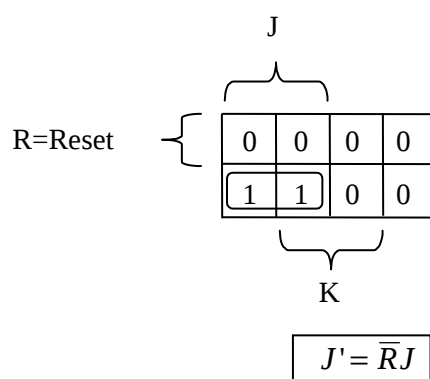


b) Ergänzen Sie ihre Schaltung um einen Reset-Eingang, mit dem man den Zähler auf 0 zurücksetzen kann.

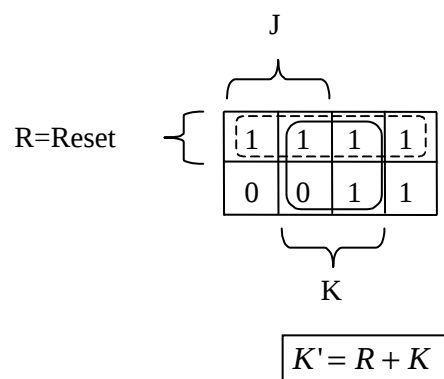
Erweiterung der Schaltung um einen Reseteingang, über den der Zähler auf 0 zurückgesetzt werden kann.

| Reset | $J$ | $K$ | $J'$ | $K'$ |
|-------|-----|-----|------|------|
| 0     | 0   | 0   | 0    | 0    |
| 0     | 0   | 1   | 0    | 1    |
| 0     | 1   | 0   | 1    | 0    |
| 0     | 1   | 1   | 1    | 1    |
| 1     | 0   | 0   | 0    | 1    |
| 1     | 0   | 1   | 0    | 1    |
| 1     | 1   | 0   | 0    | 1    |
| 1     | 1   | 1   | 0    | 1    |

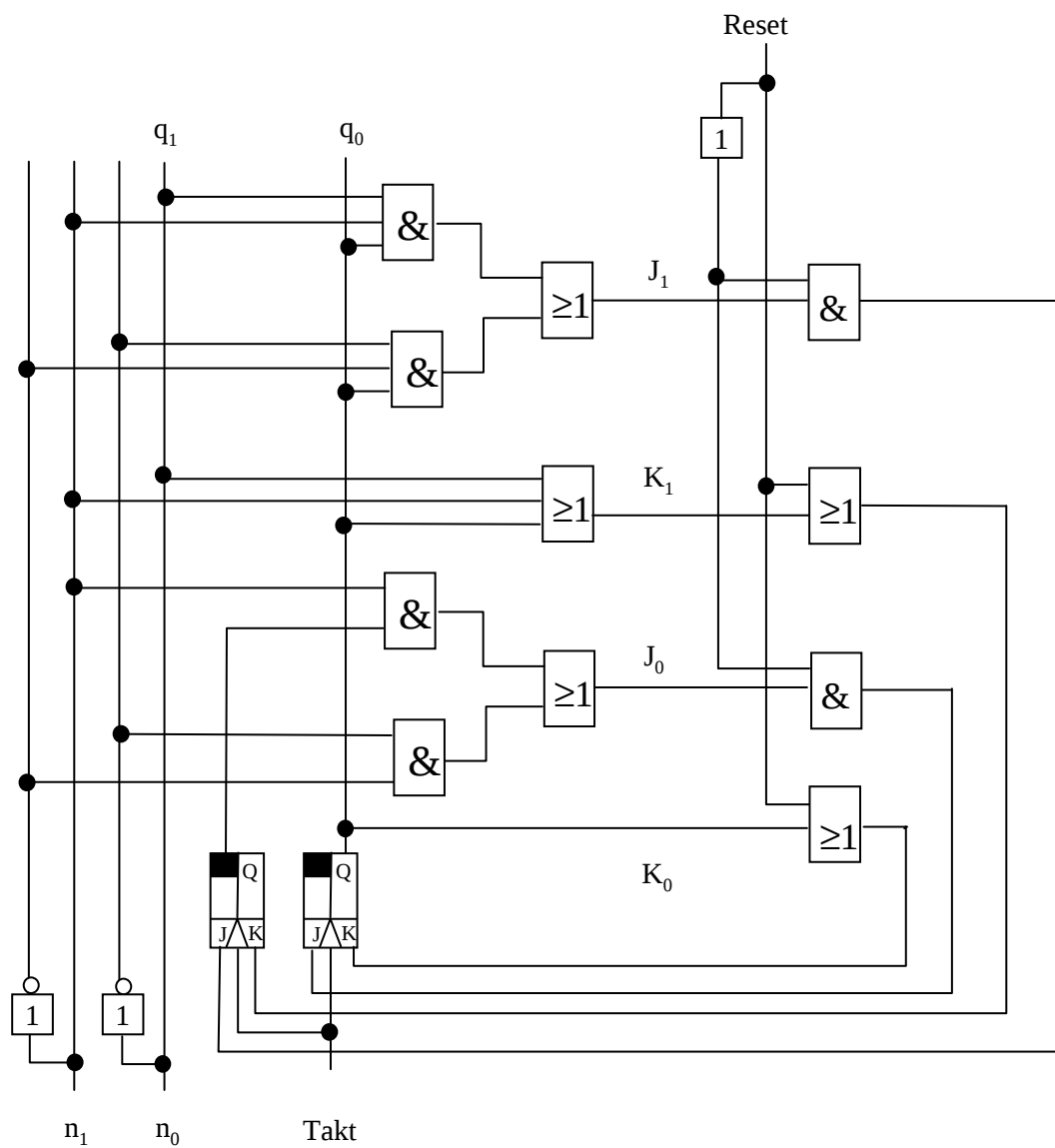
$J'_0:$



$K'_0:$



Die erweiterte Schaltung sieht wie folgt aus:



