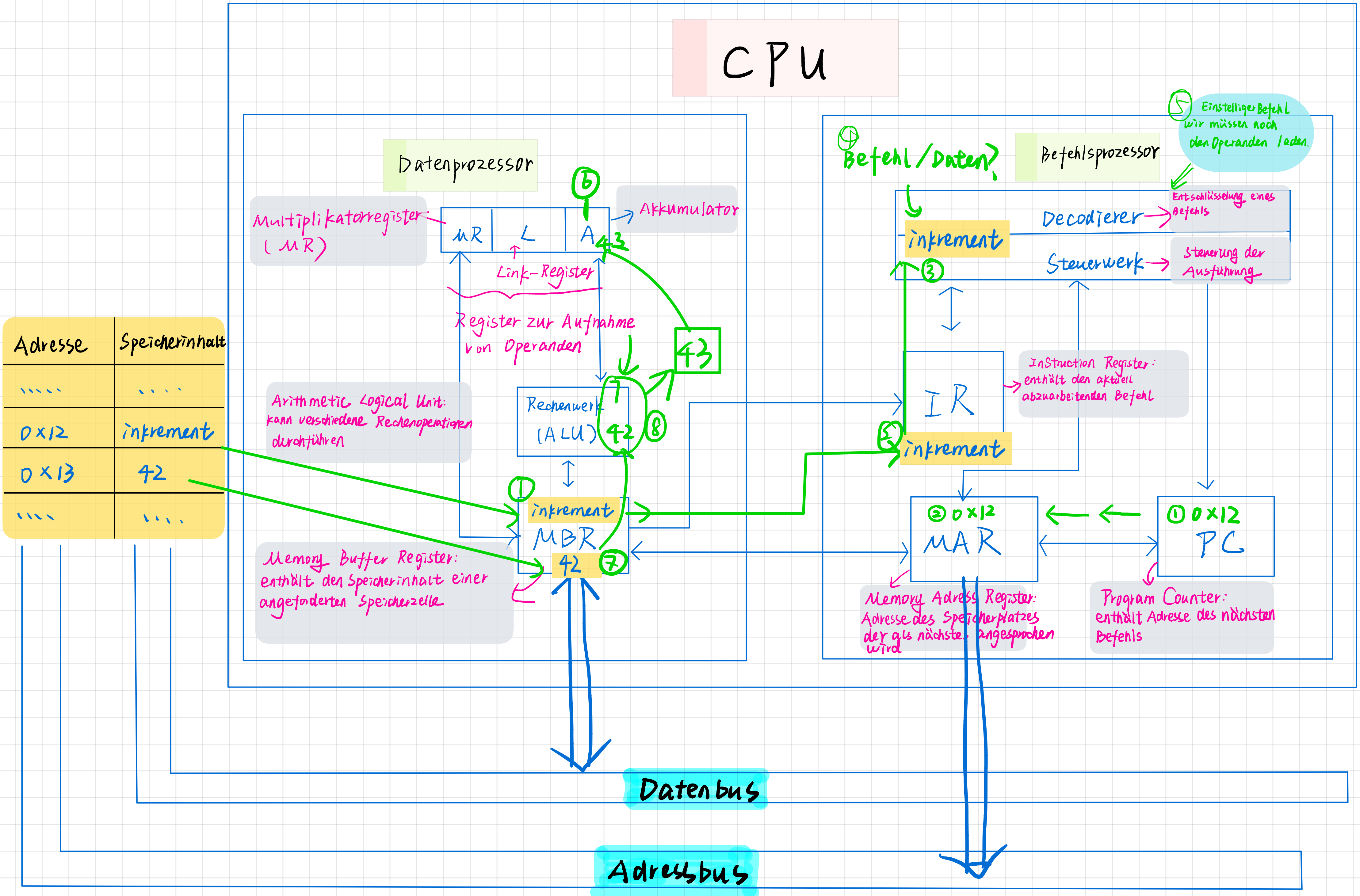

CPU
Datenprozessor
Multiplikatorregister (MR)
MR
L
A
Akkumulator
43
Link-Register
Register zur Aufnahme von Operanden
Arithmetic Logical Unit: kann verschiedene Rechenoperationen durchführen
Rechenwerk (ALU)
42
inkrement
MBR
42
Memory Buffer Register: enthält den Speicherinhalt einer angeforderten Speicherzelle
Befehl/Daten?
Befehlsprozessor
Einstelliger Befehl wir müssen noch den Operanden laden.
Decodierer
Entschlüsselung eines Befehls
Steuerwerk
Steuerung der Ausführung
inkrement
IR
Instruction Register: enthält den aktuell abzuarbeitenden Befehl
inkrement
0x12
MAR
0x12
PC
Memory Adress Register: Adresse des Speicherplatzes der als nächstes angesprochen wird
Program Counter: enthält Adresse des nächsten Befehls
Adresse
Speicherinhalt
.....
....
0x12
inkrement
0x13
42
....
....
Datenbus
Adressbus

Dazu kommen noch Verbindungen zwischen diesen Einheiten, sogenannte Busse (siehe Ab-

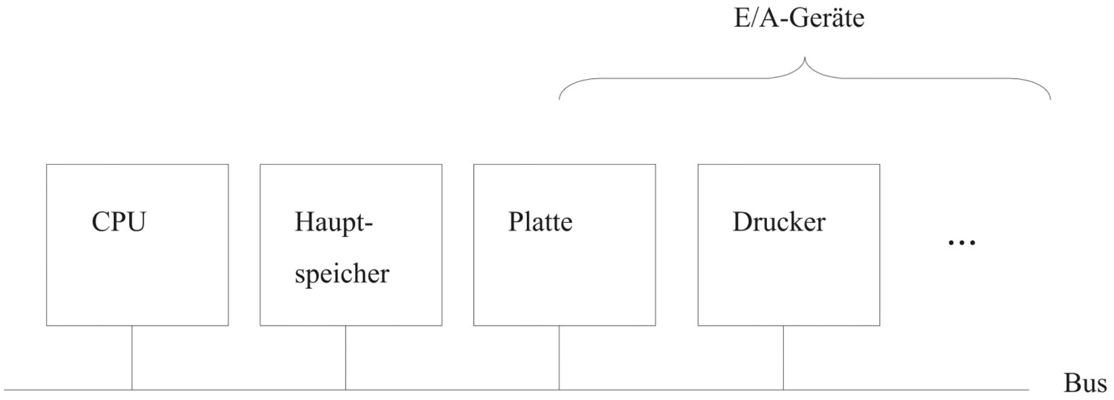



Abbildung 3.1: Komponenten eines Computers

bildung 3.1).

Diese Bestandteile wollen wir im folgenden genauer betrachten.

## Darstellung von Informationen ⇒ (Wahrheitswerte)

Menge B von Wahrheitswerten aus genau 2 Elementen bestehen: {wahr, falsch} / {1, 0}

Solche Informationen können vom Rechner gespeichert werden.

Die einfachste und fundamentalste Art der Darstellung von Informationen sind sogenannte **Wahrheitswerte**. An späterer Stelle werden wir formaler die Boolesche Algebra einführen. Im folgenden soll eine Menge B von Wahrheitswerten aus genau zwei Elementen bestehen: {wahr, falsch} oder {1, 0} oder {L, 0} oder ...
Solche Informationen können vom Rechner gespeichert werden. Dabei wollen wir eine Struktur zugrunde legen, bei der 8 Elemente, sogenannte Bits (Binary Digits) zu einer Information zusammengefaßt werden, zu einem **Byte**. Beispiele für Bytes sind 00000000 oder 01010101 oder auch 01000110.

Den beiden Zuständen eines Bits ordnet man die Werte 0 bzw. 1 zu. Jeder Zustand eines Bits kann unabhängig von den Zuständen der anderen Bits variieren. Damit lassen sich aus den 8 Bits eines Bytes $2^8 = 256$ verschiedene Bitmuster bilden.

Sollte man alle Bitmuster aufschreiben, so empfiehlt sich ein systematisches Vorgehen, bei dem man das letzte Bit alterniert und bei Erschöpfung aller Bitmuster das Bit davor betrachtet.

| | | | |
|---|---|---|---|
| 0 | 0 0 | 0 00 | 0 000 |
| 1 | 0 1 | 0 01 | 0 001 |
| | 1 0 | 0 10 | 0 010 |
| | 1 1 | 0 11 | 0 011 |
| | | 1 00 | 0 100 |
| | | 1 01 | 0 101 |
| | | 1 10 | 0 110 |
| | | 1 11 | 0 111 |
| | | | 1 000 |
| | | | 1 001 |
| | | | 1 010 |
| | | | 1 011 |
| | | | 1 100 |
| | | | 1 101 |
| | | | 1 110 |
| | | | 1 111 |
| $2^1 = 2$ Bitmuster bei $n = 1$ Stelle | $2^2 = 4$ Bitmuster bei $n = 2$ Stellen | $2^3 = 8$ Bitmuster bei $n = 3$ Stellen | $2^4 = 16$ Bitmuster bei $n = 4$ Stellen |

## 不同单位之间的转换

**Kilobyte:** 1 KB = 1024 Byte = $2^{10}$ Byte = 8.192 Bit ≈ $10^3$ Byte

**Megabyte:** 1 MB = 1024 KB = $2^{20}$ Byte = 1.048.576 Byte = 8.388.608 Bit ≈ $10^6$ Byte ≈ $10^6$ Byte

**Gigabyte:** 1 GB = 1024 MB = $2^{30}$ Byte =1.048.576 KB = 1.073.741.824 Byte = 8.589.934.592 Bit ≈ $10^9$ Byte

**Terabyte:** 1 TB = 1024 GB = $2^{40}$ Byte ≈ $10^{12}$ Byte

**Petabyte:** 1 PB = 1024 TB = $2^{50}$ Byte ≈ $10^{15}$ Byte

## 怎么把十进制转二,十六,八进制

以 255 为例

binär

1111 1111

2 | 255 1
2 | 127 1
2 | 63 1
2 | 31 1
2 | 15 1
2 | 7 1
2 | 3 1
2 | 1 1
0

okt. 8 | 255 7
8 | 31 7
8 | 3 3
0

okt. 377

16 | 255 15
16 | 15 15
0

hex. FF

## 怎么把十六进制 → 十进制

1. 首先，识别十六进制数中的每一位。从右向左（从低位到高位）给每一位赋一个索引值，最右边的位（最低位）的索引为0，向左的每一位索引值依次增加。
2. 对每一位进行处理：将该位的值（如果是A到F，则转换为10到15）乘以16的对应索引次方（例如，第0位乘以16的0次方，第1位乘以16的1次方，以此类推）。
3. 将所有处理过的位的值加起来，得到的总和就是十进制的表示。

例如，我们来转换十六进制数2A3（这是一个3位的十六进制数）：

- 第0位是3，所以计算为3 * (16^0) = 3 * 1 = 3
- 第1位是A，代表10，所以计算为10 * (16^1) = 10 * 16 = 160
- 第2位是2，所以计算为2 * (16^2) = 2 * 256 = 512

将这些值加起来，我们得到3 + 160 + 512 = 675，所以十六进制的2A3等于十进制的675。

Digital speichern → verarbeiten → übertragen (Fehlererkennung, Fehler korrektur)

Was kann man digital abspeichern?

- Ziffern
- Zahlen
- Buchstaben
- Text
- Bilder
- Audio
- ....

Wie speichern?

Signal

0 / falsch / negativ (kein Strom übertragen)

1 / wahr / positiv (Strom übertragen)

## 7.1 Boolesche Algebra

- OR-Operator $+$ oder $\vee$ (logische Summe)
- AND-Operator $*$ oder $\wedge$ (logische Produkt)
- NOT-Operator $\overline{a}$ oder $\neg a$ (Invertierung)

Kommutativgesetz: $a \vee b = b \vee a$ und $a \wedge b = b \wedge a$

Assoziativgesetz: $(a \vee b) \vee c = a \vee (b \vee c)$ und $(a \wedge b) \wedge c = a \wedge (b \wedge c)$

Distributivgesetz: $a \wedge (b \vee c) = (a \wedge b) \vee (b \wedge c)$ und $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$

Identität~ : $a \vee 0 = a$ und $a \wedge 1 = a$

Null- und Eins-Gesetz $a \wedge 0 = 0$ und $a \vee 1 = 1$

Komplement~ $a \vee \overline{a} = 1$ und $a \wedge \overline{a} = 0$

Verschmelzungs~ $(a \vee b) \wedge a = a$ und $(a \wedge b) \vee a = a$

de Morgansche Regeln: $\overline{a \vee b} = \overline{a} \wedge \overline{b}$ und $\overline{a \wedge b} = \overline{a} \vee \overline{b}$

Boolesche Funktion: $f: B^n \to B$   n: Stelligkeit

Wie viele einstellige Boolesche Funktionen gibt?

| a | $f_1$ | $f_2$ | $f_3$ | $f_4$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |

$B^2 \to B$   (4 ↑ einstellige Bu Fuktionen)

Jede beliebige Funktion

Im Fall $n = 2$ kommt man auf 16 zweistellige Boolesche Funktionen, insbesondere gehören dazu auch AND und OR.

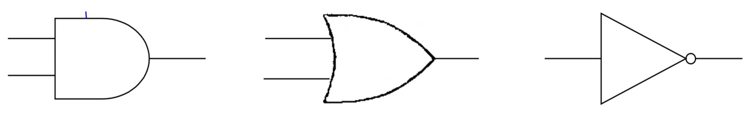
| $x$ | $y$ | 0 | AND | $x\bar{y}$ | $x$ | $\bar{x}y$ | $y$ | $\nleftrightarrow$ | OR | NOR | $=$ | $\bar{y}$ | $\overline{\bar{x}y}$ | $\bar{x}$ | $\overline{x\bar{y}}$ | NAND | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| | | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ |

Allgemein gilt, dass es für jedes beliebige $n \in \mathbb{N}$ mit $n \geq 1$ genau $2^{2^n}$ n–stellige Boolesche Funktionen gibt.

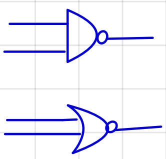
$2^{2n}$ ↑

## 7.2.1 Gatter

$B^n \to B^7$

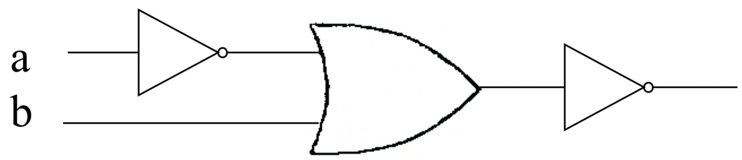


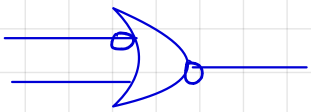
AND   OR   NOT




$\overline{a \vee b}$   NOR

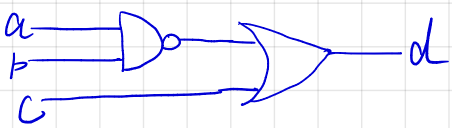
$\overline{\bar{a} + b}$



$\overline{(a \vee b)} \wedge c$

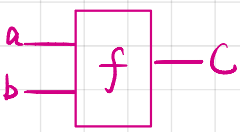
$d = f(a, b, c)$




$f: B^3 \to B$

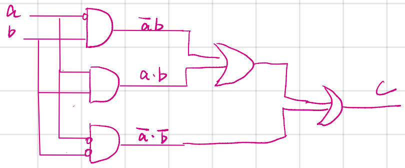
$f: B^2 \to B$

$f(a, b) = c$

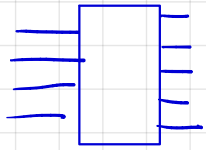


$C := (\bar{a} \wedge b) \vee (a \wedge b) \vee (\bar{a} \wedge \bar{b})$

$= \bar{a} \cdot b \vee a b \vee \bar{a} \cdot \bar{b}$




mehr Ausgänge —— Schaltfunktion

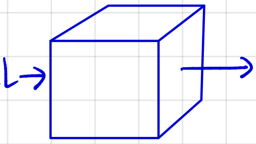
n  m

n Eingänge

m Ausgänge

$F: B^n \to B^m$

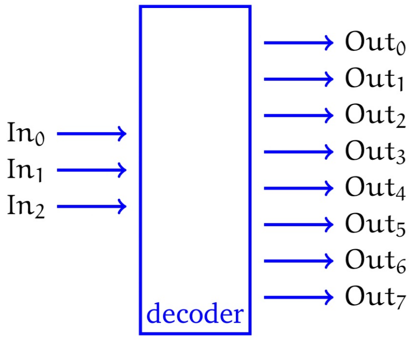
## 7.2.2 Decoder

Dualzahl →  → Dezimalzahl

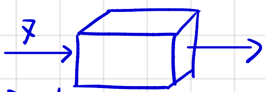
(011) (3)

n Eingänge

$2^n$ Ausgänge




## 7.2.3 Encoder

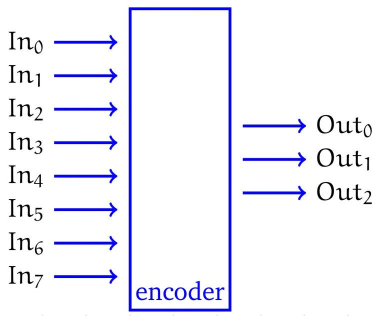
7 →  → 111

Dezimal Dual

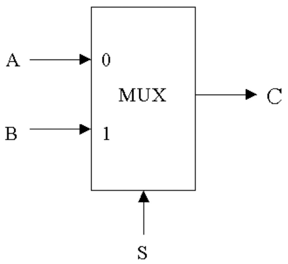
8 Input $I_0 \cdots I_7$

$2^n$ Eingänge

n Ausgänge




## 7.2.4 Multiplexer

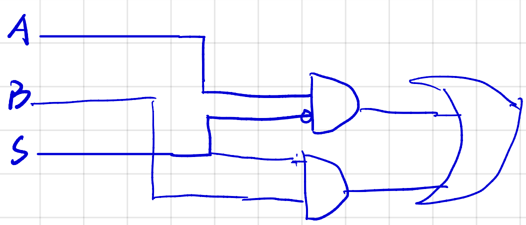



$C = (A \cdot \bar{S}) + (B \cdot S)$

3 stellig $f: B^3 \to B$

2 Nutzeingabe

1 Steuereingabe




比如规定$S=0$时 C端输出A的值

$S=1$时 C端输出B的值

Bei einem 4-Eingaben-Multiplexer haben wir folgendes Prinzip (vgl. Abbildung 7.13):

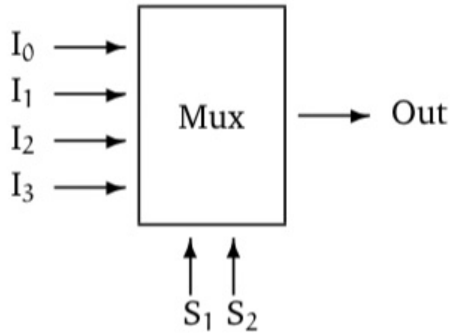



Abbildung 7.13: 4-Eingaben-Multiplexer

Die Funktionstabelle lautet in Kurzform:

| $S_1$ | $S_2$ | Out |
|---|---|---|
| 0 | 0 | $I_0$ |
| 0 | 1 | $I_1$ |
| 1 | 0 | $I_2$ |
| 1 | 1 | $I_3$ |

Die Funktionalität kann durch folgende Boolesche Funktion beschrieben werden:

$$\text{Out} = I_0 \cdot \overline{S_1} \cdot \overline{S_2} + I_1 \cdot \overline{S_1} \cdot S_2 + I_2 \cdot S_1 \cdot \overline{S_2} + I_3 \cdot S_1 \cdot S_2$$

Die zugehörige Boolesche Funktion ist dabei 6-stellig.

Bei einem 3-Eingaben-Multiplexer würden auch 2 Steuereingänge benötigt werden, wobei eine Bitmusterkombination der Steuerwerte nicht benötigt werden würde. Die zugehörige Boolesche Fünktion wäre 5-stellig.

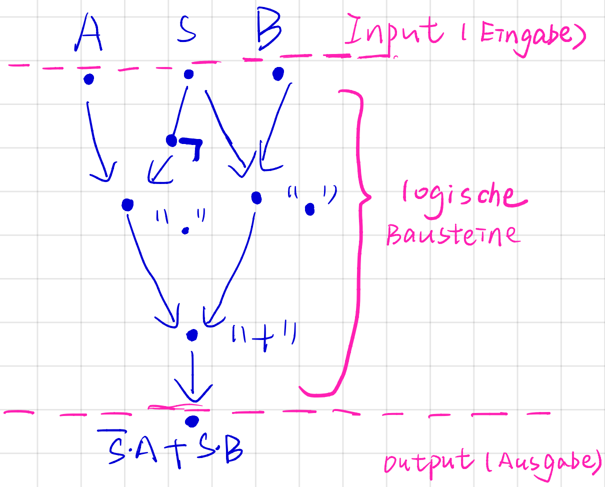
Nun wollen wir den allgemeinsten Fall eines Multiplexers mit $n$ Eingaben (im Sinne von Nutzeingaben) und $\lceil \log_2 n \rceil$ Steuer- oder Selektereingaben anschauen.