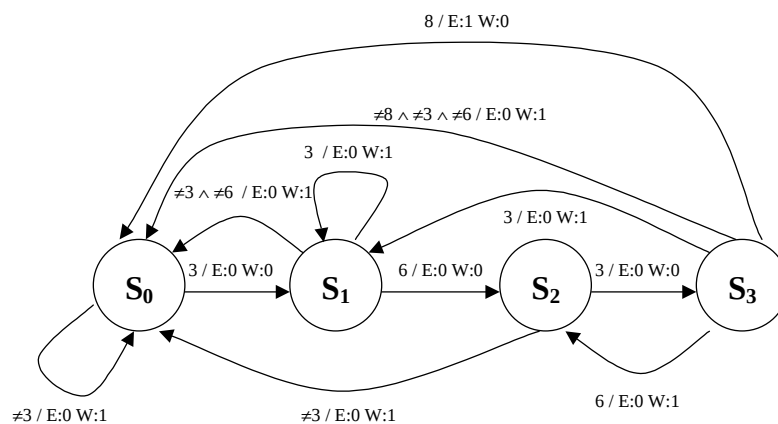
**Serie 9**

**Aufgabe 1**

Bei einer Sicherheitstür muss zum Öffnen die Tastenkombination 3-6-3-8 eingegeben werden, damit das Schloss sich öffnet. Die Ziffern werden nacheinander in eine Tastatur eingegeben. Der Benutzer kann beliebige Tasten (0...9) auf der Tastatur drücken; dabei soll allerdings ein Warnsignal W (angezeigt z.B. durch Aufleuchten einer Lampe beim Pförtner des Gebäudes) erzeugt werden, wenn eine fehlerhafte Eingabe festgestellt wird ( $W=0$ : keine fehlerhafte Eingabe,  $W=1$ : fehlerhafte Eingabe). Sobald aber die Ziffern 3-6-3-8 in der richtigen Reihenfolge eingegeben werden, soll das Schloss über das Signal  $E=1$  (E für entriegeln) geöffnet werden. Zu beachten ist aber, dass es Eingaben geben kann, durch die die bisherige Eingabe fehlerhaft wird, die aber selbst zur richtigen Kombination gehören. Diese richtige Kombination muss trotzdem erkannt werden. Außerdem muss in diesem Fall W auf 1 gesetzt werden, weil (auch) eine fehlerhafte Eingabe erkannt wurde. Die Tür fällt nach der Benutzung automatisch wieder ins Schloss und ist verriegelt, bis eine erneute Eingabe der Kombination sie erneut entriegelt.

Entwerfen Sie das Schaltwerk für die Steuerung dieses Schlosses:

- a) Zeichnen Sie einen Automatengraphen



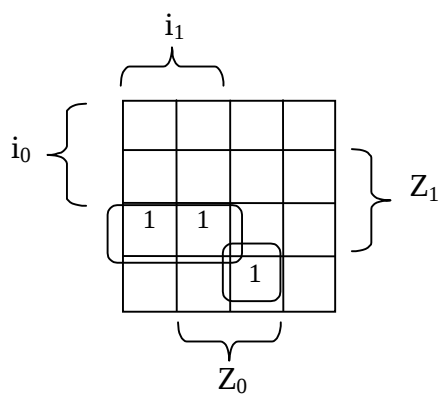
- b) Denken Sie sich eine sinnvolle Kodierung für die Eingabesignale (Die Ziffern der Tastatur) aus, um den Aufwand möglichst klein zu halten.

| $i_0$ | $i_1$ | Kodiert  |
|-------|-------|----------|
| 0     | 0     | 3        |
| 0     | 1     | 6        |
| 1     | 0     | 8        |
| 1     | 1     | Der Rest |

c) Stellen Sie eine Wertetabelle auf und minimieren Sie die Funktionen.