Dann besteht ein Multiplexer aus drei Teilen (vgl. Abbildung 7.14).

# 7.3 Grundlagen der Schaltnetze

(Directed Acyclic Graph)

$$C = \overline{S} \cdot A + S \cdot B$$




Realisierung mit geringen KOSTEN

- Zeit (schnell) → kurze Zeit
- Raum (klein) → wenig Materiall

## 7.3.1

### Minterm

$$m_i(x_1 \cdots x_n) = x_1^{i_1} \cdot x_2^{i_2} \cdot x_3^{i_3} \cdots x_n^{i_n}$$

| i | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| ... | | | |

i-ter Minterm (Zeile)

$$x_j^{i_j} = \begin{cases} x_j & \text{falls } i_j = 1 \\ \overline{x_j} & \text{falls } i_j = 0 \end{cases}$$

j-te Stelle der boolesche Funktion (Spalte)

$m_3(x_1, x_2, x_3)$
$= \overline{x_1} \cdot x_2 \cdot x_3$

### Maxterm

$$M_i(x_1, \cdots, x_n) = x_1^{i_1} + x_2^{i_2} + \cdots x_n^{i_n}$$

| i | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| ... | | | |

i-ter Minterm (Zeile)

$$x_j^{i_j} = \begin{cases} x_j & \text{falls } i_j = 0 \\ \overline{x_j} & \text{falls } i_j = 1 \end{cases}$$

j-te Stelle der boolesche Funktion (Spalte)

$M_3(x_1, x_2, x_3)$
$= x_1 + \overline{x_2} + \overline{x_3}$

# Einschlägiger Index

Wenn $f(x_1 \cdots x_n) = 1$ gilt, ist $i$ einschlägiger Index.

Jede boolesche Funktion $f: B^n \to B$ ist eindeutig darstellbar als

- Summe der Minterme ihrer einschlägiger Indizes
- Produkt der Maxterme ihrer nicht einschlägiger Indizes.

# DNF & KNF

Anzahl einschlägiger Indizes < Anzahl nicht einschlägige Indizes → DNF

Anzahl einschlägiger Indizes > Anzahl nicht einschlägige Indizes → KNF

# PLA

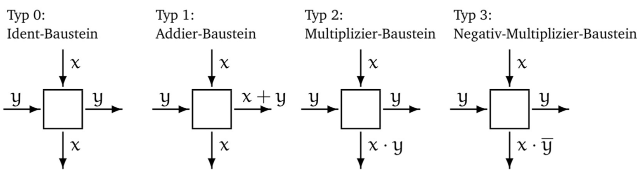
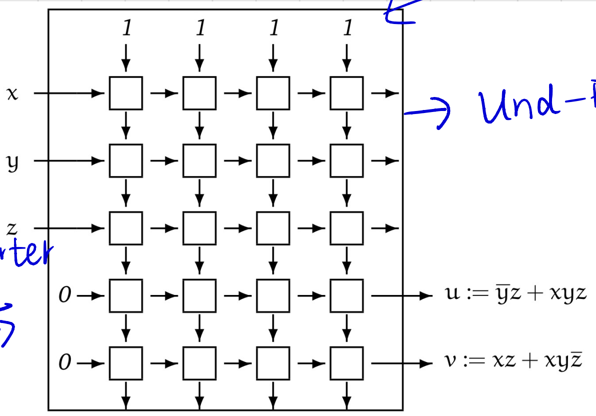



Abbildung 7.18: verschiedene Gitterpunkt-Typen

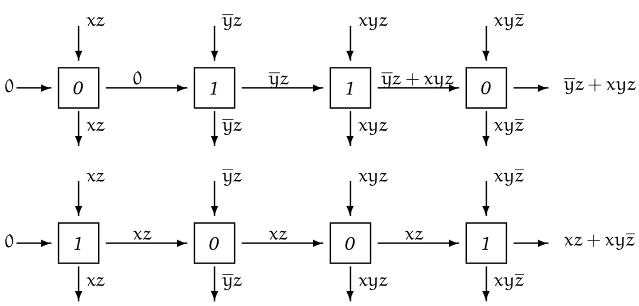
Neutralisierte Eingänge



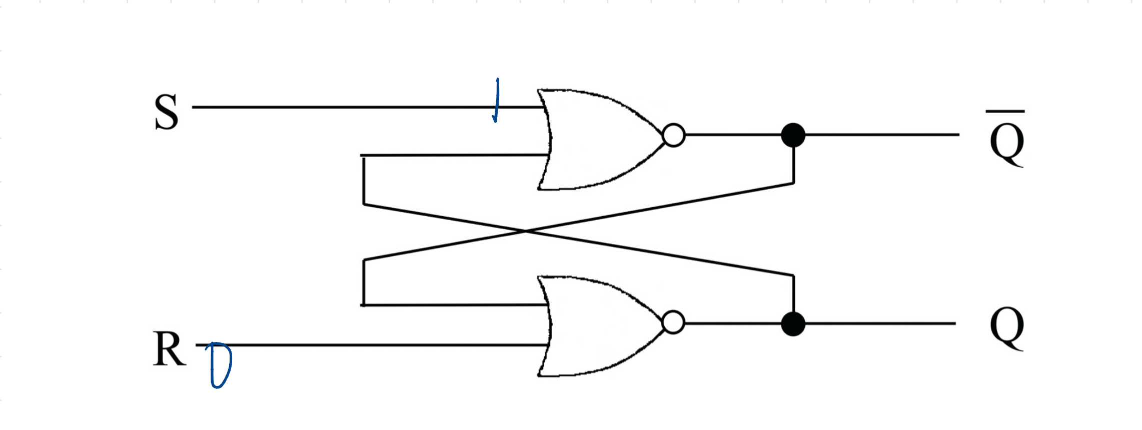

Und-Ebene

Gesperrter Eingang

Oder-Ebene



SR-Latch




| S | R | $\overline{Q}$ | Q |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 1 | 0 | | |

1 Bit Speicherung
Realisierung →

## Aufgabe Ü1: Adressdarstellung

Übung 7

(9 Pkt.)

Viele Rechner besitzen einen Hauptspeicher, in dem 4-Byte-Worte gespeichert werden können. Das bedeutet, dass mit einer einzigen Operation 4 Bytes zwischen Speicher und Prozessor ausgetauscht werden können. Jedes Wort besitzt eine *Adresse* (wie eine Raumnummer). Adressen selbst sind binäre Zahlen, d.h. Bitfolgen einer gegebenen Länge. Die Bitfolge 0...00 adressiert das erste Wort, die Bitfolge 0...01 adressiert das zweite Wort, etc.

a. Nehmen Sie an, dass

(i) für die Adressen eine Bitfolge von 1 Byte verwendet wird;

(ii) für die Adressen eine Bitfolge von 2 Byte verwendet wird;

Geben Sie für jeden Fall die Adresse des letzten Speicherwortes an, das mit der gegebenen Anzahl von Bytes adressiert werden kann:

a) in binärer Notation

b) in oktaler Notation[1]

c) in hexadezimaler Notation

d) in dezimaler Notation

(Hinweis: die Adresse des ersten Wortes ist 0)

计算机的内存是由一系列连续的存储位置组成的 1个 byte是由8个bitfolge组成的

4-Byte-Worte 就是 32位的处理器 (4字节)

a、(i) $2^8 - 1 = 256 - 1 = 255$

a). binär 1111 1111

okt. 377

hex. FF

dez. = 255

(ii) $(2^8)^2 - 1 = 65536 - 1 = 65535$

binär 1111 1111 1111 1111

okt. 177777

hex. FFFF

dez. 65535

Nehmen Sie an die Wortlänge sei 4 Bytes, und 2 Bytes werden für den Adressen-Bitstring verwendet.[2] Nehmen Sie weiterhin an, dass 380 Bytes im Speicher abgelegt werden sollen. Die ersten 4 Bytes werden an der Adresse $123A_{16}$ gespeichert. Der Rest wird ohne Lücken in den folgenden Speicherzellen abgelegt. Bestimmen Sie die Adresse des letzten Speicherwortes, das noch verwendet wird, um die 380 Bytes zu speichern.

Geben Sie die Antwort als i) binäre Zahl, ii) oktale Zahl, iii) dezimale Zahl und iv) als hexadezimale Zahl an.

123A $10 \times 16^0 + 3 \times 16^1 + 2 \times 16^2 + 1 \times 16^3$

$= 10 + 48 + 512 + 4096$

$= 4666$

2 Bytes $= (2^8)^2 = 2^{16} = 65536$ Adresse

$65536 \times 4 = 262144$ Bytes (speichern kann)

$380 \div 4 = 95$ (brautcht 95 Adresse)

$95 - 1 = 94$

$4666 + 94 = 4760$

binär 0001 1001 1000

okt. 11230

dez. 4760

hex. 1298

## Aufgabe Ü2: Übertragungsgeschwindigkeiten

1 Zoll entspricht 2.54 cm

(10 Pkt.)

Die meisten Laserdrucker können mit einer Auflösung von 1200 dpi (hier synonym zu Pixel pro Zoll, oder kurz ppi) drucken.

a. Wenn die Datenübertragung zum Drucker Pixel für Pixel bei 3*8-Bit Farben erfolgt, wie lange dauert dann die Übertragung einer DIN A4 Farbseite (21 cm x 29,7 cm) zum Drucker bei Verwendung folgender Übertragungsmöglichkeiten?

**Hinweis:** Sie können davon ausgehen, dass die hier angegebenen Übertragungsraten verlustlos ohne Protokoll-Overhead ausgenutzt werden können. Rechnen Sie die cm zunächst in Zoll um und runden Sie dieses Ergebnis auf 2 Nachkommastellen! Der Rechenweg muss ersichtlich und nachvollziehbar sein!
Aus historischen Gründen wird bei der Angabe von Datenmengen in `KByte`, `MByte`, ... der Umrechnungsfaktor 1024 verwendet; bei Angaben in `KBit/s`, `MBit/s`, ... wird der Umrechnungsfaktor 1000 verwendet. Sie können die folgende Umrechnungstabelle benutzen:

| | | |
|---|---|---|
| 1 GBit/s = $10^9$ Bit/s | 1 MBit/s = $10^6$ Bit/s | 1 KBit/s = $10^3$ Bit/s |
| 1 GByte = $2^{30}$ Byte | 1 MByte = $2^{20}$ Byte | 1 KByte = $2^{10}$ Byte |

(i) Wireless LAN (IEEE 802.11n) mit 600 MBit/s

(ii) Ethernet mit 1 GBit/s

(i) $21 \div 2.54 = 8.27$ Zoll

$29.7 \div 2.54 = 11.69$ Zoll

$8.27 \times 1200 \times 11.69 \times 1200 \times 3 \times 8$

$= 334\,113\,2928$ Bits

$334\,113\,2928 \div 10^6$

$= 3341.13298$ MBit

$3341.13298 \div 600$

$= 5.57$ s

(ii) $334\,113\,2928 \div 10^9 = 3.34\,113\,2928$ GBit

$3.34 \div 1 = 3.34$ s

b. Nehmen Sie nun an, dass anstatt von Pixeln eine 16-Bit Codierung jedes Zeichens (ein Zeichen wird durch 16 Bit dargestellt) zusammen mit seinen Koordinaten auf der Seite übertragen wird. Dabei können die Zeichen frei auf der Seite positioniert werden indem der Ankerpunkt eines Zeichens eine Koordinate erhält.

(i) Wie viele Bits werden für die Koordinaten benötigt, wenn man die volle Auflösung von 1200 dpi ausnutzen will (d.h. wenn die horizontalen und vertikalen Koordinaten aller Pixel binär kodiert werden müssen)?

(ii) Wie lange dauert es jeweils, 100 Seiten mit 1800 Zeichen an den Drucker zu übertragen, wenn das Adressierungsschema aus Aufgabe bi und die gleichen Übertragungsmöglichkeiten wie aus Aufgabe a zur Verfügung stehen? Runden Sie ihr Ergebnis auf 4 Nachkommastellen!

在这个问题中，我们正在讨论一个具有1200 dpi分辨率的A4纸张。这意味着，每一英寸（2.54厘米）的纸面上，都有1200个独立的像素点。由于A4纸张的大小为21厘米×29.7厘米，我们可以把它转换成英寸（8.27英寸×11.69英寸）并计算出整个纸面上的像素点总数。我们发现这个数值接近于1.4亿。

接下来，问题要求我们考虑如何用二进制表示每一个像素点的坐标。因为我们正在处理一个二维平面（纸面），所以我们需要两个坐标（横坐标和纵坐标）来准确地指出每一个像素点的位置。问题的关键在于，我们需要确定用多少位二进制数才能表示出1.4亿这么多不同的坐标。

这里使用了一个基本的数学概念，即如果我们有N个不同的事物，并且我们想要用二进制表示这些事物，那么我们需要的位数就是log2(N)。比如说，如果我们有8个不同的事物，我们需要3位二进制数（2^3=8）来表示它们。如果我们有16个不同的事物，我们需要4位二进制数（2^4=16）来表示它们。以此类推。

在这个问题中，我们有1.4亿个像素点，所以我们需要log2(1.4亿)位二进制数来表示它们。这个值接近于27。但是，因为我们在处理二维坐标（即我们需要两个数来表示每一个像素点的位置），所以我们实际上只需要sqrt(1.4亿)个不同的值来表示横坐标，以及sqrt(1.4亿)个不同的值来表示纵坐标。这个值接近于11800。所以，我们实际上需要的二进制位数是log2(11800)，这个值接近于14。因此，我们需要大约14位来表示横坐标，以及14位来表示纵坐标，总共需要28位二进制数来表示一个像素点的位置。

这就是为什么在原始答案中说，我们需要大约17位来表示x坐标，17位来表示y坐标，总共需要34位来表示一个像素点的位置。这个数字可能是由于在实际计算中使用了不同的近似值或者舍入策略，所以得出的结果可能会有些微的差别。但是基本的理念和计算过程是一样的。

(i) $8.27 \times 11.69 \times 1200 \times 1200 = 139\,213\,872$ Pixel (Bits)

$\log(139\,213\,872) \approx 17$ Bits

$17 \times 2 = 34$ Bits

(ii) $1800 \times 100 = 180\,000$ Zeichen

Daten = Zeichen × (Zeicheninformation + Koordinateninfo)

$= 180\,000 \times (16$ Bit/Zeichen + 34 Bit/Zeichen)

$= 900\,000\,000$ Bits

$= 9$ GBit $\approx 9.0$ Sekunde

## Aufgabe Ü3: Zahlensysteme

(9 Pkt.)

Bearbeiten Sie folgende Fragen zu Zahlensystemen:

a. Geben sie zu jeder der folgenden Dezimalzahlen ihre Binär-, Oktal- und Hexadezimaldarstellung an.

(i) $(17)_{10}$ 10001 $(21)_8$ $11(16)$

(ii) $(42)_{10}$ $(101010)_2$ $(52)_8$ $(2A)_{16}$

(iii) $(255)_{10}$ $(11111111)_2$ $(377)_8$ $(FF)_{16}$

b. Geben Sie zu folgenden Dualzahlen die Oktal-, Dezimal- und Hexadezimaldarstellung an:

(i) $(10001111)_2$ 143 217 8f

(ii) $(11010101)_2$ 213 325 d5

(iii) $(00011110)_2$ 30 36 1e

## Aufgabe Ü4: Einfachauswahlaufgabe: Einführung

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n“). Nennen Sie dazu in Ihrer Abgabe die jeweils ausgewählte Antwortnummer ((i), (ii), (iii) oder (iv)). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

| | | | |
|---|---|---|---|
| a) Wie viele Bit enthält ein Byte? | | | |
| (i) 16 | (ii) 8 | (iii) 64 | (iv) 32 |
| | | | |
| b) Welche Binärzahl entspricht dem hexadezimalen Wert `C`? | | | |
| (i) `1110` | (ii) `0111` | (iii) `0110` | (iv) `1100` |
| | | | |
| c) Welche Komponente ist gewöhnlich nicht an die South Bridge eines Mainboard-Chipsatzes angebunden? | | | |
| (i) USB-Schnittstellen | (ii) Audio-Ausgang | (iii) Hauptspeicher | (iv) Festplatten |
| | | | |
| d) Wie lautet die höchste Speicheradresse bei einer Adressbreite von $n$ Bit? | | | |
| (i) $2 + n - 1$ | (ii) $2^n - 1$ | (iii) $2/n - 1$ | (iv) $2 * n - 1$ |
| | | | |
| e) Was ist keine Komponente der Prozessorgrundstruktur? | | | |
| (i) Drucker | (ii) Operanden-register | (iii) Arithmetisch-logische Einheit | (iv) Befehlsregister |
| | | | |

## Aufgabe 1: (T) Boolesche Algebra

Tutor 2

(- Pkt.)

Beweisen Sie unter Verwendung des Kommutativ-, Distributiv-, Identitäts- und Komplementärgesetzes (und nur mit diesen alleine) die Gültigkeit folgender Aussagen (Es reicht also nicht die Eigenschaften für {0, 1} zu zeigen!).
**Hinweis:** Sie können bereits bewiesene Aussagen verwenden, um darauf folgende Aussagen zu beweisen.

a. Idempotenz

(i) a · a = a bzw. (ii) a + a = a

b. Null- und Einsgesetz

(i) a · 0 = 0 bzw. (ii) a + 1 = 1

c. Absorptionsgesetz

(i) a · (a + b) = a bzw. (ii) a + (a · b) = a

a.

(i) $a \cdot a$
$= (a \cdot a) + 0$
$= (a \cdot a) + (a \cdot \bar{a})$
$= a(a + \bar{a})$
$= a$

(ii) $a + a = ?$
$= (a + a) \cdot 1$
$= (a + a) \cdot (a + \bar{a})$
$= a(a + \bar{a})$
$= a$

b.

(i) $a \cdot 0 =$
$(a \cdot 0) + 0$
$= (a \cdot 0) + (a \cdot \bar{a})$
$= a(\bar{a} + 0)$
$= a \cdot \bar{a}$
$= 0$

$a + 1$
$= (a + 1) \cdot 1$
$= (a + 1) \cdot (a + \bar{a})$
$= a + (1 \cdot \bar{a})$
$= a + \bar{a}$
$= 1$

c. $a \cdot (a + b)$
$(a + 0) \cdot (a + b)$
$= a + (b \cdot 0)$
$= a + 0$
$= a$

$a + (a \cdot b)$
$= (a \cdot 1) + (a \cdot b)$
$= a \cdot (1 + b)$
$= a \cdot 1$
$= a$

## Aufgabe 2: (T) Funktionstabelle

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

Gegeben sei folgende Booleschen Funktion $f(a, b, c) = a \wedge b \wedge (a \vee \bar{c})$.
Füllen Sie folgende Funktionstabelle aus:

| $a$ | $b$ | $c$ | $\bar{c}$ | $f(a,b,c) = a \wedge b \wedge (a \vee \bar{c})$ |
|---|---|---|---|---|
| 0 | 0 | 0 | | 0 |
| 0 | 1 | 0 | | 0 |
| 0 | 0 | 1 | | 0 |
| 0 | 1 | 1 | | 0 |
| 1 | 0 | 0 | | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | | 0 |
| 1 | 1 | 1 | 0 | 1 |

## Aufgabe 3: (T) Decoder

(- Pkt.)

Bearbeiten Sie die folgenden Aufgaben:

a. Wie viele Ausgänge können beim Decoder gleichzeitig den Wert `wahr` annehmen?

b. Wie viele Eingangsleitungen benötigt ein Decoder, der 16 Ausgangsleitungen besitzt?

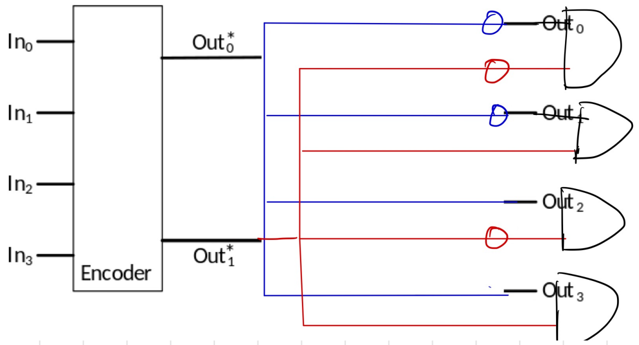
c. Stellen Sie die Kurzform der Funktionstabelle eines 2-zu-4-Decoders mit den Eingangsleitungen $In_0, In_1$ und den Ausgangsleitungen $Out_0, Out_1, Out_2, Out_3$ auf. Tragen Sie Ihre Lösung in die folgende Tabelle ein:

| $In_0$ | $In_1$ | $Out_0$ | $Out_1$ | $Out_2$ | $Out_3$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

a. zu jedem Zeitpunkt nur eine.

b. $\log 16 = 4$ 4 Eingangsleitungen

d. Ergänzen Sie das folgende Schaltnetz so, dass stets gilt $Out_0 = In_0$, $Out_1 = In_1$, $Out_2 = In_2$ und $Out_3 = In_3$. Bei Ihrer Ergänzung dürfen Sie nur auf das Signal an den Leitungen $Out_0^*$ und $Out_1^*$ zugreifen. Es dürfen ausschließlich Leitungen, `NOT`-, `AND`- und `OR`-Bausteine ergänzt werden.