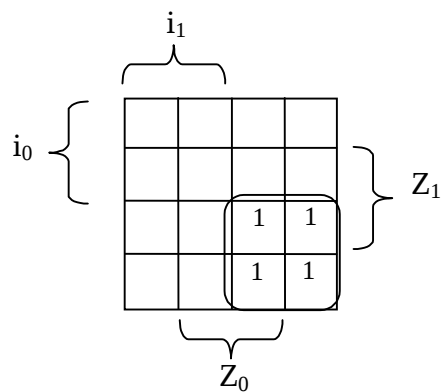
| $Z_0$ | $Z_1$ | $i_0$ | $i_1$ | $Z'_0$ | $Z'_1$ | E | W |
|-------|-------|-------|-------|--------|--------|---|---|
| 0     | 0     | 0     | 0     | 0      | 1      | 0 | 0 |
| 1     | 0     | 0     | 0     | 1      | 1      | 0 | 0 |
| 0     | 1     | 0     | 0     | 0      | 1      | 0 | 1 |
| 1     | 1     | 0     | 0     | 0      | 1      | 0 | 1 |
| 0     | 0     | 1     | 0     | 0      | 0      | 0 | 1 |
| 1     | 0     | 1     | 0     | 0      | 0      | 0 | 1 |
| 0     | 1     | 1     | 0     | 0      | 0      | 0 | 1 |
| 1     | 1     | 1     | 0     | 0      | 0      | 1 | 0 |
| 0     | 0     | 0     | 1     | 0      | 0      | 0 | 1 |
| 1     | 0     | 0     | 1     | 0      | 0      | 0 | 1 |
| 0     | 1     | 0     | 1     | 1      | 0      | 0 | 0 |
| 1     | 1     | 0     | 1     | 1      | 0      | 0 | 1 |
| 0     | 0     | 1     | 1     | 0      | 0      | 0 | 1 |
| 1     | 0     | 1     | 1     | 0      | 0      | 0 | 1 |
| 0     | 1     | 1     | 1     | 0      | 0      | 0 | 1 |
| 1     | 1     | 1     | 1     | 0      | 0      | 0 | 1 |

$Z'_0$ :



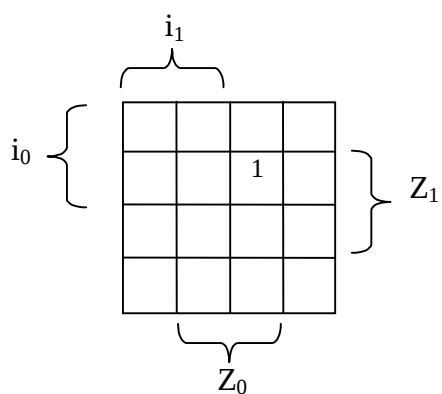
$$Z_1 \bar{i}_0 i_1 + Z_0 \bar{Z}_1 i_0 i_1$$

$Z'_1$ :



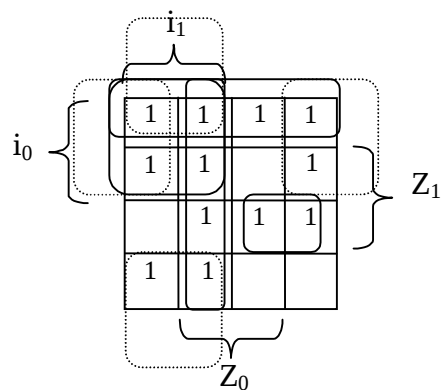
$$\bar{i}_0 \bar{i}_1$$

E:



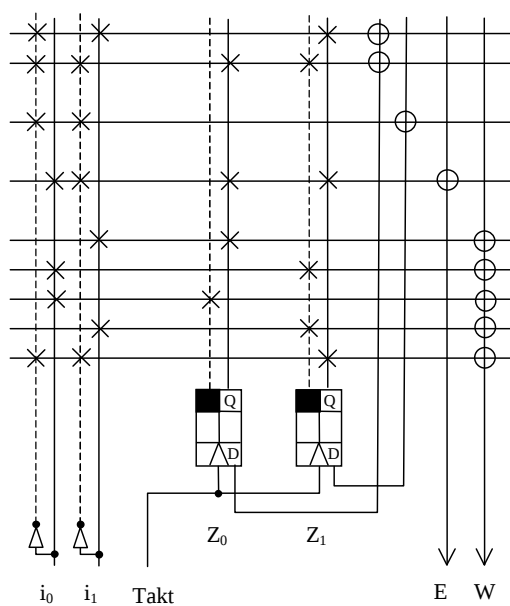
$$Z_0 Z_1 i_0 \bar{i}_1$$

W:



$$Z_0 i_1 + \bar{Z}_1 i_0 + \bar{Z}_0 i_0 + \bar{Z}_1 i_1 + Z_1 \bar{i}_0 \bar{i}_1$$

d) Zeichnen Sie eine Realisierung mit Hilfe eines FPLA mit D-Flipflops



## Aufgabe 2

Schreiben Sie ein Programm für die 4-Bit ALU-Maschine aus der Vorlesung, das einen Eingabewert im Register R0 testet, ob er gerade oder ungerade ist. Wenn er gerade ist, soll das Register R7 auf 1111 gesetzt werden, wenn er ungerade ist auf 0000.

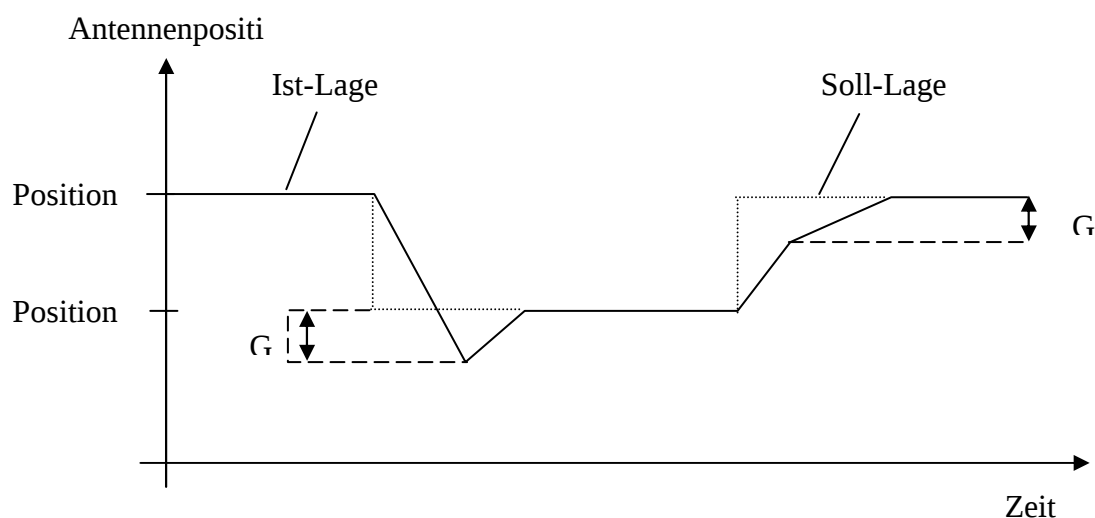
|    | Befehl | Register   | Bedeutung:                                             |
|----|--------|------------|--------------------------------------------------------|
| 1. | SETF   | R2         | R2=1111 (-1)                                           |
| 2. | DEC    | R3, R2     | R3=1110 (-2)                                           |
| 3. | OR     | R4, R3, R0 | R4=1110, falls R0 gerade<br>R4=1111, falls R0 ungerade |
| 4. | SUB    | R7, R3, R4 | R7=0000, falls R0 gerade<br>R7=1111, falls R0 ungerade |

**Serie 10**

**Aufgabe 1**

Ein synchrones Schaltwerk soll den Stellantrieb einer Antenne steuern. Der Antrieb hat zwei Geschwindigkeiten und wird mit den Signalen "auf" und "ab" und "schnell gesteuert". Wenn sich die Antenne von unten bis auf einen festen Grenzwert  $G$  der Soll-Lage genähert hat, darf der Antrieb nur noch langsam verstellt werden. Es steht ein Vergleicher zur Verfügung, der die Differenz  $DL = \text{Ist-Lage} - \text{Soll-Lage}$  ausgewertet und die Signale  $DL > 0$ ,  $DL < 0$  und  $|DL| < G$  ausgibt.

Um das Getriebeispiel auszuschalten, muß die Soll-Lage immer von der gleichen Seite, in diesem Fall von unten, angefahren werden. Soll z.B. die Position 1 in der unten angegebenen Fahrkurve angefahren werden, so ist die Soll-Lage zunächst schnell zu überfahren bevor dann langsam die Zielposition angesteuert wird. Drei Lampen sollen den Zustand des Antriebs anzeigen. Eine "auf"-Lampe soll während des Aufwärtsbewegens der Antenne leuchten, eine "ab"-Lampe während des Abwärtsbewegens. Wenn die Signale des Vergleichers widersprüchlich sind, dann soll eine "Fehler"-Lampe aufleuchten und die Antenne nicht mehr bewegt werden. Dies soll auch dann so bleiben, wenn der Widerspruch zwischenzeitlich verschwindet. Die "Fehler"-Lampe soll erst dann wieder abgeschaltet und die normale Operation wieder aufgenommen werden, wenn eine Resettaste gedrückt wird. Weiterhin soll es dem Bedienpersonal möglich sein über die Anzeige der "auf"- und "ab"-Lampen festzustellen, ob die Antenne beim Auftreten des Fehlers in einer Aufwärts-, einer Abwärtsbewegung oder im Stillstand war.