## Aufgabe 4: (T) 2-zu-1 Multiplexer

(- Pkt.)

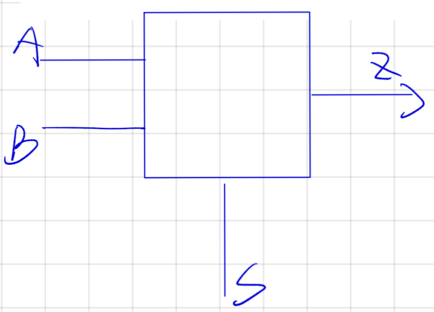
In dieser Aufgabe soll ein 2-zu-1 Multiplexer entworfen werden. Als Input erhält der Multiplexer zwei 1-Bit Kanäle A und B sowie eine 1-Bit Auswahlleitung S. Als Ausgabe liefert der Multiplexer einen 1-Bit Kanal Z. Der Multiplexer soll den Kanal A auf Z schalten, wenn die Auswahlleitung S auf 0 steht. Wenn die S auf 1 steht, soll der Multiplexer den Kanal B auf Z schalten.

a. Erläutern Sie kurz die Funktionsweise eines Multiplexers.

b. Geben Sie die Funktionstabelle, die Boolesche Funktion und das Schaltnetz an.

$S = 0 \Rightarrow A$

$S = 1 \rightarrow B$



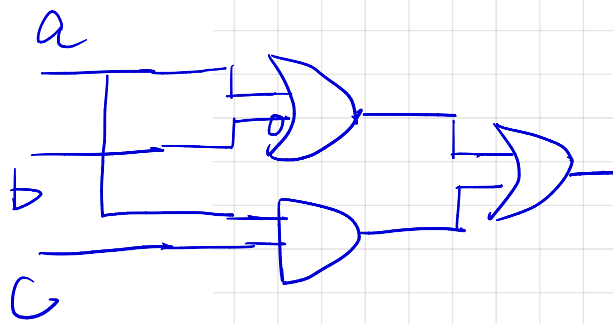

| A | B | S | Z |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 |

übung 2

## Aufgabe Ü1: Funktionstabelle

(10 Pkt.)

Gegeben sei folgende Booleschen Funktion $g(a,b,c) = a \vee \bar{b} \vee (a \wedge c)$.




a. Füllen Sie folgende Funktionstabelle aus:

| $a$ | $b$ | $c$ | $g(a,b,c) = a \vee \bar{b} \vee (a \wedge c)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 |

b. Gegeben sei die folgende Funktionstabelle von sechs dreistelligen Booleschen Funktion $f_1, \ldots, f_6$.

$B^3 \to B^6$

| $A$ | $B$ | $C$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

Schreiben Sie diese Funktionen als Boolesche Terme, wobei sie ausschließlich die Variablen $A, B$ und $C$ benutzen dürfen (insbesondere dürfen Sie die Werte 0 und 1 nicht verwenden)! Nicht in jedem der resultierenden Terme müssen alle Variablen vorhanden sein.

$f_1 = \overline{ABC}$

$f_2 = \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C + ABC$

$f_3 = \overline{B}$

$f_4 = B\overline{C}$

$f_5 = 0$

$f_6 = C$

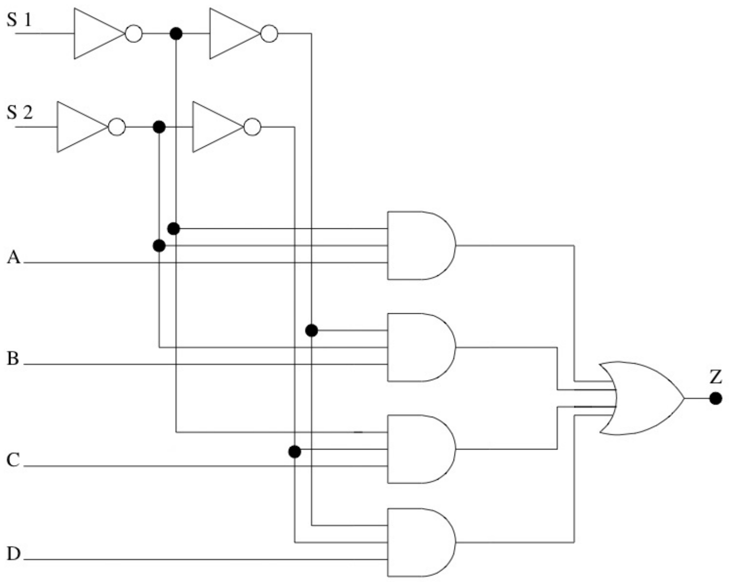
c. Stellen Sie die Funktion $h(a,b,c) = (a \wedge b) \vee c$ unter ausschließlicher Verwendung des NOR-Operators dar! Der Rechenweg muss klar ersichtlich sein!

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

## Aufgabe Ü2: Schaltnetze

(6 Pkt.)

Betrachten Sie das folgende Schaltbild.




a. Beschreiben Sie das Schaltnetz mittels einer Booleschen Funktion für $Z$!

b. Ordnen Sie das Schaltnetz einem Ihnen bekannten Schaltungsbaustein höherer Ordnung zu (Name dieses Bausteins). Wozu werden diese Bausteine ganz allgemein benötigt?

a. $f(A, S_1, S_2) = A * \overline{S_1} * \overline{S_2}$

$f(B, S_1, S_2) = B * S_1 * \overline{S_2}$

$f(C, S_1, S_2) = C * \overline{S_1} * S_2$

$f(D, S_1, S_2) = D * S_1 * S_2$

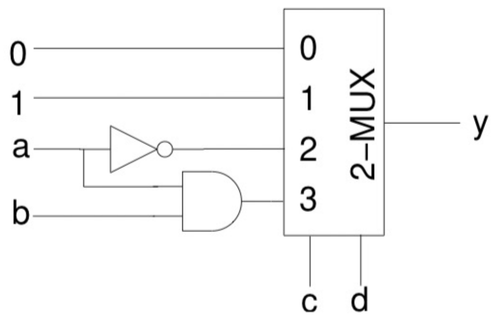
$Z = f(A, B, C, D, S_1, S_2) = (A * \overline{S_1} * \overline{S_2}) + (B * S_1 * \overline{S_2}) + (C * \overline{S_1} * S_2) + (D * S_1 * S_2)$

b. MUX 4 to 1 ; Selektion eines Eingabesignals über zwei Steuerleitungen.

## Aufgabe Ü3: Multiplexer und Boolesche Funktionen

(6 Pkt.)

a. Gegeben ist folgendes Schaltnetz:




Stellen Sie die Boolesche Funktion y=f(a,b,c,d) auf.

b. Gegeben ist folgende Boolesche Funktion:

$$f(a,b,c,d) = ((\overline{a+b}) \cdot c) + \bar{d}$$

Entwerfen Sie das Schaltnetz zu dieser Funktion mit den elementaren Gattern UND, ODER, NICHT.

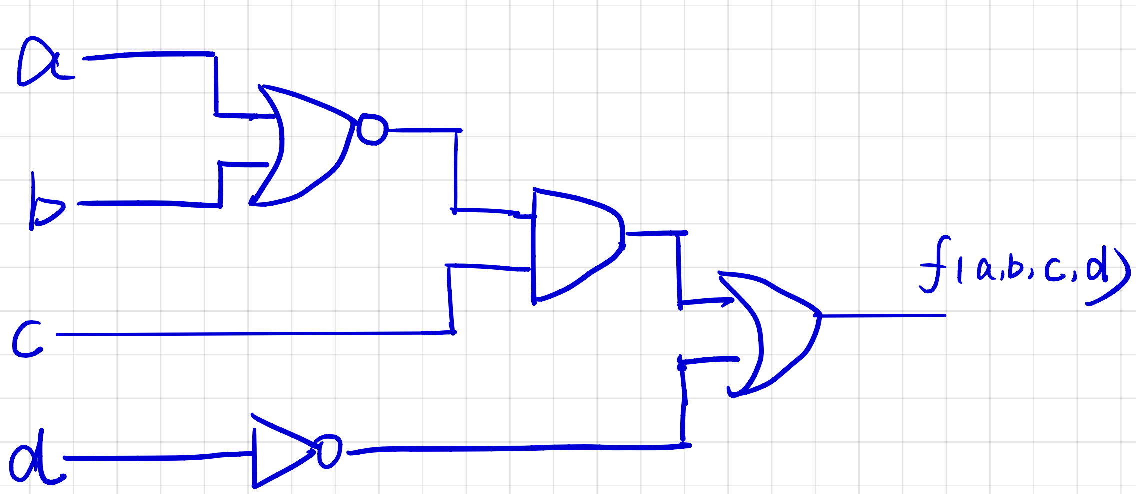
a.

y =

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

$$f(a,b,c,d) = (\bar{c} * d * 0) + (c * \bar{d} * 1) + (\bar{c} * \bar{d} * a) + (c * d * b)$$

b.



## Aufgabe Ü4: Einfachauswahlaufgabe: Boolesche Algebra

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n"). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

| | | | |
|---|---|---|---|
| a) Bei welcher Belegung $(a, b)$ ergibt der OR–Operator ($+$ oder $\vee$) den Wert $0$? | | | |
| (i) $(0, 0)$ | (ii) $(0, 1)$ | (iii) $(1, 0)$ | (iv) $(1, 1)$ |
| | | | |
| b) Bei welcher Belegung $(a, b)$ ergibt der AND–Operator ($\cdot$ oder $\wedge$) den Wert $1$? | | | |
| (i) $(0, 0)$ | (ii) $(0, 1)$ | (iii) $(1, 0)$ | (iv) $(1, 1)$ |
| | | | |
| c) Eine Funktion $f : B^n \to B$ heißt n–stellige Boolesche Funktion ($B = \{0, 1\}$). Wie viele n–stellige Boolesche Funktionen gibt es für jedes beliebige $n \in N$ mit $n \geq 1$? | | | |
| (i) $2^n$ | (ii) $2^{2^n}$ | (iii) $2 \cdot 2^n$ | (iv) $2^{2 \cdot n}$ |
| | | | |
| d) Bei welcher Belegung $(x_1, x_2, x_3, x_4)$ ergibt die Boolesche Funktion $f(x_1, x_2, x_3, x_4) = (x_1 \cdot x_2 \cdot \overline{x_3}) + (x_3 \cdot x_4) + \overline{x_2}$ den Wert 1? | | | |
| (i) $(1, 1, 1, 0)$ | (ii) $(0, 1, 1, 0)$ | (iii) $(0, 1, 0, 1)$ | (iv) $(0, 0, 0, 0)$ |
| | | | |
| e) Wie wird die Anzahl der benötigten Steuereingänge $s$ für einen $n$-Eingaben Multiplexer berechnet? | | | |
| (i) $s = n$ | (ii) $s = 2 * n$ | (iii) $s = log_n n$ | (iv) $s = log_2 n$ |
| | | | |

?

?

对于一个n个输入的多路复用器（Multiplexer），需要的控制输入数量s可以通过公式s = log2(n)计算。

一个n个输入的多路复用器有n个数据（输入）输入和s个控制输入。s表示控制输入的数量，用于选择所需的数据位。控制输入的数量取决于多路复用器要处理的数据位数。

公式s = log2(n)给出了表示输入数量n所需的位数。由于每个控制输入可以有两种状态（0或1），控制输入的每个位表示两个输入选择之间的选择。因此，控制输入位数等于log2(n)。

例如，如果我们有一个4个输入的多路复用器，那么n = 4。根据公式s = log2(n)，我们可以计算出所需的控制输入数量：

s = log2(4) = 2

这意味着对于这个4个输入的多路复用器，需要2个控制输入来选择4个输入之间的选择。

总之，对于一个n个输入的多路复用器，所需的控制输入数量s可以通过s = log2(n)计算得

Tutor 4

## Aufgabe 1: (T) Normalformen einer Schaltfunktion

(- Pkt.)

Gegeben ist folgende Wahrheitstabelle:

| a | b | c | d | f(a,b,c,d) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

a. Geben Sie die Schaltfunktion von f in disjunktiver Normalform (DNF) an.

b. Geben Sie die Schaltfunktion von f in konjunktiver Normalform (KNF) an.

a.

$$f_{DNF}(a,b,c,d) = (\bar{a}\bar{b}\bar{c}\bar{d}) + (\bar{a}\bar{b}c\bar{d}) + (\bar{a}\bar{b}cd) + (\bar{a}b\bar{c}d) + (\bar{a}bcd) +$$
$$(a\bar{b}\bar{c}\bar{d}) + (a\bar{b}c\bar{d}) + (a\bar{b}cd) + (ab\bar{c}d) + (abc\bar{d}) +$$
$$(abcd)$$

b.

$$f_{KNF}(a,b,c,d) = (a+b+c+\bar{d}) \cdot (a+\bar{b}+c+d) \cdot (a+\bar{b}+\bar{c}+d) \cdot (\bar{a}+b+c+\bar{d}) \cdot (\bar{a}+\bar{b}+c+d)$$

## Aufgabe 2: (T) Schaltfunktion in DNF bzw. KNF und Entwurf eines Schaltnetzes

(- Pkt.)

In einer Gefahrenmeldeanlage sollen drei Gefahrentypen durch drei Lämpchen angezeigt werden. Spricht nur einer der drei Melder ($a$, $b$, $c$) an, soll die gelbe Lampe $G$ leuchten ($G = 1$). Melden zwei Melder gleichzeitig, soll die orange Lampe $O$ leuchten ($O = 1$) und nur wenn alle drei Melder Alarm geben, soll die rote Lampe $R$ aufleuchten ($R = 1$).

a. Stellen Sie die Funktionstabelle der Gefahrenmeldeanlage auf.

| a | b | c | G | O | R |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |

b. Leiten Sie aus der Funktionstabelle die Schaltfunktionen für Ausgang $R$ sowohl in disjunktiver Normalform (DNF), als auch in konjunktiver Normalform (KNF) her.

$f_{DNF}(a,b,c) = abc$

$f_{KNF}(a,b,c) = (a+b+\bar{c}) \cdot (a+\bar{b}+c) \cdot (a+b+c) \cdot (a+\bar{b}+\bar{c}) \cdot (\bar{a}+b+\bar{c}) \cdot (\bar{a}+\bar{b}+c) \cdot (\bar{a}+b+c)$

c. Welche der beiden Darstellungen (KNF, DNF) ist in diesem Fall günstiger? Begründen Sie ihre Aussage.

kosten $f_{DNF}(a,b,c) = 2$

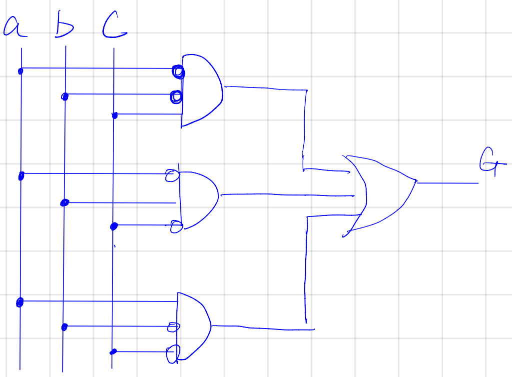
$f_{KNF}(a,b,c) = 2 \times 7 + 6 = 20$

DNF ist in diesem Fall günstiger

d. Geben Sie eine Funktionsgleichung der gelben Lampe an.

$f_{DNF}(a,b,c) = (\bar{a}\bar{b}c) + (\bar{a}b\bar{c}) + (a\bar{b}\bar{c})$

e. Zeichnen Sie ein Schaltbild für den Ausgang $G$.



# Aufgabe 3: (T) Programmierbare logische Arrays

a. Erläutern Sie kurz die grundlegende Idee eines PLAs!

**Entwurf eines universell verwendbaren Einheitsbausteins mit möglichst homogener Netzstruktur, der für unterschiedliche Anwendungen eingesetzt werden kann.**

这句话的意思是设计一个具有尽可能均匀网络结构的通用模块，可用于不同的应用领域。"Einheitsbaustein"指的是一个统一的模块或组件，它具有通用性，可以在不同的应用中使用。这个模块可能是电子电路、软件算法或其他系统的一部分。"möglichst homogene Netzstruktur"表示设计中希望实现一种尽可能均匀的网络结构。这意味着在模块内部的连接或组织方式应该是均匀的，以便适应各种不同的应用场景。通过具有均匀的网络结构，模块可以更灵活地适应多个应用，并提供更好的可扩展性和可重用性。

b. Erläutern Sie, was es bedeutet, wenn Eingänge

(i) neutralisiert werden!

(ii) gesperrt werden!

(i) Anlegen von 1 an den Inputs der oberen Feldzeile

(ii) Anlegen von 0 an nicht benötigten Eingängen

c. Ein normiertes PLA besteht aus einer Und-Ebene und aus einer Oder-Ebene. Erklären Sie diese beiden Begriffe kurz. Ausgehend von einem 5-mal-4-PLA: Wie groß werden Und- und Oder-Ebene jeweils, wenn durch das PLA eine dreistellige Boolesche Funktion realisiert werden soll?

Und-Ebene:

Besteht nur aus Identen und (Negat-) Multiplizieren Sie dient zur Erzeugung der benötigten Produkt-Terme

Oder-Ebene:

Besteht nur aus Identen und Addieren. Sie dient zur Erzeugung der Summen

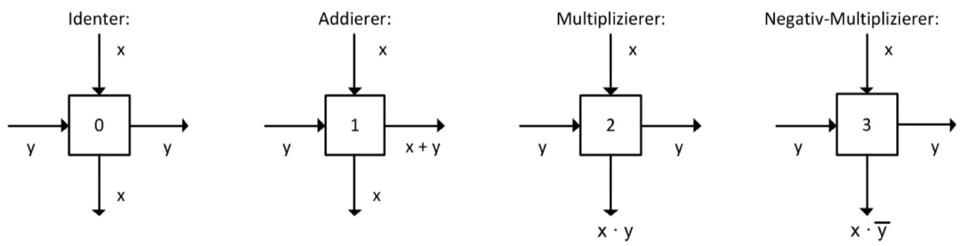
Und-Ebene

3 Zeilen (3 Eingänge)

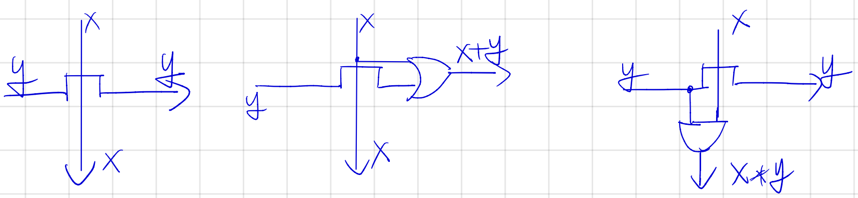
Oder-Ebene

2 Zeilen (verbleibende Zeilen)

d. Intern ist jedes PLA gitterförmig verdrahtet, wobei sich an jedem Kreuzungspunkt von zwei Drähten einer von vier möglichen Bausteinen befindet. Diese Bausteine sind:




Zeichnen Sie das Schaltbild für jeden der vier Bausteine. Verwenden Sie dazu Und-, Oder- und Nicht-Gatter!



e. Gegeben sei die folgende Boolesche Funktion $f : B^3 \to B^2$:

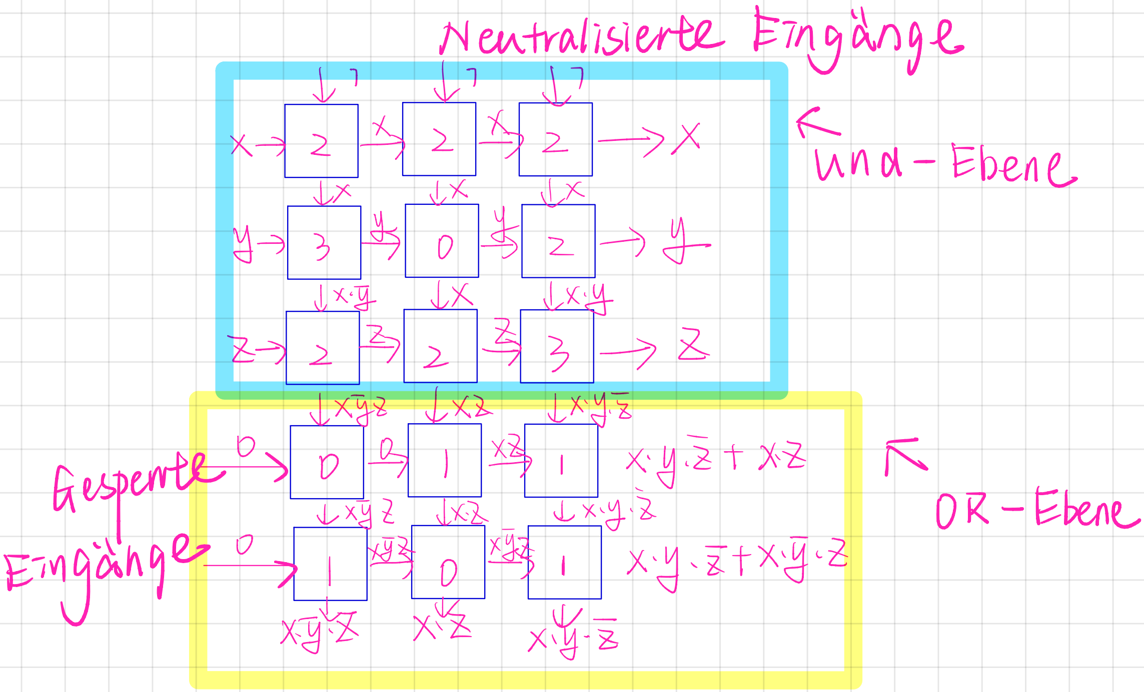
$$f(x,y,z) = (x \cdot y \cdot \overline{z}) + (x \cdot z), (x \cdot y \cdot \overline{z}) + (x \cdot \overline{y} \cdot z)$$

Realisieren Sie diese Funktion durch ein normiertes PLA, welches aus der minimal möglichen Anzahl an Zeilen und Spalten besteht. Verwenden Sie ausschließlich die in Aufgabenteil d) gegebenen Bausteine vom Typ 0 bis 3. Kennzeichnen Sie in Ihrer Skizze die Und- und die Oder-Ebene. Markieren Sie gesperrte und neutralisierte Eingänge. Beschriften Sie jeden Pfeil (sowohl ausgehende als auch die innerhalb des PLAs) mit der jeweils anliegenden logischen Funktion.