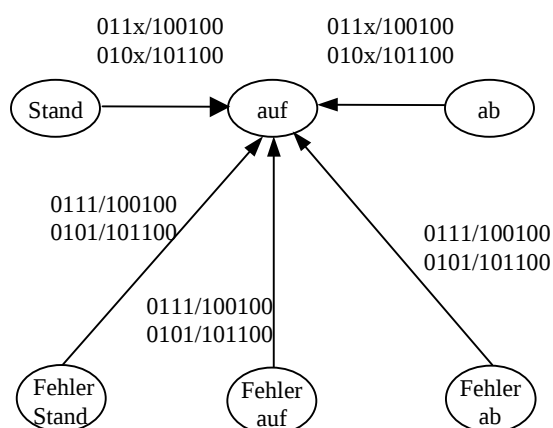
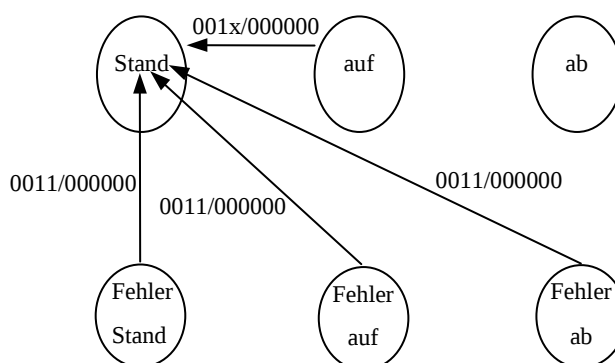
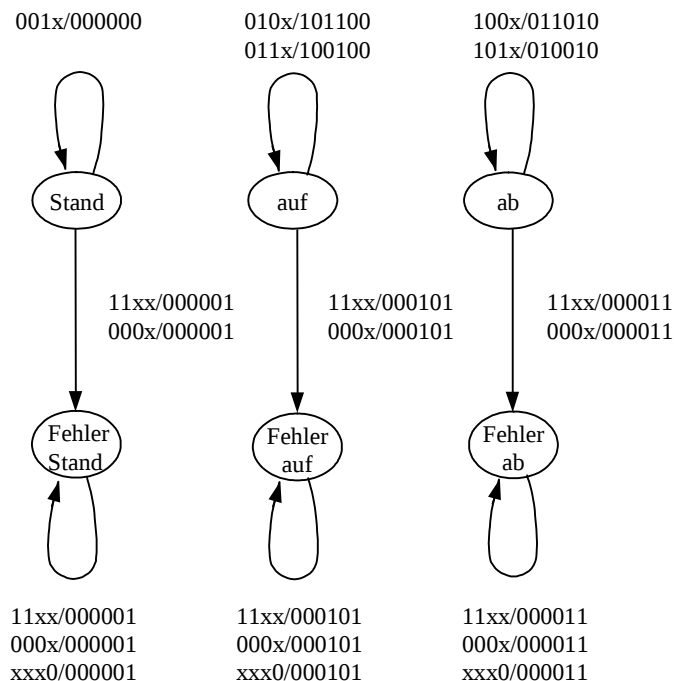


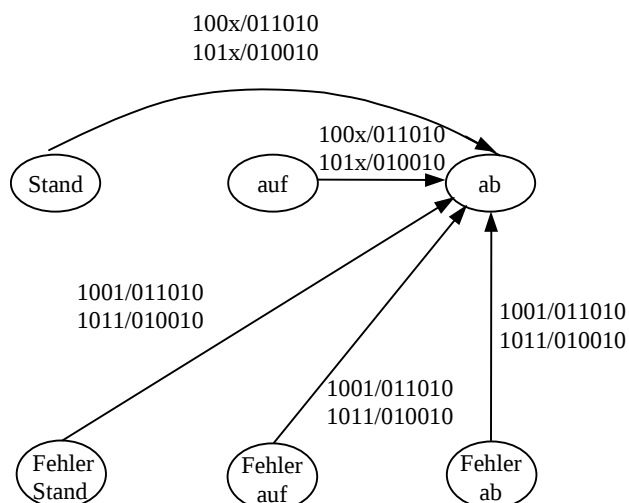
- a) Geben Sie die Eingangs- und Ausgangssignale des Schaltwerks an.  
Eingangssignale:  $DL > 0$ ,  $DL < 0$ ,  $|DL| < G$  und Reset

Ausgangssignale: aufwärts, abwärts, schnell, 1\_auf, 1\_ab, 1\_fehler

b) Entwerfen Sie den Mealy-Automatengraphen für das Schaltwerk.

Die Reihenfolge der Eingangs- und Ausgangssignale an den Transitionen entspricht der obigen.



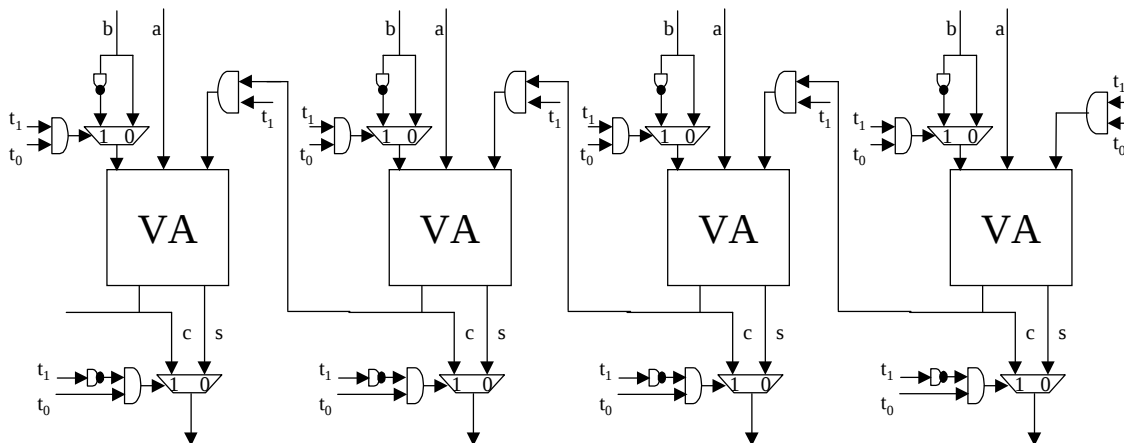


## Aufgabe 2

Entwerfen Sie durch eine geringfügige Modifizierung eines Ripple Carry Addierers eine 4 Bit ALU, die folgende vier Befehle ausführen kann:

XOR, AND, ADD und SUB

Lösung:



Kodierung:

|    |     |
|----|-----|
| 00 | XOR |
| 01 | AND |
| 10 | ADD |
| 11 | SUB |

Lösungen zur Hausaufgabe zur Vorlesung  
Digitale Systeme

WS 2004/2005  
Prof. Dr. rer. nat. Manfred Schimmmler  
Dipl.-Inform. Viktor Bunimov  
Dipl.-Ing. Gerd Pfeiffer  
Dipl.-Ing. Heinz Kreft

### Serie 11

#### Aufgabe 1

20% der Befehle eines Programms haben die CPI=1; 30 % haben CPI= 2; 30 % haben CPI=5. Die restlichen Befehle haben CPI=10.

Wie hoch ist die Ausführungszeit eines Programms mit  $25 \cdot 10^6$  Befehlen auf einem mit 2GHz getakteten Prozessor?

$$CPI = 0,2 \cdot 1 + 0,3 \cdot 2 + 0,3 \cdot 5 + (1 - 0,2 - 0,3 - 0,3) \cdot 10 = 4,3;$$

$$\text{Zykluszeit} = 1 / \text{Taktrate};$$

$$\text{CPU-Zeit} = IC \cdot CPI \cdot \text{Zykluszeit} = 25 \cdot 10^6 \cdot 4,3 \cdot 1 / (2 \cdot 10^9) = 0,05375 \text{ Sekunden.}$$

#### Aufgabe 2

Die durchschnittliche Geschwindigkeit eines Fußgängers liegt bei 6 km/Stunde. Ein Radfahrer fährt mit 18 km/Stunde. Ein Autofahrer fährt 60 km/Stunde. Eine Rakete kann unendlich schnell fliegen. Ein Reisender muss von Punkt X nach Punkt Y kommen. Sein Fahrrad, sein Auto und seine Rakete stehen 3 km entfernt vom Punkt X. Der Abstand zwischen X und Y beträgt 9 km. Das folgende Bild veranschaulicht die Situation:



Welchen Speedup erreicht der Reisende wenn er

- a) das Fahrrad
  - b) das Auto
  - c) die Rakete
- benutzen darf gegenüber dem Fußweg?

$$\text{Speedup} = \frac{1}{A_u + \frac{1 - A_u}{\text{Anteilbeschleunigt}}}$$

$$\text{a) Speedup} = \frac{1}{3/9 + \frac{1 - 3/9}{18/6}} = 1,8$$

$$b) \text{ Speedup} = \frac{1}{3/9 + \frac{1-3/9}{60/6}} = 2,5$$

$$c) \text{ Speedup} = \lim_{n \rightarrow \infty} \frac{1}{3/9 + \frac{1-3/9}{n/6}} = \frac{1}{\lim_{n \rightarrow \infty} (3/9 + \frac{1-3/9}{n/6})} = \frac{1}{3/9 + 0} = 3$$

### Aufgabe 3

Liest ein Rechner die Daten von der Festplatte, so besteht ein solcher Vorgang aus zwei Teilen: Die Positionierung des Lese-Schreib-Kopfes und dem Lesen der Daten. Die Zeit für die Positionierung des Lese-Schreib-Kopfes der Festplatte beträgt 1 ms. Beim Lesevorgang wird die Datenrate von 32 Mbit/s erreicht.

- a) Wie viel Zeit wird benötigt um eine 100 KByte große Datei zu lesen?

$$\text{Zeit} = \text{Zeit}_{\text{Pos}} + \text{Zeit}_{\text{Les}} = 1\text{ms} + 100\text{kByte}/32\text{Mbit/s} = [10^{-3} + 8 \cdot 10^5 / (32 \cdot 10^6)]\text{s} = 0,026\text{s} = 26\text{ms};$$

- b) Wie hoch ist der Speedup, wenn die Datenrate verdoppelt wird? Die Datengröße ist wieder 100 KByte.

$$\text{Speedup} = \frac{1}{A_u + \frac{1-A_u}{\text{Anteilbeschleunigt}}} = \frac{1}{\frac{1}{26} + \frac{1-1/26}{2}} = \frac{52}{27} = 1.9259$$

- c) Wie hoch muss die Datenrate sein, damit die Festplatte aus b) gegenüber der ursprünglichen Festplatte den Speedup von 2 erzielt? Benutzen Sie Amdals Gesetz.

$$\text{Speedup} = \frac{1}{A_u + \frac{1-A_u}{\text{Anteilbeschleunigt}}} = \frac{1}{\frac{1}{26} + \frac{1-1/26}{x}} = 2$$

$$\Rightarrow x = 50 / 24 = 2,08(3)$$

Die Datenrate liegt bei  $2,08(3) \cdot 32\text{Mbit/s} = 66,6(6) \text{ Mbit/s}$

### Aufgabe 4

Ein Prozessor besitzt einen Level 1-Cache und einen Level 2-Cache. Die Hit-Zeit für den Level 1-Cache ist 1 Taktzyklus. Die Miss-Rate des Level 1-Caches ist 10%. Die Hitzeit des L2-Caches beträgt 5 Zyklen. Der L2-Cache hat bei 50 Zugriffen durchschnittlich 1 Fehlzugriff. Die Miss Penalty des L2-Caches beträgt 70 Zyklen.