Minterme 3 Spalten

Schaltfunktionen 3+2 = 5

5x3 PLAs

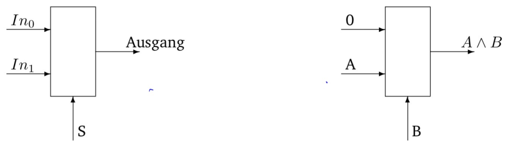


Übung 4

**Aufgabe Ü1:** Multiplexer

(6 Pkt.)

In dieser Aufgabe sollen logische Gatter durch 2-Eingaben Multiplexer dargestellt werden. Gehen Sie davon aus, dass $S$ die Steuerleitung ist und für $S = 0$ der Eingang $In_0$ und für $S = 1$ der Eingang $In_1$ selektiert wird. So kann die Funktion $A \wedge B$ zum Beispiel durch Anlegen von 0 an den Eingang $In_0$, A an den Eingang $In_1$ und B an die Steuerleitung $S$ durch einen Multiplexer realisiert werden.

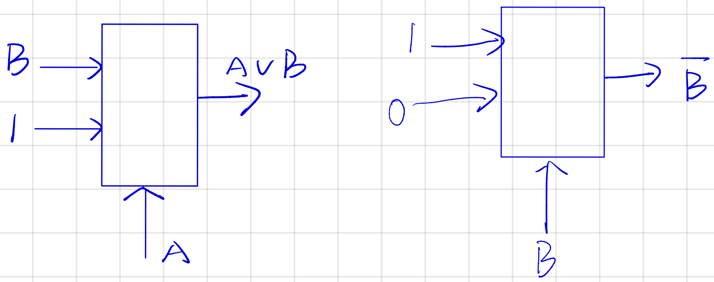



Erstellen Sie zwei 2-Eingaben Multiplexer, welche folgende Eigenschaften erfüllen sollen:

a. Dieser Multiplexer soll den Term $A \vee B$ an seinem Ausgang erzeugen.

b. Dieser Multiplexer soll den Term $\overline{B}$ an seinem Ausgang erzeugen.

Für jeden der Eingänge des Multiplexers ($In_0$, $In_1$ sowie die Steuerleitung $S$) dürfen Sie ausschließlich die Werte A, B, $\overline{A}$, $\overline{B}$ sowie 0 und 1 benutzen. Sie dürfen keine weiteren Bausteine oder Gatter benutzen.




| $\overline{A}$ | A | B | $A \vee B$ | $\overline{B}$ | $A \wedge B$ |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |

**Aufgabe Ü2:** Normalformen einer Schaltfunktion

Gegeben ist folgende Wahrheitstabelle:

| a | b | c | d | f(a,b,c,d) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

a. Geben Sie die Schaltfunktion von f in disjunktiver Normalform (DNF) an.

b. Geben Sie die Schaltfunktion von f in konjunktiver Normalform (KNF) an.

c. Welche der beiden Darstellungen (KNF, DNF) ist in diesem Fall günstiger? Begründen Sie ihre Aussage unter der Annahme, dass für ein einzelnes Gatter (NOT, AND and OR) Kosten von 1 entstehen.

a) $f_{DNF}(a,b,c,d) =$
$\bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}cd + \bar{a}bcd + a\bar{b}\bar{c}d + a\bar{b}c\bar{d}$
$+ ab\bar{c}d + abcd$

b) $KNF = (a+b+c+d) * (a+\bar{b}+c+d)$
$* (a+\bar{b}+c+\bar{d}) * (a+\bar{b}+\bar{c}+d)$
$* (\bar{a}+b+c+d) * (\bar{a}+b+\bar{c}+\bar{d})$
$* (\bar{a}+\bar{b}+c+d) * (\bar{a}+\bar{b}+\bar{c}+d)$

(NOT)

$3 \times 8 + 7 + 13 = 44$

$3 \times 8 + 7 + 13 = 44$

c) gleich günstig

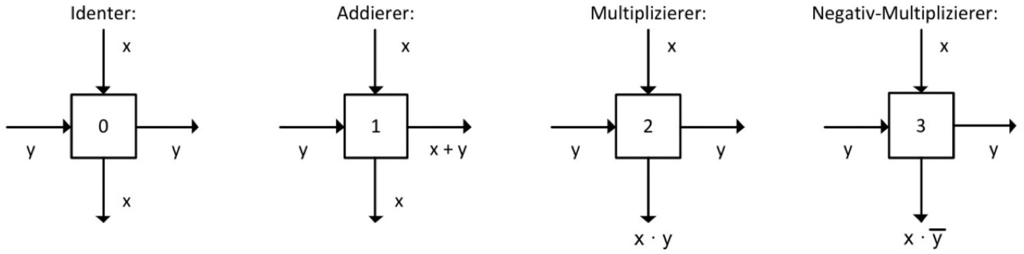
## Aufgabe Ü3: PLA-Entwurf

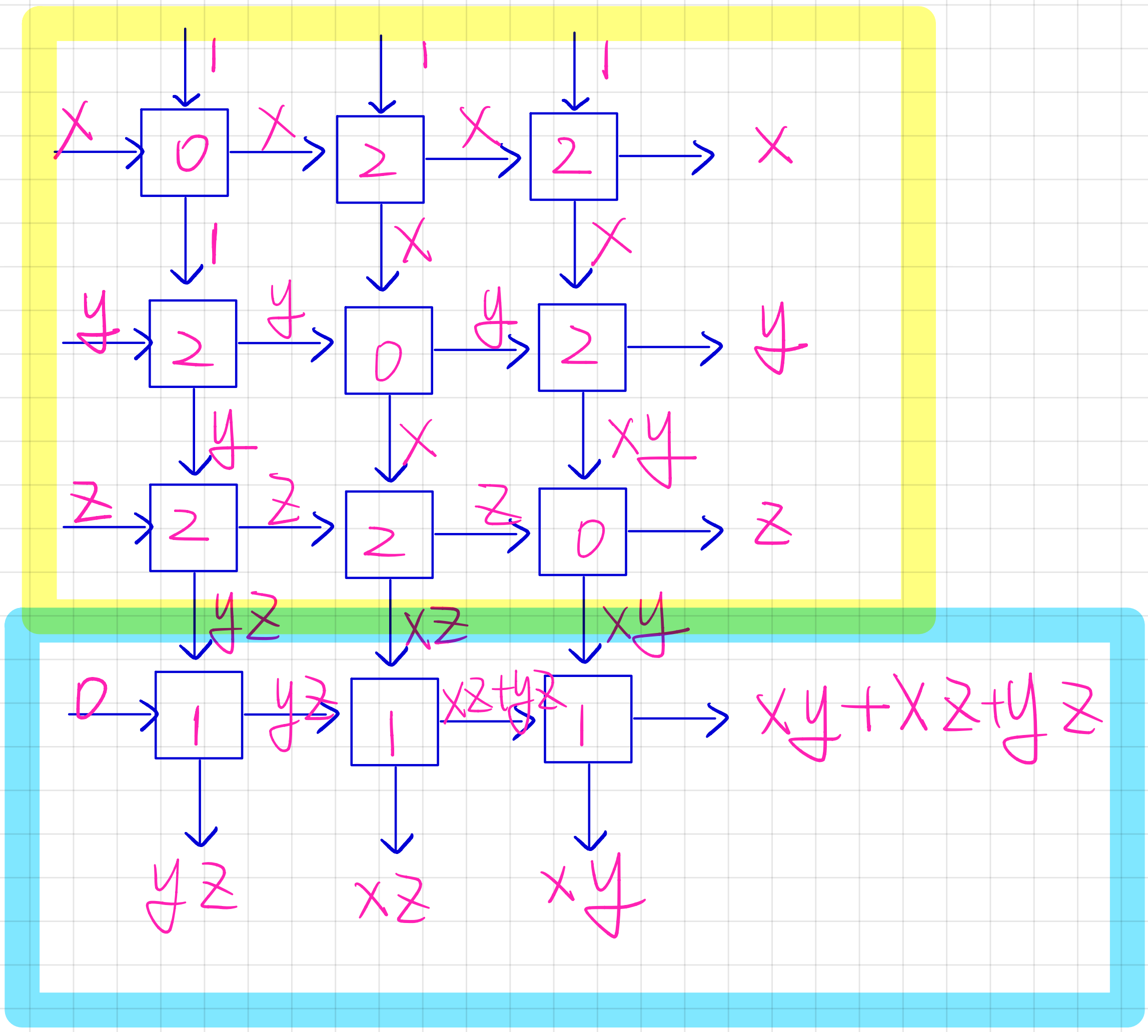
(10 Pkt.)

Gegeben sei die folgende Boolesche Funktion

$$f(x, y, z) = (x \cdot y) + (x \cdot z) + (y \cdot z)$$

Realisieren Sie diese Funktion durch ein normiertes PLA, welches aus der minimal möglichen Anzahl an Zeilen und Spalten besteht. Verwenden Sie ausschließlich Bausteine der in Aufgabe 14) dargestellten Typen 0 bis 3. Kennzeichnen Sie in Ihrer Skizze die Und- und die Oder-Ebene. Markieren Sie gesperrte und neutralisierte Eingänge. Beschriften Sie jeden Pfeil (sowohl ausgehende als auch die innerhalb des PLAs) mit der jeweils anliegenden logischen Funktion. Die zur Verfügung stehenden Bausteine sind:



## Aufgabe Ü4: Einfachauswahlaufgabe: Normalformen und PLA

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n“). Nennen Sie dazu in Ihrer Abgabe die jeweils ausgewählte Antwortnummer ((i), (ii), (iii) oder (iv)). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

a) Sei folgende Wahrheitstafel einer Booleschen Funktion gegeben. Was ist die Menge der einschlägigen Indizes?

| $i$ | $x_1$ | $x_2$ | $f(x_1, x_2)$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 0 |

| | | | |
|---|---|---|---|
| (i) $\{0\}$ | (ii) $\{2\}$ | (iii) $\{0, 1\}$ | (iv) $\{0, 1, 3\}$ |

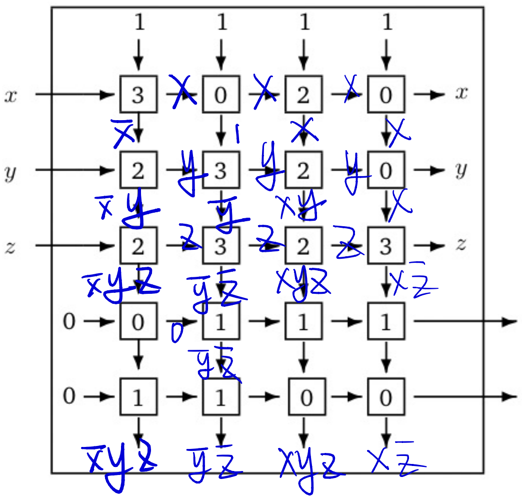
b) Welche der folgenden Mengen an Booleschen Funktionen ist nicht funktional vollständig?

| | | | |
|---|---|---|---|
| (i) $\{OR, NOT\}$ | (ii) $\{AND, OR\}$ | (iii) $\{NAND\}$ | (iv) $\{AND, NOT\}$ |

c) Jede Boolesche Funktion $F : B^n \to B$ ist eindeutig darstellbar als...

| | | | |
|---|---|---|---|
| (i) ... Produkt der Minterme ihrer einschlägigen Indizes. | (ii) ... Summe der Minterme ihrer einschlägigen Indizes. | (iii) ... Summe der Minterme ihrer nichteinschlägigen Indizes. | (iv) ... Summe der Maxterme ihrer einschlägigen Indizes. |

d) Welche Boolesche Funktion realisiert folgendes PLA?




| | | | |
|---|---|---|---|
| (i) $f(x, y, z) = (yz + xy + \overline{z}, \overline{x}z + \overline{y}\,\overline{z})$ | (ii) $f(x, y, z) = (\overline{y}\,\overline{z}, xyz + \overline{y})$ | (iii) $f(x, y, z) = (\overline{y}\,\overline{z}, \overline{x}yz + \overline{y}z)$ | (iv) $f(x, y, z) = (\overline{y}\,\overline{z} + xyz + \overline{z}, \overline{x}yz + \overline{y}\,\overline{z})$ |

e) Welcher der folgenden Booleschen Terme ist äquivalent zu $(x_1 \cdot x_2) + x_1 + x_3$?

| | | | |
|---|---|---|---|
| (i) $(\overline{x_1}x_2x_3) + (x_1\overline{x_2}x_3)$ | (ii) $(x_1x_2x_3) \cdot (x_1\overline{x_2}x_3)$ | (iii) $(x_1 + x_2 + x_3) + (x_1\overline{x_2}x_3)$ | (iv) $(x_1 + x_2 + x_3) \cdot (x_1 + \overline{x_2} + x_3)$ |

# Tutor 6

## Aufgabe 1: (T) Resolutionsregel

(- Pkt.)

Vereinfachen Sie den folgenden Booleschen Term unter Anwendung der Resolutionsregel soweit wie möglich:

$$f(x_1, x_2, x_3, x_4) = \overline{x}_1\overline{x}_2x_3\overline{x}_4 + \overline{x}_1x_2x_3\overline{x}_4 + x_1\overline{x}_2x_3x_4 + x_1x_2\overline{x}_3x_4 + x_1x_2x_3x_4$$

Geben Sie dabei **alle** notwendigen Zwischenschritte an!
**Hinweis:** Diese Aufgabe wurde in Tutoriumsvideo 8 zum 2ten Wiederholungsblatt ab Minute 29:55 besprochen.

$= (\overline{x}_1\overline{x}_2x_3\overline{x}_4) + (\overline{x}_1x_2x_3\overline{x}_4) + (x_1\overline{x}_2x_3x_4) + (x_1x_2\overline{x}_3x_4) + (x_1x_2x_3x_4)$

$= (\overline{x}_1\overline{x}_2x_3\overline{x}_4) + (\overline{x}_1x_2x_3\overline{x}_4) + (x_1\overline{x}_2x_3x_4) + (x_1x_2\overline{x}_3x_4) + (x_1x_2x_3x_4) + (x_1x_2x_3x_4)$

$= \overline{x}_1x_3\overline{x}_4 + x_1x_3x_4 + x_1x_2x_4$

## Aufgabe 2: (T) Minimierung mittels Karnaugh

(- Pkt.)

Minimieren Sie folgende Funktionen mit Hilfe des Karnaugh-Diagramms.
Geben Sie dabei sowohl das jeweilige gezeichnete Karnaugh-Diagramm, als auch die zugehörige minimierte Funktion in disjunktiver Form an!

a. $y_1 = (x_1x_2\overline{x}_3) + (x_1\overline{x}_2\overline{x}_3) + (\overline{x}_1\overline{x}_2\overline{x}_3) + (\overline{x}_1x_2\overline{x}_3) + (x_1\overline{x}_2x_3) + (x_1x_2x_3)$

b. $y_2 = (\overline{x}_2x_3x_4) + (\overline{x}_1x_2x_3x_4) + (x_1x_2\overline{x}_3x_4) + (\overline{x}_1x_2\overline{x}_3x_4) + (\overline{x}_1x_2x_3\overline{x}_4) + (\overline{x}_1x_2\overline{x}_3\overline{x}_4) + (\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4)$

| $x_3$ \ $x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | | | 1 | 1 |

$\overline{x}_3 + x_1$

| $x_3x_4$ \ $x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 10 | | 1 | | |
| 11 | 1 | 1 | | 1 |
| 01 | | 1 | 1 | |
| 00 | 1 | 1 | | |

$(\overline{x}_1x_2) + (\overline{x}_2x_3x_4) + (x_2\overline{x}_3x_4) + (\overline{x}_1\overline{x}_3\overline{x}_4)$

## Aufgabe 3: (T) Schaltfunktion

(- Pkt.)

Gegeben ist folgende Wahrheitstabelle:

| a | b | c | d | f(a,b,c,d) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

a. Geben Sie die Schaltfunktion von f in disjunktiver Normalform (DNF) an.

$$f_{DNF}(a,b,c,d) = (\bar{a}\bar{b}c\bar{d}) + (\bar{a}\bar{b}cd) + (\bar{a}bc\bar{d}) + (\bar{a}bcd) + (a\bar{b}c\bar{d}) + (a\bar{b}cd) + (abc\bar{d})$$

b. Vereinfachen Sie die Funktion unter Verwendung eines Karnaugh-Diagramms.

c. Nehmen Sie an, dass die Wahrheitstabelle wie oben gegeben ist, jedoch ohne die letzte Zeile. Das heißt, die neue Funktion f' ist auf dem Eingabe-4-Tupel (a=1, b=1, c=1, d=1) undefiniert. Wie wirkt sich das auf Ihre Möglichkeiten aus, die neue Funktion f' zu vereinfachen? Verdeutlichen Sie Ihre Antwort an einem neuen Karnaugh-Diagramm, und geben Sie eine möglichst einfache Darstellung von f' an.

b.

| cd \ ab | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 10 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | | 1 |
| 01 | | | | |
| 00 | | | | |

两个不懂点:
① 为什么这里不合成一个四方格
② 那种首尾接法怎么接

可以直接变四格

$\bar{a}c + cd + \bar{b}c$

Don't Care

c.

| cd \ ab | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 10 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | D | 1 |
| 01 | | | | |
| 00 | | | | |

C

## Aufgabe 4: (T) Quine-McCluskey

(- Pkt.)

a. Vereinfachen Sie den folgenden Booleschen Term unter Anwendung des Algorithmus von Quine-McCluskey:
$f(x) = \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4 + \overline{x}_1x_2x_3x_4 + x_1\overline{x}_2\overline{x}_3\overline{x}_4 + x_1\overline{x}_2\overline{x}_3x_4 + x_1\overline{x}_2x_3x_4 + x_1x_2\overline{x}_3\overline{x}_4 + x_1x_2\overline{x}_3x_4$
Geben Sie dabei alle notwendigen Schritte an!

Implikanten bestimmen

| Gruppe | Minterm | Einschlägiger Index | |
|---|---|---|---|
| 1 | $\bar{x}_1x_2x_3x_4$ | 0111 | 1 |
| | $x_1\bar{x}_2x_3x_4$ | 1011 | 2 |
| | $x_1x_2\bar{x}_3x_4$ | 1101 | 11 |
| 2 | $x_1\bar{x}_2\bar{x}_3x_4$ | 1001 | 10 |
| | $x_1x_2\bar{x}_3\bar{x}_4$ | 1100 | 15 |
| 3 | $x_1\bar{x}_2\bar{x}_3\bar{x}_4$ | 1000 | 14 |
| 4 | $\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$ | 0000 | 12 |

| $x_3x_4$ \ $x_1x_2$ | 00 | 01 | 10 | 11 |
|---|---|---|---|---|
| 11 | 0 | 1: 1 | 2: 1 | 3 |
| 10 | 4 | 5 | 6 | 7 |
| 01 | 8 | 9 | 10: 1 | 11: 1 |
| 00 | 12: 1 | 13 | 14: 1 | 15: 1 |

Implikanten verkürzen

| Gruppe | Minterm | Einschlägiger Index | |
|---|---|---|---|
| 1 | $\bar{x}_1x_2x_3x_4$ | 0111 | 1 |
| | $x_1\bar{x}_2x_4$ | 10*1 | 2,10 |
| | $x_1x_2\bar{x}_3$ | 110* | 11,15 |
| | $x_1\bar{x}_3x_4$ | 1*01 | 10,11 |
| 2 | $x_1\bar{x}_2\bar{x}_3$ | 100* | 10,14 |
| | $x_1\bar{x}_3\bar{x}_4$ | 1*00 | 14,15 |
| 3 | $\bar{x}_2\bar{x}_3\bar{x}_4$ | *000 | 12,14 |

| Gruppe | Minterm | Einschlägiger Index | |
|---|---|---|---|
| 1 | $\bar{x}_1x_2x_3x_4$ | 0111 | 1 |
| | $x_1\bar{x}_2x_4$ | 10*1 | 2,10 |
| | $x_1\bar{x}_3$ | 1*0* | 10,11,14,15 |
| 3 | $\bar{x}_2\bar{x}_3\bar{x}_4$ | *000 | 12,14 |

verkürzte Boolsche Funktion bestimmen

Primimplikant

b. Berechnen Sie die Kosten $K_1$ vor und $K_2$ nach der Optimierung. Wie viel kann an Kosten eingespart werden? Gehen Sie davon aus, dass die Gatter AND, OR und NOT jeweils Kosten von 1 verursachen.

c. Begründen Sie, ob in diesem Beispiel auch eine Optimierung mittels Karnaugh-Diagramme möglich wäre.

Übung 6

## Aufgabe Ü1: Resolutionsregel

(4 Pkt.)

Vereinfachen Sie den folgenden Booleschen Term unter Anwendung der Resolutionsregel soweit wie möglich:

$$f(x_1, x_2, x_3, x_4) = \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4 + \overline{x}_1\overline{x}_2\overline{x}_3x_4 + \overline{x}_1x_2x_3x_4 + x_1x_2\overline{x}_3x_4 + x_1x_2x_3x_4$$

Geben Sie dabei **alle** notwendigen Zwischenschritte an!

$f(x_1, x_2, x_3, x_4) =$

$(x_1 x_2 x_3 x_4) + (\overline{x}_1 \overline{x}_2 \overline{x}_3 \overline{x}_4) + (\overline{x}_1 \overline{x}_2 \overline{x}_3 x_4) + (\overline{x}_1 x_2 x_3 x_4) + (x_1 x_2 \overline{x}_3 x_4) + (x_1 x_2 x_3 x_4)$

$= (\overline{x}_1 \overline{x}_2 \overline{x}_3) + (x_2 x_3 x_4) + (x_1 x_2 x_4)$

## Aufgabe Ü2: Optimierung von Schaltnetzen

(11 Pkt.)

a. Gegeben sei folgende Wahrheitstabelle einer Funktion $f(x_1, x_2, x_3, x_4)$.
Leiten Sie aus dieser Wahrheitstabelle die Schaltfunktion in ihrer vollständigen konjunktiven Normalform (**KNF**) her.

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $f(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | 1 |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 1 |
| 13 | 1 | 1 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 | 1 |

$(x_1 + x_2 + x_3 + x_4) * (x_1 + x_2 + x_3 + \overline{x}_4) * (x_1 + x_2 + \overline{x}_3 + \overline{x}_4) * (x_1 + \overline{x}_2 + x_3 + x_4)$

$(x_1 + \overline{x}_2 + \overline{x}_3 + x_4) * (\overline{x}_1 + x_2 + \overline{x}_3 + \overline{x}_4)$

b. Im Folgenden ist die Wahrheitstabelle der Funktion $g(x_1, x_2, x_3, x_4)$ gegeben.
Minimieren Sie die Funktion $g$ unter Verwendung eines Karnaugh-Diagramms grafisch. Kennzeichnen Sie **alle** Blöcke innerhalb Ihres Karnaugh-Diagramms, die Sie für Ihre Vereinfachung verwenden! Geben Sie abschließend die minimierte Funktion in disjunktiver Form an!

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $g(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 |
| 6 | 0 | 1 | 1 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 |
| 10 | 1 | 0 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 0 | 0 |
| 15 | 1 | 1 | 1 | 1 | 1 |

| $x_3x_4$ \ $x_1x_2$ | 00 | 01 | 10 | 11 |
|---|---|---|---|---|
| 11 | | 1 | | 1 |
| 10 | | 1 | | |
| 01 | 1 | | | 1 |
| 00 | 1 | | | |

$$\bar{x}_1 x_2 x_3 + \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_3 x_4$$

## Aufgabe Ü3: Karnaugh

(9 Pkt.)

Gegeben sei die Wahrheitstabelle einer partiellen Booleschen Funktion $g(x_1, x_2, x_3, x_4)$. Undefinierte Ausgaben sind mit einem **D** gekennzeichnet:

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $g(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | **D** |
| 9 | 1 | 0 | 0 | 1 | **D** |
| 10 | 1 | 0 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | **D** |
| 13 | 1 | 1 | 0 | 1 | **D** |
| 14 | 1 | 1 | 1 | 0 | **D** |
| 15 | 1 | 1 | 1 | 1 | **D** |

Minimieren Sie die Funktion $g$ unter Verwendung eines Karnaugh-Diagramms grafisch. Beachten Sie dabei die **Don't Care** Argumente. Kennzeichnen Sie **alle** Blöcke innerhalb Ihres Karnaugh-Diagramms, die Sie für Ihre Vereinfachung verwenden! Geben Sie abschließend die minimierte Funktion in disjunktiver Form an!

| $x_3x_4$ \ $x_1x_2$ | 00 | 01 | 10 | 11 |
|---|---|---|---|---|
| 11 | | 1 | | D |
| 10 | | | | D |
| 01 | 1 | 1 | D | D |
| 00 | 1 | | D | D |

$\overline{x_3} + \overline{x_3}x_4 + x_1\overline{x_3} + x_1x_2$

# Tutor 7

# 1. Darstellung ganzer Zahl

a. Geben Sie die folgenden Dezimalzahlen als Dualzahlen in ihrer 1er-Komplement-, 2er-Komplement- und in Sign/Magnitude-Darstellung an (jeweils 10 Bit). Bei der Sign/Magnitude-Darstellung wird das hochwertigste Bit als Vorzeichen interpretiert: $(b_9...b_1b_0)_2 = (-1)^{b_9} * \sum_{i=0}^{8} b_i 2^i$

(i) $(123)_{10}$

(ii) $(-123)_{10}$

(i) $(123)_{10} = (0001111011)_2$

$(-123)_{10}$ Sign/Magnitude

$(1001111011)_2$

1er-komplement

$(1110000100)_2$

2er-komplement

$(1110000101)_2$

b. Wandeln Sie folgende Dualzahlen in ihre Dezimaldarstellung um. Interpretieren Sie die Dualzahlen jeweils als in 1er- und 2er-Komplement-Darstellung sowie in Sign/Magnitude-Darstellung gegeben.

(i) $(1111101011)_2$

(ii) $(0001011010)_2$

(i) Sign/Magnitude

$(1111101011)_2$

$(0111101011)_{10}$ $\quad 1\times 2^0 + 1\times 2^1 + 1\times 2^3 + 2^5 + 2^6 + 2^7 + 2^8$

$= 1+2+8+32+64+128+256$

$= 491$

$(1111101011)_2$

$= -491$

1er-Darstellung

$(0000010100)_{10}$ $\quad 2^2 + 2^4 = 4+16 = 20$

$(1111101011)_2$

$= -20$

2er-Darstellung

$(0000010101)_{10} = 2^0 + 2^2 + 2^4 = 21$

$(1111101011)_2$

$= -21$

c. Geben Sie jeweils in 1er- und 2er-Komplement-Darstellung und in Sign/Magnitude-Darstellung bei Verwendung von 10 Bits an:

(i) die größte darstellbare positive Zahl,

(ii) die kleinste darstellbare positive Zahl,

(iii) die größte darstellbare negative Zahl (d.h. die negative Zahl, die den geringsten Abstand zur Null hat),

(iv) die kleinste darstellbare negative Zahl (d.h. die negative Zahl, die den größten Abstand zur Null hat),

(v) die Zahl Null.

| (i) | Magnitude | 1er- | 2er- |
|---|---|---|---|
| grö. po. | 0111111111 | 0111111111 | 0111111111 |
| klein. | 0000000001 | 0000000001 | 0000000001 |
| grö. ne. | 1000000001 | 1111111110 | 1111111111 |
| klein. ne. | 1111111111 | 1000000000 | 1000000000 ? |
| 0 | 0000000000<br>1000000000 | 0000000000<br>1111111111 | 0000000000<br>(eindeutig) |

d. Gibt es einen Unterschied zwischen „2er-Komplement“ und „2er-Komplement-Darstellung“? Wenn ja, welchen?

?

## Aufgabe 2: (T) Addition von Dualzahlen

(- Pkt.)

In dieser Aufgabe sollen die Grundlagen der Addition in Einer- bzw. Zweierkomplement-Darstellung vertieft werden. Verwenden Sie zur binären Darstellung sämtlicher vorkommenden Zahlen jeweils 8 Bits.

a. Gegeben seien die Zahlen $(-17)_{10}$ sowie $(7)_{10}$.

(i) Geben Sie die Einerkomplement-Darstellung der beiden Zahlen an.

(ii) Geben Sie die Zweierkomplement-Darstellung der beiden Zahlen an.

(i) $(-17)_{10}$ $(7)_{10}$

$(17)_2 = 00010001$ $(7)_2 = 00000111$

$(-17)_2 = 11101110$

(ii) $(-17)_2 = 11101111$ $(7)_2 = 00001000$

正数不用加1

11101111

b. Addieren Sie die Zahlen $(-17)_{10}$ und $(7)_{10}$ binär. Verwenden Sie dazu

(i) die Einerkomplement-Darstellung.

(ii) die Zweierkomplement-Darstellung.

(i) $(-17)_2 = 11101110$

$(7)_2 = 00000111$

$\overline{\phantom{0}11110101}$

$(00001010)_{10} = 10$

$(11110101)_{10} = -10$

(ii) $(-17)_2 = 11101111$

$(7)_2 = 00000111$

$\overline{\phantom{0}11110110}$

## Aufgabe 3: (T) Gleitkommazahlen

(- Pkt.)

Nach dem IEEE 754 Standard gilt:

$$(-1)^S \cdot (1 + \text{Signifikant}) \cdot 2^{(\text{Exponent}-\text{Bias})}$$

wobei der Standard

- für das Vorzeichen $S$ ein Bit, —用于表示正负 0是正 1是负
- für den Signifikanten (Mantisse) 23 Bit bei einfacher und 52 Bit bei doppelter Genauigkeit,
- für den Exponenten 8 Bit bei einfacher und 11 Bit bei doppelter Genauigkeit

reserviert und den Bias auf $127 = 2^{8-1} - 1$ bei einfacher bzw. auf $1023 = 2^{11-1} - 1$ bei doppelter Genauigkeit setzt.

a. Geben Sie die Darstellung folgender Zahlen als Gleitkommazahl nach IEEE 754 in einfacher (32-Bit) Genauigkeit an:

(i) $(11,25)_{10}$

(ii) $(0,2)_{10}$

b. Wandeln Sie folgende Zahl, die in Gleitkommadarstellung (IEEE 754) gegeben ist, in ihre Dezimaldarstellung um.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | Exponent | | | | | | | | Significand | | | | | | | | | | | | | | | | | | | | | | |

(i) 11

Übung 7

## Aufgabe Ü1: Addition von Dualzahlen

(9 Pkt.)

Beantworten Sie folgende Fragen im Bezug auf die 2er-Komplement-Darstellung ganzer Zahlen:

a. Geben Sie

(i) die kleinste darstellbare Zahl,
(ii) die größte darstellbare Zahl,
(iii) sowie die Null

unter Verwendung von 8 Bit an.

(i) 11111111

(ii) 01111111

(iii) 00000000

+ 1 ~~00~~

00000001

11111110

b. Es seien die Zahlen $(-30)_{10}$ sowie $(82)_{10}$ gegeben.

i) Geben Sie die Zweierkomplement-Darstellung der beiden Zahlen an. Verwenden Sie hierfür jeweils 8 Bits.

ii) Addieren Sie die beiden Zahlen binär. Der Rechenweg muss klar ersichtlich sein!

(i) $(-30)_{10}$

$(30)_2 = 11110$

$= 00011110$

$(82)_2 = 1010110$

$= 01010110$

$(-30)_2 = 11100001$

$+ \; 1$

$= 11100010$

82

(ii)

Tutor 8

| x | y | 0 | AND | $x\overline{y}$ | x | $\overline{x}y$ | y | $\nleftrightarrow$ | OR | NOR | = | $\overline{y}$ | $\overline{\overline{x}y}$ | $\overline{x}$ | $\overline{x\overline{y}}$ | NAND | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| | | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ |

## Aufgabe 1: (T) Boolesche Aussagen

(- Pkt.)

In dieser Aufgabe sind Beispiele für aussagenlogische Ausdrücke $z = f(x, y)$ gegeben. Stellen Sie für jedes Beispiel die Wahrheitstabelle auf und ordnen Sie dem Beispiel eine der 16 zweistelligen Boolesche Funktionen von Seite 52 des Skriptes zu! Entscheiden Sie zudem, ob es günstiger wäre, die Funktion in DNF oder KNF anzugeben und geben Sie die jeweilige DNF oder KNF an!

a. x bedeutet: Es regnet.
y bedeutet: Ich habe einen Schirm dabei.
z bedeutet: Ich kann nach draußen gehen, ohne nass zu werden.

| x | y | z |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 0 | 1 |
| 1 | 1 | 1 |

$z = \overline{\overline{x}y}$

DNF $= \overline{x} + y$

b. x bedeutet: Es ist ein Gang eingelegt (die Kupplung soll nicht beachtet werden).
y bedeutet: Das Gaspedal wird betätigt.
z bedeutet: Das Fahrzeug bewegt sich nach vorn.

| x | y | z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$z = xy$ KNF $= xy$

e. x bedeutet: Team X zieht am Tau.
y bedeutet: Team Y zieht am Tau.
z bedeutet: Es gewinnt eines der Teams (beide sind gleich stark) beim Tauziehen.

| x | y | z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

$z =$ or

KNF$=(x+y)$

c. x bedeutet: Es ist nicht windig.
y bedeutet: Die Sonne scheint.
z bedeutet: Ich kann einen Drachen steigen lassen.

| x | y | z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

$z = \overline{x}$

DNF $xy + x\overline{y}$

KNF $(x+\overline{y}) \cdot (x+y)$

d. x bedeutet: Der Zug kommt zu spät.
y bedeutet: Es steht ein Taxi als Alternativverbindung zur Verfügung.
z bedeutet: Ich komme zu spät zu meinem Termin.

| x | y | z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

$z = x\overline{y}$

DNF

$z = x\overline{y}$

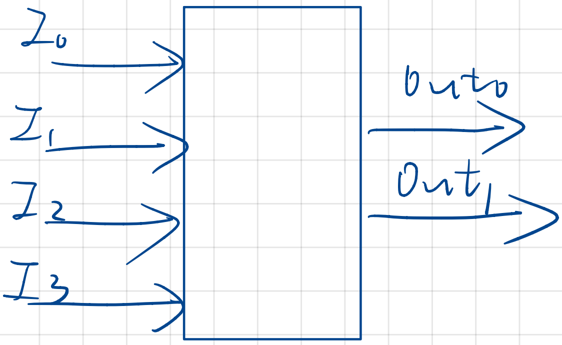
## Aufgabe 2: (T) Encoder

(- Pkt.)

Ein Encoder besitzt die umgekehrte Funktionalität eines Decoders. Er besitzt 4 Eingänge $I_0, I_1, I_2, I_3$ und die zwei Ausgänge $Out_0$ und $Out_1$. Es wird angenommen, dass stets genau einer der Eingänge mit einer 1 belegt ist. Ist ein Eingang $I_j$ mit einer 1 belegt, so ist $(Out_1, Out_0)$ die duale Darstellung der Dezimalzahl $j$. Bearbeiten Sie dazu folgende Teilaufgaben:

a. Bestimmen Sie die Schaltfunktion des 4-zu-2-Encoders. Verwenden Sie dabei die Bezeichnungen gemäß der obigen Beschreibung.

b. Zeichnen Sie das Schaltnetz eines 4-zu-2-Encoders gemäß der obigen Beschreibung eines 4-zu-2-Encoders! *Hinweis:* Die Erstellung der zugehörigen Wahrheitstabelle kann hierbei hilfreich sein.

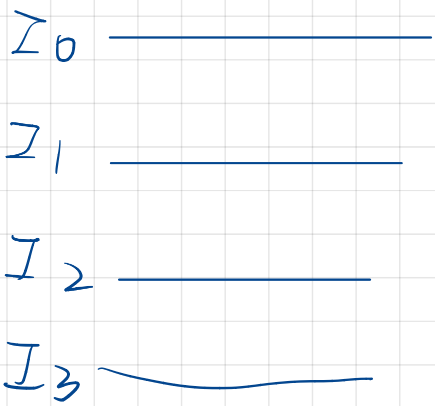



| $I_0$ | $I_1$ | $I_2$ | $I_3$ | $out_1$ | $out_0$ |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 |

a). $out_0 = I_1 \vee I_3$

$out_1 = I_2 \vee I_3$

b).

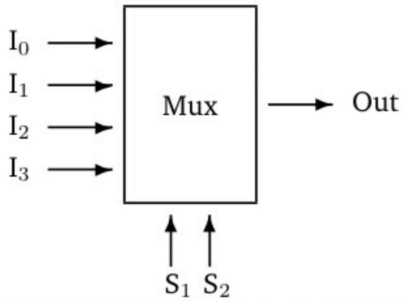


## Aufgabe 3: (T) Multiplexer

(- Pkt.)

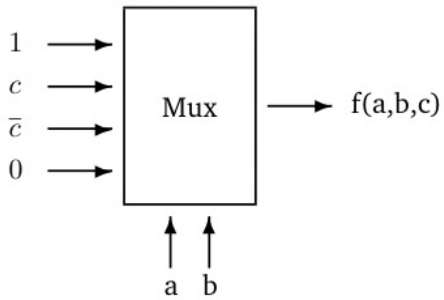
Für einen 4-Eingaben-Multiplexer gilt folgende verkürzte Funktionstabelle:

| $S_1$ | $S_2$ | Out |
|---|---|---|
| 0 | 0 | $I_0$ |
| 0 | 1 | $I_1$ |
| 1 | 0 | $I_2$ |
| 1 | 1 | $I_3$ |




Mit Hilfe eines 4-Eingaben-Multiplexers kann die Boolesche Funktion $f(a, b, c)$ dargestellt werden, indem dessen Eingänge bzw. Steuerleitungen wie folgt belegt werden.

| | $a$ | $b$ | $c$ | $f(a,b,c)$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 0 |



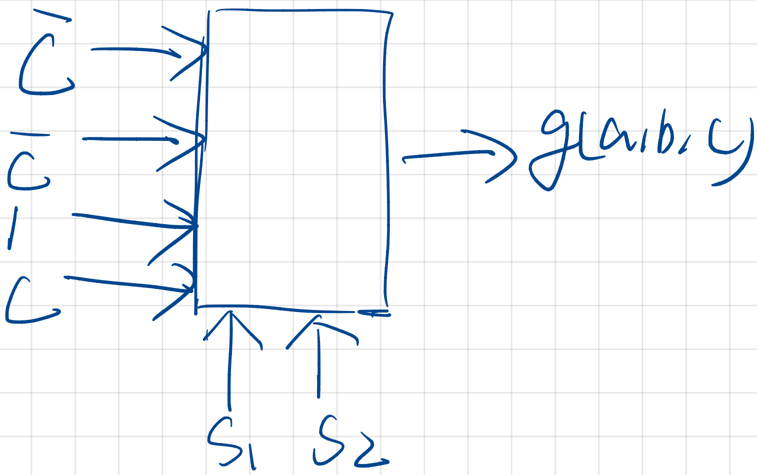

Geben Sie analog zum Beispiel eine Belegung der Eingänge eines 4-Eingaben-Multiplexers ($I_0, \ldots, I_3$) sowie der Steuerleitungen $S_1$ und $S_2$ an, sodass dieser die Boolesche Funktion

$$g(a, b, c) = (\bar{a} \cdot \bar{b} \cdot \bar{c}) + (\bar{a} \cdot b \cdot \bar{c}) + (\bar{a} \cdot b \cdot c) + (a \cdot \bar{b} \cdot c) + (a \cdot b \cdot c)$$

realisiert. Maxtem

Sie dürfen ausschließlich die Werte $a, b, c, \bar{c}$ sowie $0$ und $1$ benutzen. Es dürfen keine weiteren Bausteine oder Gatter verwendet werden.

| a | b | c | g(a,b,c) |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |



Tutor 9

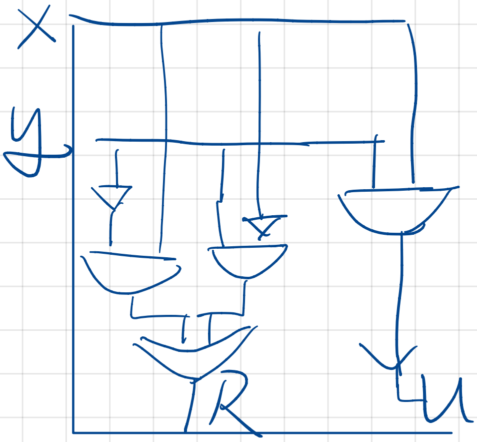
## Aufgabe 1: (T) Entwurf eines 4-Bit-Addiernetzes

(- Pkt.)