*Zugriffszeit = Hit Zeit + Miss Rate \* Miss Penalty*

- a) Geben Sie die durchschnittliche Zugriffszeit des Systems bei einer Taktfrequenz von 2GHz an.

$$\text{Zugriffszeit} = \frac{1 + 0,1 * (5 + \frac{50 - 49}{50} * 70)}{2 * 10^9} = \frac{1,64}{2 * 10^9} = 0,82 \text{ ns}$$

- b) Geben Sie eine allgemeine Cache-Performance-Gleichung für ein solches L1-L2-Cache-System an.

$$\text{Zugriffszeit} = \text{Hit Zeit L1} + \text{Miss Rate L1} * (\text{Hit Zeit L2} + \text{Miss Rate L2} * \text{Miss Penalty L2})$$

- c) Wie würde diese Gleichung bei einem dreistufigen L1-L2-L3-Cache-System aussehen?

$$\begin{aligned} \text{Zugriffszeit} = & \text{Hit Zeit L1} + \text{Miss Rate L1} \\ & * (\text{Hit Zeit L2} + \text{Miss Rate L2} \\ & * (\text{Hit Zeit L3} + \text{Miss Rate L3} * \text{Miss Penalty L3})) \end{aligned}$$

**Serie 12**

**Aufgabe 1**

Schreiben Sie ein DLX-Assemblerprogramm, welches drei 32-Bit Integerzahlen, die nacheinander ab der Adresse 1064 im Hauptspeicher abgelegt sind, aufsteigend sortiert.

|           |      |              |                                |
|-----------|------|--------------|--------------------------------|
| start:    | LW   | R1,1064(R0)  | Einlesen der Zahlen aus dem    |
|           | LW   | R2,1068(R0)  | Hauptspeicher in die           |
|           | LW   | R3,1072(R0)  | Prozessorregister R1,R2 und R3 |
|           | SLE  | R4,R1,R2     | Regs[R1] ≤ [R2]?               |
|           | BNEZ | R4, weiter   |                                |
|           | ADD  | R4,R1,R0     | falls nicht, vertausche        |
|           | ADD  | R1,R2,R0     | die Inhalte der                |
|           | ADD  | R2,R4,R0     | Register R1 und R2             |
| weiter:   | SLE  | R4,R2,R3     | Regs[R2] ≤ [R3]?               |
|           | BNEZ | R4, weiter_2 |                                |
|           | ADD  | R4,R2,R0     | falls nicht, vertausche        |
|           | ADD  | R2,R3,R0     | die Inhalte der                |
|           | ADD  | R3,R4,R0     | Register R2 und R3             |
| weiter_2: | SLE  | R4,R1,R2     | Regs[R1] ≤ [R2]?               |
|           | BNEZ | R4, sortiert |                                |
|           | ADD  | R4,R1,R0     | falls nicht, vertausche        |
|           | ADD  | R1,R2,R0     | die Inhalte der                |
|           | ADD  | R2,R4,R0     | Register R1 und R2             |
| sortiert: | SW   | 1064(R0),R1  | schreibe sortierte             |
|           | SW   | 1068(R0),R2  | Zahlen zurück                  |
|           | SW   | 1072(R0),R3  | in den Hauptspeicher           |

## Aufgabe 2

Schreiben Sie ein DLX-Assemblerprogramm, welches die Anzahl der Einsen im Inhalt des Registers R3 ermittelt und das Ergebnis im Register R2 ablegt. Der Inhalt von R3 soll dabei nicht verändert werden.

### DLX-Programm

Anzahl der Einsen in Regs[R2] bestimmen:

|        |        |           |                                                                         |
|--------|--------|-----------|-------------------------------------------------------------------------|
| start: | ADD    | R2,R0,R0  | Anzahl der Einsen = Regs[R3]=0                                          |
|        | ADD    | R4,R3,R0  | Regs[R4]=Regs[R2]                                                       |
|        | ADDI   | R5,R0,#32 | Zähler=Regs[R5]=32                                                      |
| loop:  | ANDI   | R6,R4,#1  | Regs[R6] = 1 falls niederwertigstes Bit von Regs[R4]=1 Regs[R6]=0 sonst |
|        | ADD    | R2,R2,R6  | Regs[R3] wird um eins erhöht, falls Regs[R6]=1                          |
|        | SRLI   | R4,R4,#1  | Regs[R4]um ein Bit nach rechts schieben                                 |
|        | SUBI   | R5,R5,#1  |                                                                         |
|        | BNEZ   | R5,loop   |                                                                         |
| halt:  | fertig |           |                                                                         |

oder effizienter:

|        |        |          |                                                                         |
|--------|--------|----------|-------------------------------------------------------------------------|
| start: | ADD    | R2,R0,R0 | Anzahl der Einsen = Regs[R3]=0                                          |
|        | ADD    | R4,R3,R0 | Regs[R4]=Regs[R2]                                                       |
| loop:  | ANDI   | R6,R4,#1 | Regs[R6] = 1 falls niederwertigstes Bit von Regs[R4]=1 Regs[R6]=0 sonst |
|        | ADD    | R2,R2,R6 | Regs[R3] wird um eins erhöht, falls Regs[R6]=1                          |
|        | SRLI   | R4,R4,#1 | Regs[R4]um ein Bit nach rechts schieben                                 |
|        | BNEZ   | R4,loop  | (von links werden dabei in R4 Nullen nachgeschoben)                     |
| halt:  | fertig |          |                                                                         |

### Aufgabe 3

- a) Schreiben Sie ein DLX-Assemblerprogramm, welches das Minimum einer Folge von  $n$  natürlichen Zahlen bestimmt. Sie sollen dabei von folgenden Voraussetzungen ausgehen. Das Register R2 enthält die Zahl  $n$ . Das Register R1 enthält die Adresse der ersten Zahl in Folge. Alle Zahlen der Folge sind aufeinanderfolgend im Speicher abgelegt.