Es soll systematisch ein Addiernetz entworfen werden, das in der Lage ist, zwei 4-stellige Dualzahlen zu addieren. Dazu wird das Problem aufgespaltet, indem man überlegt, wie eine einzelne Stelle addiert wird.

a. Entwerfen Sie einen Halbaddierer, der in der Lage ist, zwei einstellige Dualzahlen zu addieren.

| x | y | R | V |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |




b. Entwerfen Sie einen Volladdierer, der in der Lage ist, eine beliebige Stelle zweier n-stelliger Dualzahlen zu addieren.

| x | y | u | R | U |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

c. Entwerfen Sie nun das Addiernetz, indem Sie Halb- und Volladierer verwenden.

## **Aufgabe 2: (T)** Einfache ALU

(- Pkt.)

Entwerfen Sie eine einfache 1-Bit ALU, die den folgenden Spezifikationen genügt:

a. Operationen: `AND`, `OR`, Addition und Subtraktion.

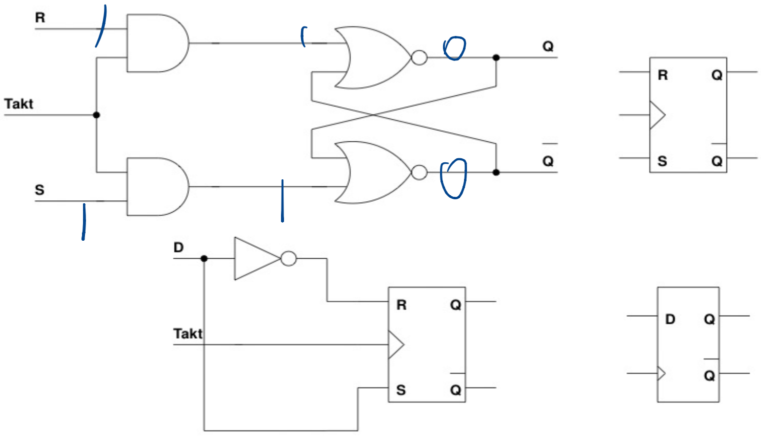
b. Inputs: Operanden $a$ und $b$, CarryIn (Übertrag aus einer vorgeschalteten ALU), gewisse Steuerleitungen (z.B. zur Auswahl des Typs der Operation).

c. Outputs: Resultat, CarryOut (Übertrag).

## Aufgabe 1: (T) Arbeitsweise von Latches

(- Pkt.)

Betrachten Sie die folgenden Schaltbilder eines RS- und eines D-Latches:




a. Machen Sie sich die Funktionsweise der Latches klar, indem Sie die Zustandstabellen aufstellen. Jede Tabelle soll folgendermaßen aufgebaut sein:

– Jede Spalte entspricht einem Ein- bzw. Ausgang. Ein RS-Latch zum Beispiel verfügt über die drei Eingänge $S$, $R$ und $C$ (Clock/Takt), sowie über die Ausgänge $Q$ und $\overline{Q}$.

– Jede Zeile entspricht einem bestimmten Zustand des Latches, abhängig von den Signalen an den Eingängen.

– Mögliche Zustände sind: Set, Reset, Speichern und Kippen. Geben Sie hinter jeder Tabellenzeile an, welcher Zustand vorliegt. (Hinweis: Nicht alle Zustände kommen bei jedem Latch vor.)

– Kennzeichnen Sie unzulässige Zustände als solche.

– Verwenden Sie *Don't-Care-Argumente*, falls es keinen Unterschied für die Belegung der Ausgänge macht, ob an einem Eingang eine 0 oder 1 anliegt. Verwenden Sie zur Kennzeichnung solcher Belegungen in der Tabelle ein $D$.

– Verwenden Sie die Notation $Q^*$, um den alten Wert von $Q$ zu symbolisieren, falls dieser in diesem Zustand nicht explizit (0 oder 1) bekannt ist.

| R | S | C | Q | $\overline{Q}$ | |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | set |
| 1 | 0 | 1 | 0 | 1 | Reset |
| 0 | 0 | 1 | $Q^*$ | $Q^*$ | speichern |
| 1 | 1 | 1 | — | — | unzlässig |
| D | D | 0 | $Q^*$ | $Q^*$ | speichern |

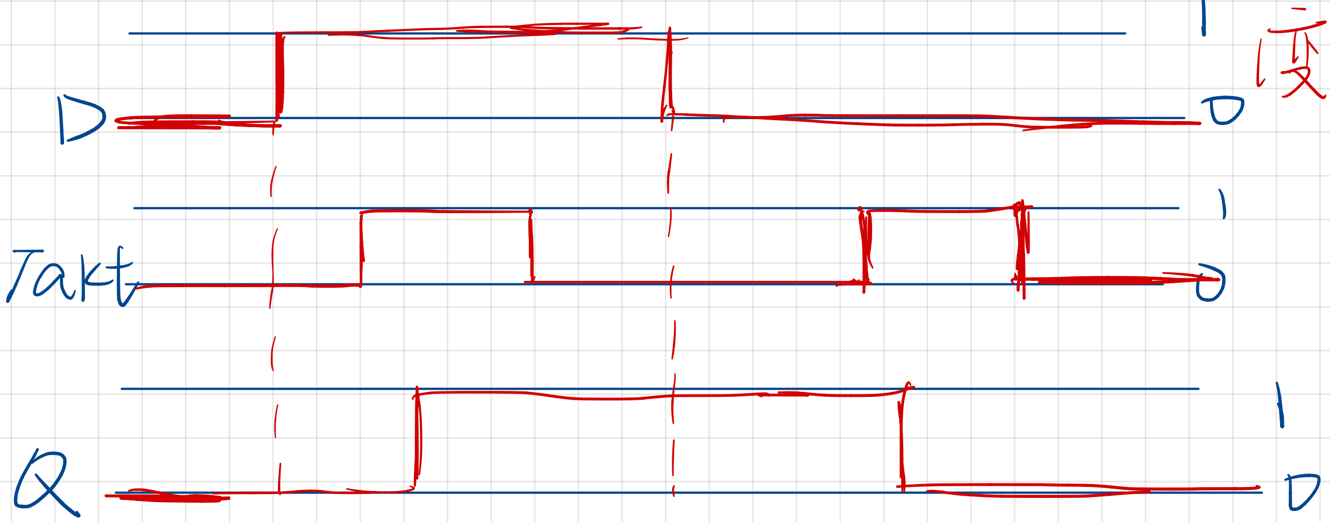
b. Welchen Vorteil besitzt das D-Latch gegenüber dem RS-Latch?

| D | R | S | Q | $\overline{Q}$ |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |

Das D-Latch hat keinen unzulässigen Zustand.

c. Welches Problem ergibt sich aber beim D-Latch im Hinblick auf das Speichern über mehrere Takte hinweg? Verdeutlichen Sie das Problem durch ein Impulsdiagramm, das die Verläufe der Signale $D$, $C$ (Clock/Takt) und $Q$ darstellt. Setzen Sie alle drei Signale anfangs auf 0, zeichnen Sie dann zunächst den Verlauf für das Taktsignal und führen Sie anschließend eine Set-Operation durch.

? 这里D为什么变成0

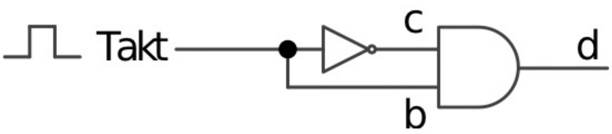


## Aufgabe 2: (T) Flip-Flops

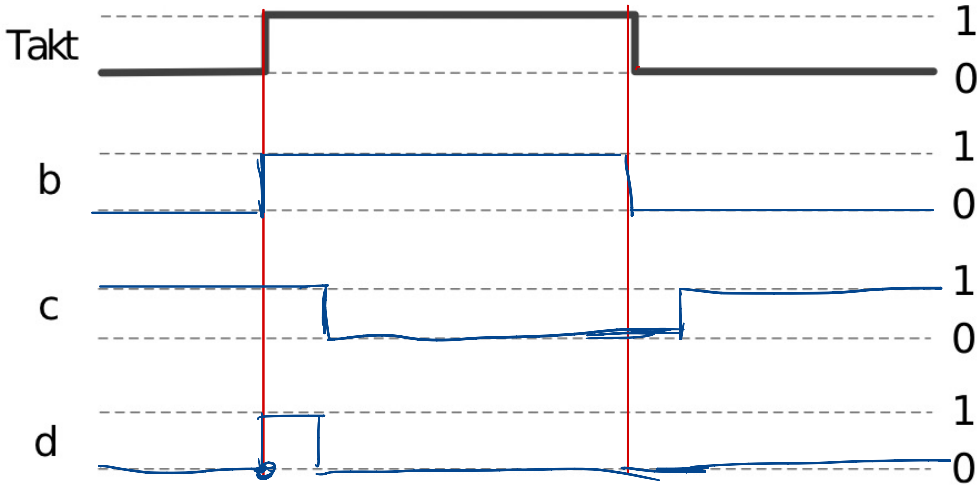
(- Pkt.)

Bearbeiten Sie die folgenden Teilaufgaben zum Thema Flip-Flops:

a. Gegeben sei folgendes Schaltnetz, welches einen Impulsgenerator realisiert, der aus Taktflanken kurze Impulse erzeugt:

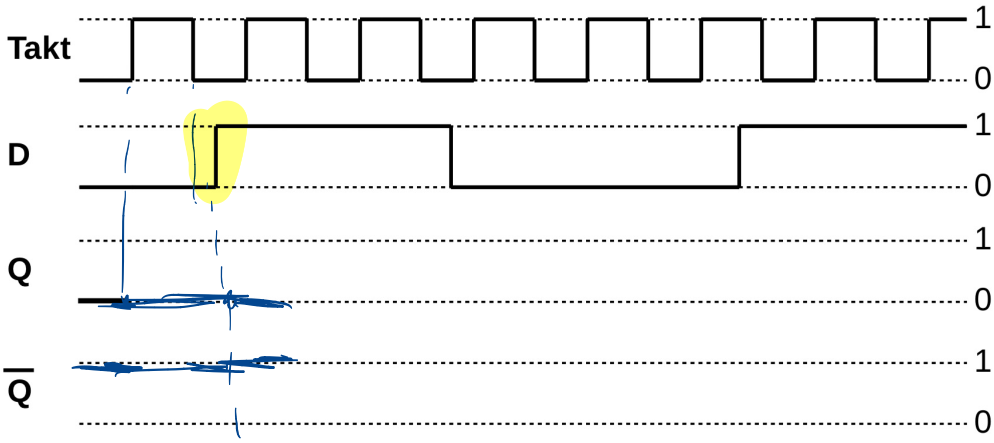



Ergänzen Sie folgende Vorlage zu einem Impulsdiagramm für die Ausschnitte $b$, $c$, $d$ basierend auf dem eingezeichnetem Takt. Gehen Sie davon aus, dass das AND–Gatter keine Verzögerung verursacht und das NOT–Gatter eine nicht vernachlässigbare Verzögerung verursacht, deren Auswirkungen im Impulsdiagramm deutlich werden müssen:




b. Gegeben sei das nachfolgende Impulsdiagramm eines D-Flip-Flops mit dem Taktgeber aus der vorherigen Teilaufgabe a. Vervollständigen Sie das folgende Impulsdiagramm für die Ausgänge $Q$ und $\bar{Q}$ unter der Annahme, dass der Baustein ohne Zeitverzögerung schaltet:

谁有 verzögerung →




?

kehrt.

**Beispiel 7.18.** *Betrachte das vorangegangene Beispiel:*

$$f(x_1, x_2, x_3, x_4) = x_1\bar{x}_2x_3x_4 + x_1\bar{x}_2\bar{x}_3x_4 + x_1x_2x_3x_4 + \bar{x}_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_2x_3x_4$$

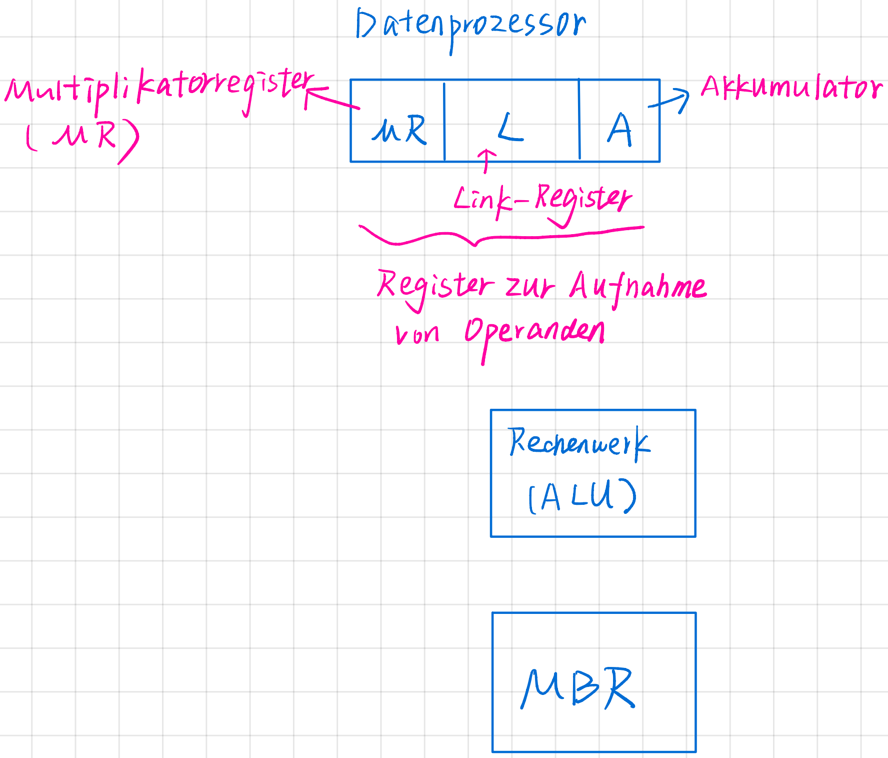
| $x_3x_4 \backslash x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 10 | | | | |
| 11 | 1 | | 1 | 1 |
| 01 | 1 | | | 1 |
| 00 | | | | |

5

4er Block

Abbildung 7.23: Beispiel für die in 7.16 gegebene Funktion.

Bild blöcke:


Datenprozessor
Multiplikatorregister
(MR)
MR
L
A
Akkumulator
Link-Register
Register zur Aufnahme von Operanden
Rechenwerk
(ALU)
MBR

Station 1 {bohren, schrauben}

Station 2 {lackieren}

Station 3 {polieren}

Rob A {Schrau. bohren. lackieren}

R B {Schneiden polieren}

R C {lackieren}

{-3, -1, 0, 5,

| | A1 | A2 | A3 | B1 | B2 | B3 | C1 | C2 | C3 |
|---|---|---|---|---|---|---|---|---|---|
| A1 | -3 | 5 | 5 | A∨B‡ ~~-5~~ 5 | A∨B‡ ~~-5~~ 0 5 | ~~-3~~ 0 5 | ~~-1~~ 5 | ~~-3~~ 0 | ~~-1~~ 0 5 |
| A2 | | -3 | 5 | ~~-1~~ 0 | ~~-3~~ 0 | ~~-3~~ 0 | -1 | ~~-3~~ 0 | -1 |
| A3 | | | -1 | 0 | | | | | |
| B1 | | | | -1 | -3 | -3 | | | |
| B2 | | | | | -1 | -3 | | | |
| B3 | | | | | | -3 | | | |
| C1 | | | | | | | -1 | | |
| C2 | | | | | | | | -3 | |
| C3 | | | | | | | | | -1 |

Ludwig-Maximilians-Universität München
Institut für Informatik
Lehrstuhl für Mobile und Verteilte Systeme
Prof. Dr. Linnhoff-Popien

mobile and distributed systems group

# Übungsblatt 5
# Rechnerarchitektur im Sommersemester 2023

### Zu den Modulen C, D, F, K

**Abgabetermin:** 28.05.23, 18:00 Uhr
**Besprechung:** 30.05.23 bis 02.06.23

## Aufgabe Ü1: NAND/NOR

(6 Pkt.)

Die beiden Mengen {NAND} und {NOR} von Booleschen Funktionen sind funktional vollständig, d.h. dass sich durch die Kombination von NAND- bzw. NOR-Funktionen jede beliebige Boolesche Funktion darstellen lässt. Dies ermöglicht es, NAND- bzw. NOR-Gatter kostengünstig in Massenproduktion herzustellen und daraus beliebige digitale Schaltungen aufzubauen.

NAND

| $i$ | $a$ | $b$ | $\overline{a \cdot b} = a$ NAND $b$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 0 |

NOR

| $i$ | $a$ | $b$ | $\overline{a + b} = a$ NOR $b$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 1 | 1 | 0 |

Stellen Sie die elementaren Booleschen Funktionen AND, OR und NOT jeweils unter ausschließlicher Verwendung von

a. NAND-Gattern

b. NOR-Gattern

dar!

## Aufgabe Ü2: PLA-Entwurf

(8 Pkt.)

In der Vorlesung haben Sie das Konzept von programmierbaren logischen Arrays (PLAs) kennen gelernt.
Wenn man in einem PLA die Anordnung der Bausteine so vornimmt, dass in der oberen Hälfte nur Bausteine vom Typ 0, 2 oder 3 und in der unteren Hälfte nur Bausteine vom Typ 0 oder 1 exisitieren – man das PLA also in eine Und- und eine Oder-Ebene unterteilen kann – spricht man auch von einem *normierten PLA*.
Gegeben sei die folgende Boolesche Funktion $f : B^3 \to B^2$

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

Realisieren Sie diese Funktion durch ein normiertes PLA, welches aus der minimal möglichen Anzahl an Zeilen und Spalten besteht. Verwenden Sie ausschließlich Bausteine der oben gennanten Typen 0 bis 3. Kennzeichnen Sie in Ihrer Skizze die Und- und die Oder-Ebene. Markieren Sie gesperrte und neutralisierte Eingänge. Beschriften Sie jeden Pfeil (sowohl ausgehende als auch die innerhalb des PLAs) mit der jeweils anliegenden logischen Funktion.

## Aufgabe Ü3: Cäsar-Verschlüsselung unter SPIM

(6 Pkt.)

Bearbeiten Sie die folgende Aufgabe zum Thema Assemblerprogrammierung unter SPIM.
**Hinweis:** Eine Übersicht der SPIM-Befehle finden Sie am Ende des Übungsblatts.
Im Folgenden soll ein MIPS-Assembler Programm vervollständigt werden, welches einen gegebenen Text mittels der **Caesar-Verschlüsselung** in einen Geheimtext umwandelt. Bei der Caesar-Verschlüsselung wird jeder Buchstabe im zu verschlüsselnden Text um eine vorher festgelegte Distanz im Alphabet verschoben. Ist z.B. die Distanz 3, so wird der Buchstabe A zum Buchstaben D, der Buchstabe B zum Buchstaben E, ..., der Buchstabe Z zum Buchstaben C.
Das folgende MIPS-Assembler Programm erwartet als Nutzereingabe die Distanz, um die die Buchstaben verschoben werden sollen und verschlüsselt dann einen gegebenen Text.
Ergänzen Sie den unten angegebenen Coderahmen um insgesamt **6 Zeilen Code**, so dass das Programm wie beschrieben funktioniert. Tragen Sie Ihre Lösung unter den mit "`# Ihre Loesung:`" markierten Stellen direkt in den folgenden Coderahmen ein:

```
.data

shift_text: .asciiz "Um wieviele Stellen soll der Text verschoben werden: "
string1: .asciiz "Der verschluesselte Text lautet:  "
secret: .asciiz "GEHEIMNIS"
string_a: .asciiz "A"
string_z: .asciiz "Z"

result: .space 9

.text
main:
        # t0  - Zum Zwischenspeichern der Position des aktuell betrachteten Buchstabens
        # t1 - Gibt die Laenge des Geheimworts an
        # t2 - ASCII Wert des Buchstaben A (65)
        # t3 - ASCII - WERT des Buchstaben Z (90)
        li  $t0, 0
        li  $t1, 9
        lb  $t2, string_a
        lb  $t3, string_z

        la $a0, shift_text      # String mit Anfangsadresse shift_text in $a0 laden
        li $v0, 4               # 4 in $v0 laden
        syscall                 # Text mit Anfangsadreesse in $a0 auf der Konsole ausgeben

        li $v0, 5               # 5 in $v0 laden
        syscall                 # Zahl eingeben

        move $s1, $v0           # Eingegebene Zahl in $s1 speichern

loop:   bge $t0, $t1, end       # Falls alle Buchstaben betrachtet wurden -> Springe end
        lb  $t4, secret($t0)    # Lade den aktuellen Buchstaben in $t4

caesar: ####################################
        # Fuegen Sie hier Ihre Loesung ein #
```

```
        ####################################



        ######################
        # Ende Ihrer Loesung #
        ######################
        bgt $t4, $t3, cadd      # Falls das Ergebnis > Z --> springe zu cadd

save:   ####################################
        # Fuegen Sie hier Ihre Loesung ein #
        ####################################






        ######################
        # Ende Ihrer Loesung #
        ######################
        j loop                  # Springe zum Label loop

cadd:   ####################################
        # Fuegen Sie hier Ihre Loesung ein #
        ####################################







        ######################
        # Ende Ihrer Loesung #
        ######################
        j save                  # Springe zum Label save


end:    la $a0, string1         # Anfangsadresse des Strings string1 wird in $a0 geladen
        li $v0, 4               # 4 wird in $v0 geladen
        syscall                 # String string1 wird auf der Konsole ausgeben

        la $a0, result          # Anfangsadresse des Strings result wird in $a0 geladen
        li $v0, 4               # 4 wird in $v0 geladen
        syscall                 # String result  wird auf der Konsole ausgeben

        li $v0, 10              # 10 wird in $v0 geladen
        syscall                 # Programm wird beendet
```

## Aufgabe Ü4: Einfachauswahlaufgabe: Wiederholung

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n“). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

| | | | |
|---|---|---|---|
| a) Bei welcher Belegung $(x_1, x_2, x_3)$ ergibt die Boolesche Funktion $f(x_1, x_2, x_3) = (x_1 \cdot \overline{x}_2) + (x_2 \cdot \overline{x_3})$ den Wert 1? | | | |
| (i) $(0, 0, 1)$ | (ii) $(0, 1, 0)$ | (iii) $(1, 1, 1)$ | (iv) $(0, 1, 1)$ |
| | | | |
| b) Was bewirkt der Spim-Befehl `li $v0 5`: | | | |
| (i) Es wird ein Integer von der Konsole eingelesen | (ii) Es wird eine Zahl vom Typ double von der Konsole eingelesen | (iii) Der Wert 5 wird in das Register `$v0` geladen | (iv) Es wird ein Integer auf der Konsole ausgegeben |
| | | | |
| c) Wofür steht CISC im Zusammenhang mit Mikroprozessoren? | | | |
| (i) Complex Instruction Set Computer | (ii) Complex Instruction Set Calculator | (iii) Controversy Instruction Set Computer | (iv) Constructive Instruction Set Computer |
| | | | |
| d) Welche Aussage ist korrekt? MIPS ist eine... | | | |
| (i) Last-in-First-Out-Architektur | (ii) Stack-Architektur | (iii) Load-Store-Architektur | (iv) Heap-Architektur |
| | | | |
| e) Welche Aussage ist falsch? Die Funktion `syscall`... | | | |
| (i) beendet das Programm sofort | (ii) erwartet die Nummer der auszuführenden Funktion in `$v0` | (iii) besitzt selbst keine Parameter | (iv) führt eine Funktion des Betriebssystems aus |
| | | | |

## **Aufgabe Ü1:** NAND/NOR

(6 Pkt.)