Das Programm in Pseudocode:

Eingabe:  $z[1], z[2], \dots, z[n]$ ,

minimum =  $z[1]$ ;

for ( $i=2; i \leq n; i++$ ) {

    if ( $z[i] < \text{minimum}$ ) minimum =  $z[i]$ ;

}

#### DLX-Programm

Voraussetzungen  $\text{Regs}[R1]=n$ ,  $\text{Regs}[R2]=\text{Anfangsadresse von } z[1]$

|        |        |          |                                                                                |
|--------|--------|----------|--------------------------------------------------------------------------------|
| start: | LW     | R3,0(R1) | $\text{Regs}[R3]=z[1]\text{minimum}$                                           |
|        | ADD    | R4,R1,R0 | $\text{Regs}[R4]=\text{Anfangsadresse der aktuellen Zahl}$                     |
|        | ADD    | R5,R2,R0 | $\text{Regs}[R5]=\text{Zähler für Zahlenindex } i=n$                           |
| loop:  | SUBI   | R5,R5,#1 |                                                                                |
|        | BEQZ   | R5,halt  |                                                                                |
|        | ADDI   | R4,R4,#4 | $\text{Regs}[R4]=\text{Anfangsadresse der aktuellen Zahl}$                     |
|        | LW     | R6,0(R4) | $\text{Regs}[R6]=\text{Mem}[\text{Regs}[R4]]=\text{aktuelle Zahl}$             |
|        | SLE    | R7,R3,R6 | $\text{Regs}[R7]=1 \text{ falls } \text{Regs}[R3] \leq \text{Regs}[R6]$        |
|        |        |          | $\text{Regs}[R7]=0 \text{ falls } \text{Regs}[R3] > \text{Regs}[R6]$           |
|        | BNEZ   | R7,loop  | altes Minimum bleibt erhalten,<br>falls $\text{Regs}[R3] \leq \text{Regs}[R6]$ |
|        | ADD    | R3,R0,R6 | sonst neues Minimum $\text{Regs}[R3]=\text{Regs}[R6]$                          |
|        | J      | loop     |                                                                                |
| halt:  | fertig |          |                                                                                |

- b) Erweitern Sie Ihr Programm für a) so, dass die Folge aufsteigend sortiert wird.

Eingabe:  $z[1], z[2], \dots, z[n]$ ,

for ( $i=1; i \leq n-1; i++$ ) {

    Bestimme  $j \in \{i, i+1, \dots, n\}$  mit

$z[j] = \min\{z[i], \dots, z[n]\}$ ;

$h = z[i]$ ;

$z[i] = z[j]$ ;

$z[j] = h$ ;

}

# DLX-Programm

Voraussetzungen Regs[R2]=n, Regs[R1]=Anfangsadresse von z[1]

|        |        |            |                                                                          |
|--------|--------|------------|--------------------------------------------------------------------------|
| start: | ADD    | R8,R2,R0   | Regs[R8]=n                                                               |
|        | ADD    | R9,R1,R0   | Regs[R9]=Anfangsadresse der Folge                                        |
| loop1: | ADD    | R4,R9,R0   | Regs[R4]=Anfangsadresse der noch zu sortierenden Restfolge z[i],...,z[n] |
|        | ADD    | R5,R8,R0   | Regs[R5]=Anzahl der Zahlen in der Restfolge =n-1                         |
|        | LW     | R3,0(R9)   | Regs[R3]=z[i]=minimum                                                    |
|        | ADD    | R10,R4,R0  | Regs[R10]=Adresse des aktuellen Minimums                                 |
| loop2: | SUBI   | R5,R5,#1   |                                                                          |
|        | BEQZ   | R5,next    |                                                                          |
|        | ADDI   | R4,R4,#4   |                                                                          |
|        | LW     | R6,0(R4)   |                                                                          |
|        | SLE    | R7,R3,R6   | Minimum der Folge z[i],...,z[n]                                          |
|        | BNEZ   | R7,loop2   | und dessen Adresse bestimmen                                             |
|        | ADD    | R3,R0,R6   |                                                                          |
|        | ADD    | R10,R4,R0  |                                                                          |
|        | J      | loop2      |                                                                          |
| next   | LW     | R11,0(R9)  | Regs[R11]=z[i]                                                           |
|        | SW     | 0(R10),R11 | Mem[Regs[R10]]=Regs[R11]                                                 |
|        | SW     | 0(R9),R3   | z[i]=minimum                                                             |
|        | ADDI   | R9,R9,#4   | i=i+1                                                                    |
|        | SUBI   | R8,R8,#1   |                                                                          |
|        | BNEZ   | R8,loop1   |                                                                          |
| halt:  | fertig |            |                                                                          |