Die beiden Mengen {`NAND`} und {`NOR`} von Booleschen Funktionen sind funktional vollständig, d.h. dass sich durch die Kombination von `NAND`- bzw. `NOR`-Funktionen jede beliebige Boolesche Funktion darstellen lässt. Dies ermöglicht es, `NAND`- bzw. `NOR`-Gatter kostengünstig in Massenproduktion herzustellen und daraus beliebige digitale Schaltungen aufzubauen.

NAND

| $i$ | $a$ | $b$ | $\overline{a \cdot b} = a$ NAND $b$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 0 |

NOR

| $i$ | $a$ | $b$ | $\overline{a + b} = a$ NOR $b$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 1 | 1 | 0 |

Stellen Sie die elementaren Booleschen Funktionen `AND`, `OR` und `NOT` jeweils unter ausschließlicher Verwendung von

a. `NAND`-Gattern

b. `NOR`-Gattern

dar!

A $\longrightarrow$

## **Aufgabe Ü2:** PLA-Entwurf

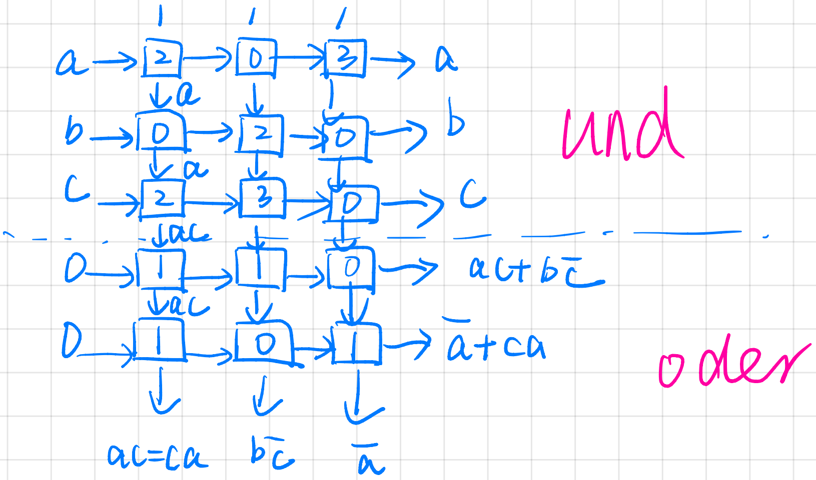
(8 Pkt.)

In der Vorlesung haben Sie das Konzept von programmierbaren logischen Arrays (PLAs) kennen gelernt.
Wenn man in einem PLA die Anordnung der Bausteine so vornimmt, dass in der oberen Hälfte nur Bausteine vom Typ 0, 2 oder 3 und in der unteren Hälfte nur Bausteine vom Typ 0 oder 1 exisitieren – man das PLA also in eine Und- und eine Oder-Ebene unterteilen kann – spricht man auch von einem *normierten PLA*.
Gegeben sei die folgende Boolesche Funktion $f : B^3 \to B^2$

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

Realisieren Sie diese Funktion durch ein normiertes PLA, welches aus der minimal möglichen Anzahl an Zeilen und Spalten besteht. Verwenden Sie ausschließlich Bausteine der oben gennanten Typen 0 bis 3. Kennzeichnen Sie in Ihrer Skizze die Und- und die Oder-Ebene. Markieren Sie gesperrte und neutralisierte Eingänge. Beschriften Sie jeden Pfeil (sowohl ausgehende als auch die innerhalb des PLAs) mit der jeweils anliegenden logischen Funktion.

$f(a,b,c) = (ac + b\bar{c},\ a + ca)$



## **Aufgabe Ü3:** Cäsar-Verschlüsselung unter SPIM

(6 Pkt.)

Bearbeiten Sie die folgende Aufgabe zum Thema Assemblerprogrammierung unter SPIM.
**Hinweis:** Eine Übersicht der SPIM-Befehle finden Sie am Ende des Übungsblatts.
Im Folgenden soll ein MIPS-Assembler Programm vervollständigt werden, welches einen gegebenen Text mittels der **Caesar-Verschlüsselung** in einen Geheimtext umwandelt. Bei der Caesar-Verschlüsselung wird jeder Buchstabe im zu verschlüsselnden Text um eine vorher festgelegte Distanz im Alphabet verschoben. Ist z.B. die Distanz 3, so wird der Buchstabe A zum Buchstaben D, der Buchstabe B zum Buchstaben E, ..., der Buchstabe Z zum Buchstaben C.
Das folgende MIPS-Assembler Programm erwartet als Nutzereingabe die Distanz, um die die Buchstaben verschoben werden sollen und verschlüsselt dann einen gegebenen Text.
Ergänzen Sie den unten angegebenen Coderahmen um insgesamt **6 Zeilen Code**, so dass das Programm wie beschrieben funktioniert. Tragen Sie Ihre Lösung unter den mit "`# Ihre Loesung:`" markierten Stellen direkt in den folgenden Coderahmen ein:

```
.data

shift_text: .asciiz "Um wieviele Stellen soll der Text verschoben werden: "
string1: .asciiz "Der verschluesselte Text lautet:  "
secret: .asciiz "GEHEIMNIS"
string_a: .asciiz "A"
string_z: .asciiz "Z"

result: .space 9

.text
main:
        # t0  - Zum Zwischenspeichern der Position des aktuell betrachteten Buchstabens
        # t1 - Gibt die Laenge des Geheimworts an
        # t2 - ASCII Wert des Buchstaben A (65)
        # t3 - ASCII - WERT des Buchstaben Z (90)
        li  $t0, 0
        li  $t1, 9
        lb  $t2, string_a
        lb  $t3, string_z

        la $a0, shift_text      # String mit Anfangsadresse shift_text in $a0 laden
        li $v0, 4               # 4 in $v0 laden
        syscall                 # Text mit Anfangsadreesse in $a0 auf der Konsole ausgeben

        li $v0, 5               # 5 in $v0 laden
        syscall                 # Zahl eingeben

        move $s1, $v0           # Eingegebene Zahl in $s1 speichern

loop:   bge $t0, $t1, end       # Falls alle Buchstaben betrachtet wurden -> Springe end
        lb  $t4, secret($t0)    # Lade den aktuellen Buchstaben in $t4

caesar: ####################################
        # Fuegen Sie hier Ihre Loesung ein #
        ####################################



        ######################
        # Ende Ihrer Loesung #
        ######################
        bgt $t4, $t3, cadd      # Falls das Ergebnis > Z --> springe zu cadd

save:   ####################################
        # Fuegen Sie hier Ihre Loesung ein #
        ####################################






        ######################
        # Ende Ihrer Loesung #
        ######################
        j loop                  # Springe zum Label loop

cadd:   ####################################
        # Fuegen Sie hier Ihre Loesung ein #
        ####################################







        ######################
        # Ende Ihrer Loesung #
        ######################
        j save                  # Springe zum Label save


end:    la $a0, string1         # Anfangsadresse des Strings string1 wird in $a0 gelac
        li $v0, 4               # 4 wird in $v0 geladen
        syscall                 # String string1 wird auf der Konsole ausgeben

        la $a0, result          # Anfangsadresse des Strings result wird in $a0 gelade
        li $v0, 4               # 4 wird in $v0 geladen
        syscall                 # String result  wird auf der Konsole ausgeben

        li $v0, 10              # 10 wird in $v0 geladen
        syscall                 # Programm wird beendet
```

## **Aufgabe Ü4:** Einfachauswahlaufgabe: Wiederholung

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n"). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

| | | | |
|---|---|---|---|
| a) Bei welcher Belegung $(x_1, x_2, x_3)$ ergibt die Boolesche Funktion $f(x_1, x_2, x_3) = (x_1 \cdot \overline{x}_2) + (x_2 \cdot \overline{x_3})$ den Wert 1? | | | |
| (i) $(0, 0, 1)$ | (ii) $(0, 1, 0)$ | (iii) $(1, 1, 1)$ | (iv) $(0, 1, 1)$ |
| | | | |
| b) Was bewirkt der Spim-Befehl `li $v0 5`: | | | |
| (i) Es wird ein Integer von der Konsole eingelesen | (ii) Es wird eine Zahl vom Typ double von der Konsole eingelesen | (iii) Der Wert 5 wird in das Register `$v0` geladen | (iv) Es wird ein Integer auf der Konsole ausgegeben |
| | | | |
| c) Wofür steht CISC im Zusammenhang mit Mikroprozessoren? | | | |
| (i) Complex Instruction Set Computer | (ii) Complex Instruction Set Calculator | (iii) Controversy Instruction Set Computer | (iv) Constructive Instruction Set Computer |
| | | | |
| d) Welche Aussage ist korrekt? MIPS ist eine... | | | |
| (i) Last-in-First-Out-Architektur | (ii) Stack-Architektur | (iii) Load-Store-Architektur | (iv) Heap-Architektur |
| | | | |
| e) Welche Aussage ist falsch? Die Funktion `syscall`... | | | |
| (i) beendet das Programm sofort | (ii) erwartet die Nummer der auszuführenden Funktion in `$v0` | (iii) besitzt selbst keine Parameter | (iv) führt eine Funktion des Betriebssystems aus |
| | | | |

$X_2 X_1 + X_2 X_1 = X_1 X_2$

**Beispiel 7.15** (Resolutionsregel). *Gegeben sei die Boolesche Funktion*

$$\begin{aligned} f(x_1, x_2, x_3) &= \overline{x_1} x_2 x_3 + x_1 x_2 x_3 \\ &= (\overline{x_1} + x_1) x_2 x_3 \quad \textit{nach Komplementärgesetz gilt: } \overline{x} + x = 1 \textit{ und } 1 \wedge x = x \\ &= x_2 x_3 \end{aligned}$$

*Diese Vereinfachung ist auch unter dem Namen* ***Resolutionsregel*** *bekannt, d.h. kommen in einer disjunktiven Form zwei Summanden vor, welche sich in* genau einer *komplementären Variablen unterscheiden, so können diese beiden Summanden durch den ihnen gemeinsamen Teil ersetzt werden.*

**Beispiel 7.16** (mehrfache Anwendung der Resolutionsregel). *Betrachte*

$$\begin{aligned} f(x_1, x_2, x_3, x_4) &= x_1 \bar{x}_2 x_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 x_2 x_3 x_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 \\ &= x_1 \bar{x}_2 x_4 + x_1 x_3 x_4 + \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 x_4 \\ &= \bar{x}_2 x_4 + x_1 x_3 x_4 + \bar{x}_2 \bar{x}_3 x_4 \end{aligned}$$

$x_1 \bar{x}_2 x_3 x_4$

*Die mehrfache Anwendung beruht dabei auf der Gesetzmäßigkeit x+x=x, die das Verdoppeln von Summanden erlaubt.*

Um die Übersicht über alle möglichen Resolutionen zu behalten, wollen wir ein graphisches Verfahren betrachten, mit dem eine Schaltfunktion recht leicht vereinfacht werden kann.

(4 Pkt.)

Vereinfachen Sie den folgenden Booleschen Term unter Anwendung der Resolutionsregel soweit wie möglich:

$$f(x_1, x_2, x_3, x_4) = \overline{x}_1 \overline{x}_2 \overline{x}_3 \overline{x}_4 + \overline{x}_1 \overline{x}_2 \overline{x}_3 x_4 + \overline{x}_1 x_2 x_3 x_4 + x_1 x_2 \overline{x}_3 x_4 + x_1 x_2 x_3 x_4$$

Geben Sie dabei **alle** notwendigen Zwischenschritte an!

$$\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 x_3 x_4$$
$$+ x_1 x_2 \bar{x}_3 x_4 + x_1 x_2 x_3 x_4$$
$$= \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 x_3 x_4 + x_1 x_2 \bar{x}_3 x_4 + x_1 x_2 x_3 x_4$$
$$= \bar{x}_1 \bar{x}_2 \bar{x}_3 + x_2 x_3 x_4 + x_1 x_2 \bar{x}_3 x_4$$

## Aufgabe Ü1: Resolutionsregel

(4 Pkt.)

Vereinfachen Sie den folgenden Booleschen Term unter Anwendung der Resolutionsregel soweit wie möglich:

$$f(x_1, x_2, x_3, x_4) = \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4 + \overline{x}_1\overline{x}_2\overline{x}_3x_4 + \overline{x}_1x_2x_3x_4 + x_1x_2\overline{x}_3x_4 + x_1x_2x_3x_4$$

Geben Sie dabei **alle** notwendigen Zwischenschritte an!

$$\overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4} + \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,x_4 + \overline{x_1}\,x_2x_3x_4 + x_1x_2\overline{x_3}\,x_4 + x_1x_2x_3x_4$$

$$= \overline{x_1}\,\overline{x_2}\,\overline{x_3} + x_2x_3x_4 + x_1x_2\overline{x_3}\,x_4 + x_1x_2x_3x_4$$

$$= \overline{x_1}\,\overline{x_2}\,\overline{x_3} + x_2x_3x_4 + x_1x_2x_4$$

## Aufgabe Ü2: Optimierung von Schaltnetzen

(11 Pkt.)

a. Gegeben sei folgende Wahrheitstabelle einer Funktion $f(x_1, x_2, x_3, x_4)$.
Leiten Sie aus dieser Wahrheitstabelle die Schaltfunktion in ihrer vollständigen konjunktiven Normalform (**KNF**) her.

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $f(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | 1 |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 1 |
| 13 | 1 | 1 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 | 1 |

a) $(x_1+x_2+x_3+\overline{x_4})\cdot(x_1+x_2+\overline{x_3}+\overline{x_4})\cdot(x_1+\overline{x_2}+x_3+x_4)\cdot(x_1+\overline{x_2}+\overline{x_3}+x_4)\cdot(\overline{x_1}+x_2+\overline{x_3}+\overline{x_4})\cdot$
$(x_1+x_2+x_3+x_4)$

b. Im Folgenden ist die Wahrheitstabelle der Funktion $g(x_1, x_2, x_3, x_4)$ gegeben. Minimieren Sie die Funktion $g$ unter Verwendung eines Karnaugh-Diagramms grafisch. Kennzeichnen Sie **alle** Blöcke innerhalb Ihres Karnaugh-Diagramms, die Sie für Ihre Vereinfachung verwenden! Geben Sie abschließend die minimierte Funktion in disjunktiver Form an!

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $g(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 |
| 6 | 0 | 1 | 1 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 |
| 10 | 1 | 0 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 0 | 0 |
| 15 | 1 | 1 | 1 | 1 | 1 |

| $x_3x_4$ \ $x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 10 | | 1 | | |
| 11 | | 1 | 1 | |
| 01 | 1 | | 1 | |
| 00 | 1 | | | |

$$f(x_1, x_2, x_3, x_4) = \overline{x}_1 \overline{x}_2 \overline{x}_3 + \overline{x}_1 x_2 x_3 + x_1 x_2 x_4$$

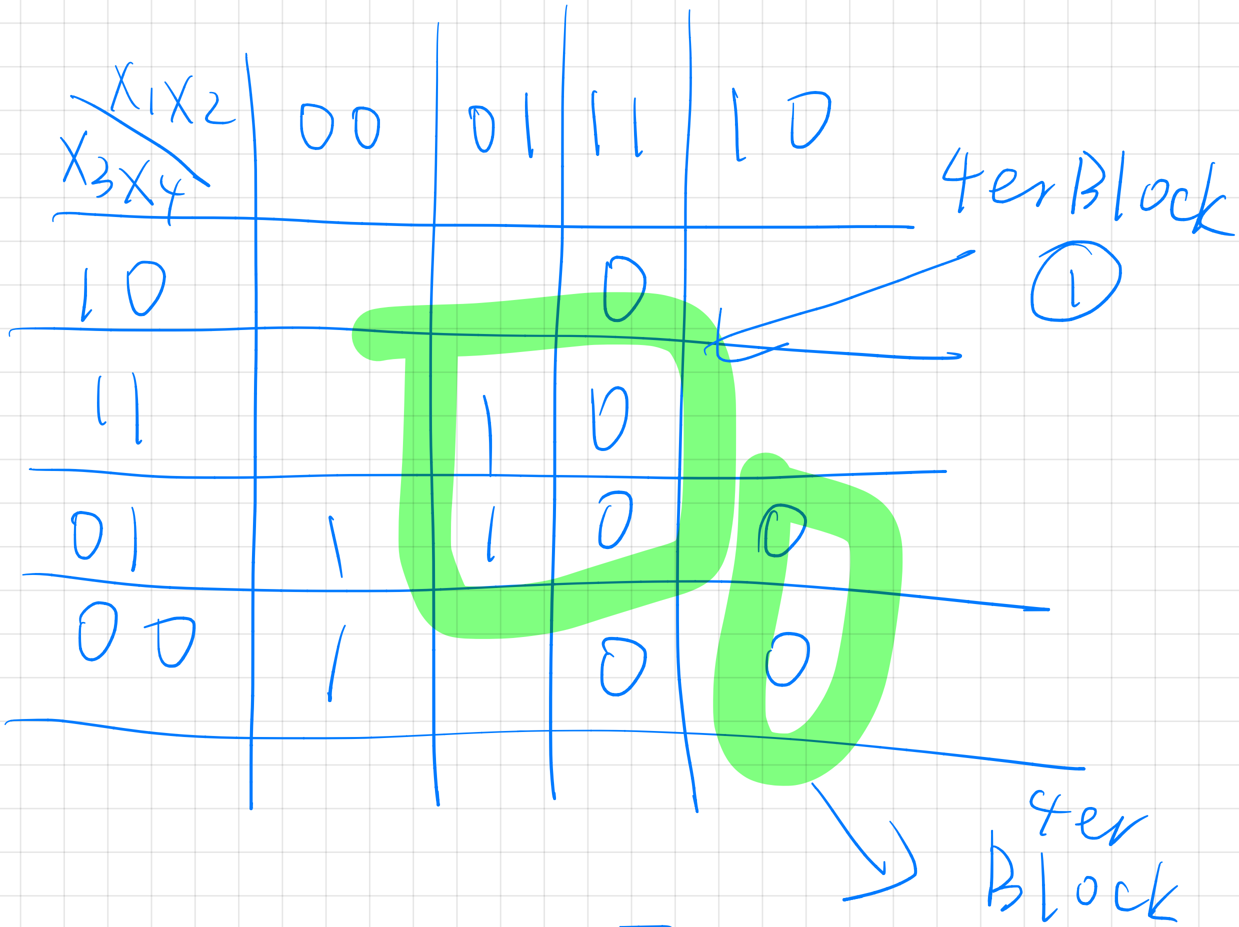
## Aufgabe Ü3: Karnaugh

(9 Pkt.)

Gegeben sei die Wahrheitstabelle einer partiellen Booleschen Funktion $g(x_1, x_2, x_3, x_4)$. Undefinierte Ausgaben sind mit einem **D** gekennzeichnet:

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $g(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | **D** |
| 9 | 1 | 0 | 0 | 1 | **D** |
| 10 | 1 | 0 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | **D** |
| 13 | 1 | 1 | 0 | 1 | **D** |
| 14 | 1 | 1 | 1 | 0 | **D** |
| 15 | 1 | 1 | 1 | 1 | **D** |

Minimieren Sie die Funktion $g$ unter Verwendung eines Karnaugh-Diagramms grafisch. Beachten Sie dabei die **Don't Care** Argumente. Kennzeichnen Sie **alle** Blöcke innerhalb Ihres Karnaugh-Diagramms, die Sie für Ihre Vereinfachung verwenden! Geben Sie abschließend die minimierte Funktion in disjunktiver Form an!




$f(x_1 x_2 x_3 x_4) = x_2 x_4 + \overline{x}_2 \overline{x}_3$

## **Aufgabe Ü4:** Einfachauswahlaufgabe: Optimierung von Schaltnetzen

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n“). Nennen Sie dazu in Ihrer Abgabe die jeweils ausgewählte Antwortnummer ((i), (ii), (iii) oder (iv)). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

| | | | |
|---|---|---|---|
| a) Angenommen ein Multiplexer hat `512` (Nutz-)Eingänge. Wie viele Steuereingänge werden benötigt, um die (Nutz-)Eingänge einzeln selektieren zu können? | | | |
| (i) `3` | (ii) `9` | (iii) `256` | (iv) `512` |
| | | | |
| b) Wie viele Felder enthält das Karnaugh-Diagramm einer Booleschen Funktion $f : B^3 \to B$? | | | |
| (i) 4 | (ii) 8 | (iii) 1 | (iv) 2 |
| | | | |
| c) Es kann sein, dass nicht alle $2^n$ Argumente einer Booleschen Funktion $f : B^n \to B$ ($n \geq 1$) auftreten können. Wie bezeichnet man die Argumente einer solchen partiellen Funktion $f$, für die der Funktionswert nicht festgelegt ist? | | | |
| (i) Don't Worry | (ii) Don't Panic | (iii) Don't Know | (iv) Don't Cares |
| | | | |
| d) Die Reihenfolge der Beschriftung eines Karnaugh-Diagramms erfolgt so, dass sich zwei zyklisch benachbarte Spalten oder Zeilen nur in... | | | |
| (i) keiner Komponente (Variable) unterscheiden. | (ii) genau einer Komponente (Variable) unterscheiden.. | (iii) zwei Komponenten (Variablen) unterscheiden | (iv) in allen Komponenten (Variablen) unterscheiden. |
| | | | |
| e) Wie lautet das Komplementärgesetz zur Manipulation logischer Gleichungen? | | | |
| (i) $a + \overline{a} = 1$ | (ii) $(a + b) + c = a + (b + c)$ | (iii) $a + b = b + a$ | (iv) $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ |
| | | | |

# Zweierkomplement

$2^n$

Dezimal

$100 \ldots 0 = 10^n$

2-Komplement

$X = (X_{n-1} \ldots X_0)$

n-stellige Dualzahl

$K_2(X) = K_1(X) + 1$

$\Sigma$ Wort kombination von n Zeichen aus Alphabet $\Sigma$

$w \in \Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$ wobei $\Sigma^n = \underbrace{\Sigma \times \Sigma \times \cdots \Sigma}_{n\text{-mal}}$

$b=10$

$\Sigma_{10} = \{0,1,2,3,4,5,6,7,8,9\}$

wortlänge $n=8$

$b=2$ $\Sigma_2 = \{0,1\}$

$b=8$ $\Sigma_8 = \{0,1,2 \cdots 7\}$

$b=16$ $\Sigma_{16} = \{0,1 \cdots 9, A \cdots F\}$

Darstellung einer Zahl zur Basis b:

$b \in N$ mit $b>1$

natürlich Zahl $z$

$0 \le z \le b^n - 1$, $n \in N$

wort der Länge n über $\Sigma_b$ darstellbar

$z = \sum_{i=0}^{n-1} z_i b^i$

$z_i \in \Sigma_b$ für $i = 0, 1 \cdots n-1$

$z = (z_{n-1} z_{n-2} \cdots z_2 z_1 z_0)_b$

## Aufgabe Ü1: Addition von Dualzahlen

(9 Pkt.)

Beantworten Sie folgende Fragen im Bezug auf die 2er-Komplement-Darstellung ganzer Zahlen:

a. Geben Sie

(i) die kleinste darstellbare Zahl, -127 —— 10 000 001

(ii) die größte darstellbare Zahl, 128 —— 00 000 001

(iii) sowie die Null —— 00000000

unter Verwendung von 8 Bit an.

$2^8 = 256$

b. Es seien die Zahlen $(-30)_{10}$ sowie $(82)_{10}$ gegeben.

i) Geben Sie die Zweierkomplement-Darstellung der beiden Zahlen an. Verwenden Sie hierfür jeweils 8 Bits.

ii) Addieren Sie die beiden Zahlen binär. Der Rechenweg muss klar ersichtlich sein!

00011110
11100001
\+ 1
11100010

01010010

11100010
01010010
1 00110100

$(-30)_{10} + (82)_{10} = 00110100$

c. Folgende Dualzahlen in 2er-Komplement-Darstellung sind gegeben: 10100110 und 10010010.

(i) Addieren Sie die beiden Zahlen. **Achtung:** Der Rechenweg muss ersichtlich sein!

(ii) Hat bei der Addition ein Überlauf (Overflow) stattgefunden? Begründen Sie kurz Ihre Antwort.

10100110
10010010
1 00111000

d. Folgende Dualzahlen in 2er-Komplement-Darstellung sind gegeben: 10011100 und 01110110. Wird bei der Addition dieser Zahlen ein Überlauf stattfinden? Bitte begründen Sie Ihre Antwort **ohne** das Ergebnis konkret zu berechnen.

## **Aufgabe Ü2:** Festkommazahlen

(8.5 Pkt.)

Alle Teilaufgaben beziehen sich auf die 2er-Komplement-Darstellung.

a. Geben Sie die Dezimaldarstellung der folgenden binären Festkommazahlen an. Alle Zahlen sind in 2er-Komplement-Darstellung angegeben, das hochwertigste Bit (ganz links) dient als Vorzeichen-Bit.

(i) 001,100

(ii) 101,110

(iii) 110,101

# 7.4 Optimierung von Schaltnetzen

Funktionen

Kosten

zeitlich → PLA

räumlich → Platz

Anzahl der Gatter minimieren

Resolutionsregeln

*Diese Vereinfachung ist auch unter dem Namen* **Resolutionsregel** *bekannt, d.h. kommen in einer disjunktiven Form zwei Summanden vor, welche sich in* genau einer *komplementären Variablen unterscheiden, so können diese beiden Summanden durch den ihnen gemeinsamen Teil ersetzt werden.*

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

Kosten: 7 (只用3个连接)

Anwendung: Das Karnaugh-Diagramm

$n \leq 4$ $f: B^n \to B$, $n \in \{3, 4\}$

Bsp: $n = 4$ 4个变量 $2^4 = 16$

16 Bitmuster 可以写成 4x4 Matrix

| $x_3x_4$ \ $x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 10 | | | | |
| 11 | | | | |
| 01 | | | | |
| 00 | | | | |

zu Minterm

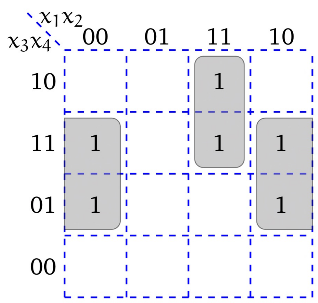



Abbildung 7.24: Beispiel für einen Zweier- und einen Viererblock.
Für den Viererblock gilt: $r = 1, s = 1$ d.h. $2^1 * 2^1 = 4 \longrightarrow$ Viererblock
Für den Zweierblock gilt: $r = 0, s = 1$ d.h. $2^0 * 2^1 = 2 \longrightarrow$ Zweierblock

meinern:
Rechteckige $2^r \times 2^s$–Blöcke, mit $r, s \in \{0, 1, 2\}$, von zyklisch benachbarten Einsen entsprechen $2^r * 2^s$ Mintermen, welche sich *paarweise* in $r + s$ Variablen unterscheiden, wobei alle Möglichkeiten des negierten bzw. nicht negierten Auftretens dieser Variablen vorkommen. Folglich läßt sich die Gesamtheit, d.h. die Summe dieser Minterme durch wiederholte Resolution zu dem Term vereinfachen, welcher gemeinsamer Bestandteil dieser Minterme ist. Er hat $(n - r - s)$ Variablen. D.h. je größer ein Block ist, desto weniger Variablen braucht man.

Für das Karnaugh–Diagramm bedeuted dies:

| |
|---|
| Man sollte alle im Diagramm auftretenden Einsen durch möglichst große Resolutionsblöcke der Form $2^r \times 2^s$ überdecken und dazu so viele Blöcke nutzen, dass jede 1 mindestens in einem Block vorkommt. |

|  | $x_1x_2$: 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| $x_3x_4$: 比如 10 |  |  |  | 1 |
| 11 | 1 |  |  | 1 |
| 01 | 1 |  |  | 1 |
| 00 |  |  |  | 1 |

绿色的部分看 10这列 (1和0是在每个组合都出现所以留下 $x_1\bar{x}_2$ 和这四行 因为找不~~找~~到一直出现的 (1 0)

看黄色部分是个 4-er Blöcke
竖着的一直有 1 那就留 $x_4$
横着的一直有 0 所以是 $\bar{x}_2$
化简后 $\bar{x}_2 x_4$

$$x_1\bar{x}_2 + \bar{x}_2 x_4$$

Don't Care – Argumente

## 7.4.3 Quine – McCluskey – Verfahren

DNF:

z.B. $\underbrace{x_1x_2\overline{x_3}x_4}_{\text{Minterm mit 4 Variablen}} + \underbrace{x_1x_2x_3x_4}_{\text{Minterm mit 4 Variablen}}$

DF $f: B^n \to B$ disjunktiver Form $\sum_{i=1}^{k} M_i,\ k \geq 1$

DF: $\underbrace{x_1x_2x_4}_{\text{Term mit 3 Variablen}}$

→ Jede DNF ist auch DF

→ Es gibt (etliche) DF, die keine DNF sind.

kann umgewandelt werden in eine DNF.

## Aufgabe Ü1: Zahlendarstellung

(12 Pkt.)

a. Geben Sie die 1er- und 2er-Komplementdarstellung der folgenden Zahlen unter der Annahme an, dass 8 Bit **inklusive** des Vorzeichenbits zur Verfügung stehen.

(i) 0

(ii) $-75$

(iii) 127

(i) 0000 0000 oder 1111 1111

0000 0000

(ii) 75 : 01001011

-75 : 10110100

10110101

127 : 0111 1111

0111 1111

b. Berechnen Sie in der 1er- und 2er-Komplementdarstellung folgende Differenzen unter der Annahme, dass 8 Bit **inklusive** des Vorzeichenbits zur Verfügung stehen. Achten Sie darauf, dass der Rechenweg ersichtlich ist.

(i) $52 - 46$

(ii) $64 - 32$

(iii) $-43 - 85$

46 ⇒ 00101110

(i) 52 ⇒ 00110100    -46 ⇒ 11010001

52 ⇒ 00110100    -46 ⇒ 11010010

(i) 1er-

```
  00110100        00110100
  11010001        11010010
 ---------       ---------
 100000101       100000110
```

(ii) 64 01000000    64-32

32 00100000

1er -32 ⇒ 11011111

2er -32 ⇒ 11100000

```
  01000000
  11011111
 ---------
 100011111

  01000000
  11100000
 ---------
 100100000
```

43 00101011 85 01010101

1er -43 11010100 -85 10101010

2er -43 11010101 -85 10101011

$$\begin{array}{r} 11010100 \\ 10101010 \\ \hline 101111110 \end{array} \qquad \begin{array}{r} 11010101 \\ 10101011 \\ \hline 1000000000 \end{array}$$

c. Nennen Sie zwei Vorteile, die sich bei der Zweierkomplementdarstellung von Binärzahlen gegenüber der Vorzeichen/Betrag-Darstellung (sign/magnitude) in Rechnern ergeben.

d. Gegeben seien die Zahlen $u = 100110$ und $v = 101111$ in Zweierkomplementdarstellung auf Basis von 6 Bit. Addieren Sie diese beiden Zahlen und achten Sie auf einen nachvollziehbaren Rechenweg. Hat bei der Addition ein Überlauf stattgefunden?

c. Man kann null ohne minus oder plus darstellen

## Aufgabe Ü2: Gleitkommazahlen

(9 Pkt.)

Geben Sie die Darstellung folgender Zahlen als Gleitkommazahl nach IEEE 754 in einfacher (32-Bit) und doppelter (64-Bit) Genauigkeit an. **Achtung:** Der Rechenweg muss ersichtlich sein!

a. $(14,5)_{10}$

b. $(0,1)_{10}$

c. $(\frac{-2}{3})_{10}$

Einfache Genauigkeit (32Bit) $1_{\text{Vorzeichen}}+8_{\text{Exponent}}+23_{\text{Significand}}$

$14{,}5_{10} = 1110{,}1_2 = (-1)^0 * 1{,}1101 * 2^3$ Vorzeichen S = 0
Exponent = $3 + 127_{\text{offset}} = 130 = 1000\ 0010_2$

Significand = 1101 0000 0000 0000 0000 000 (Nulls am Ende sind nur verwendet, um das volle Bit zu belegen)

Ergebnis: 0 10000010 1101 0000 0000 0000 0000 $000_2$

Einfache Genauigkeit (32Bit) $1_{\text{Vorzeichen}}+8_{\text{Exponent}}+23_{\text{Significand}}$

$14{,}5_{10} = 1110{,}1_2 = (-1)^0 * 1{,}1101 * 2^3$ Vorzeichen S = 0
Exponent = $3 + 127_{\text{offset}} = 130 = 1000\ 0010_2$

Significand = 1101 0000 0000 0000 0000 000 (Nulls am Ende sind nur verwendet, um das volle Bit zu belegen)

Ergebnis: 0 10000010 1101 0000 0000 0000 0000 $000_2$

Doppelte Genauigkeit (64Bit) $1_{\text{Vorzeichen}}+11_{\text{Exponent}}+52_{\text{Significand}}$

$14{,}5_{10} = 1110{,}1_2 = (-1)^0 * 1{,}1101 * 2^3$ Vorzeichen S = 0

Exponent = $3 + 1023_{\text{offset}} = 1026 = 100\ 0000\ 0010_2$

Significand = 1101 00000000 00000000 00000000 00000000 00000000 00000000
(Nulls am Ende sind nur verwendet, um das volle Bit zu belegen)

Ergebnis: 0 10000000010 1101 00000000 00000000 00000000 00000000 00000000 00000000

$14{,}5_{10} = 1110{,}1_2 = (-1)^0 * 1{,}1101 * 2^3$ Vorzeichen S = 0

Exponent = $3 + 1023_{\text{offset}} = 1026 = 100\ 0000\ 0010_2$

Significand = 1101 00000000 00000000 00000000 00000000 00000000 00000000
(Nulls am Ende sind nur verwendet, um das volle Bit zu belegen)

Ergebnis: 0 10000000010 1101 00000000 00000000 00000000 00000000 00000000 00000000

$0{,}1_{10} = (-1)^0 * 1{,}10011_2 * 2^{-4}$ (0011 wiederholt sich) Vorzeichen S = 0

Exponent = $-4 + 127_{offset} = 123 = 0111\ 1011_2$

Significand = 1001 1001 1001 1001 1001 101 (wir füllen mit möglichst viel „0011“, das letzte Bit wird gerundet)

Ergebnis: 0 01111011 1001 1001 1001 1001 1001 1001 $101_2$

Doppelte Genauigkeit (64Bit) $1_{Vorzeichen}+11_{Exponent}+52_{Significand}$

Analog zu 32Bit, $0{,}1_{10} = (-1)^0 * 1{,}10011_2 * 2^{-4}$ (0011 wiederholt sich)

Vorzeichen S = 0

Exponent = $-4 + 1023_{offset} = 1019 = 011\ 1111\ 1011_2$

Significand = 1001 10011001 10011001 10011001 10011001 10011001 10011010
(wir füllen mit möglichst viel „0011“, das letzte Bit wird gerundet)

Ergebnis: 0 01111111011 1001 10011001 10011001 10011001 10011001 10011001 10011010

Analog zu 32Bit, $0{,}1_{10} = (-1)^0 * 1{,}10011_2 * 2^{-4}$ (0011 wiederholt sich)

Vorzeichen S = 0

Exponent = $-4 + 1023_{offset} = 1019 = 011\ 1111\ 1011_2$

Significand = 1001 10011001 10011001 10011001 10011001 10011001 10011010
(wir füllen mit möglichst viel „0011“, das letzte Bit wird gerundet)

Ergebnis: 0 01111111011 1001 10011001 10011001 10011001 10011001 10011001 10011010

$-2/3_{10} = (-1)^1 * 1{,}01_2 * 2^{-1}$ (01 wiederholt sich)

Vorzeichen S = 1
Exponent = $-1 + 127_{offset} = 126 = 0111\ 1110_2$

Significand = 0101 0101 0101 0101 0101 011 (wir füllen mit möglichst viel „01“, das letzte Bit wird gerundet)

Ergebnis: $1\ 01111110\ 0101\ 0101\ 0101\ 0101\ 0101\ 011_2$

Analog zu 32Bit, $-2/3_{10} = (-1)^1 * 1{,}01_2 * 2^{-1}$ (01 wiederholt sich)

Vorzeichen S = 1

Exponent = $-1 + 1023_{offset} = 1022 = 011\ 1111\ 1110_2$

Significand = 0101 01010101 01010101 01010101 01010101 01010101 01010101
(wir füllen mit möglichst viel „01“, das letzte Bit wird gerundet)

Ergebnis: 1 01111111110 0101 01010101 01010101 01010101 01010101 01010101
01010101

Analog zu 32Bit, $-2/3_{10} = (-1)^1 * 1{,}01_2 * 2^{-1}$ (01 wiederholt sich)

Vorzeichen S = 1

Exponent = $-1 + 1023_{offset} = 1022 = 011\ 1111\ 1110_2$

Significand = 0101 01010101 01010101 01010101 01010101 01010101 01010101
(wir füllen mit möglichst viel „01“, das letzte Bit wird gerundet)

Ergebnis: 1 01111111110 0101 01010101 01010101 01010101 01010101 01010101
01010101

## Aufgabe Ü3: Quine-McCluskey

(9 Pkt.)

a. Vereinfachen Sie den folgenden Booleschen Term unter Anwendung des Algorithmus von Quine-McCluskey:
$f(x) = \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4 + \overline{x}_1 x_2\overline{x}_3\overline{x}_4 + x_1\overline{x}_2\overline{x}_3 x_4 + \overline{x}_1 x_2\overline{x}_3 x_4 + \overline{x}_1 x_2 x_3 x_4 + x_1\overline{x}_2 x_3 x_4 + x_1 x_2 x_3\overline{x}_4$
Geben Sie dabei alle notwendigen Schritte an!

b. Berechnen Sie die Kosten $K_1$ vor und $K_2$ nach der Optimierung. Wie viel kann an Kosten eingespart werden? Gehen Sie davon aus, dass die Gatter AND und OR jeweils Kosten von 1 verursachen, wobei NOT Gatter vernachlässigt werden können.

c. Begründen Sie, ob in diesem Beispiel auch eine Optimierung mittels Karnaugh-Diagrammen möglich wäre.

a) Finde die Minterme

Minterme = 0000, 0100, 1001, 0101, 0111, 1011, 1110

$= \sum m(0, 4, 9, 5, 7, 11, 14)$

Schritt 2:

Gruppiere die Minteme nach Anzahl von „0" (Anzahl von „1" auch möglich)

Gruppe 4:

0000 → 0

Gruppe 3:

0100 → 4

Gruppe 2:

0101 → 5

1001 → 9

Gruppe 1:

0111 → 7

1011 → 11

1110 → 14

Schritt 3: Vergleiche die Nachbargruppe, und dann gruppiere die Zahlen zusammen deren 3 von 4 Ziffer gleich sind.

Gruppe 3: 0,4

Gruppe 2: 4,5

Gruppe 1: 5,7

9,11

14

| | 0 | 4 | 5 | 7 | 9 | 11 | 14 |
|---|---|---|---|---|---|---|---|
| 1110 | | | | | | | X |
| 0-00 | X | X | | | | | |
| 010- | | X | X | | | | |
| 01-1 | | | X | X | | | |
| 10-1 | | | | | X | X | |

$$x_1x_2x_3\bar{x}_4 + \bar{x}_1\bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_4 + x_1\bar{x}_2x_4$$

b) $k_1 = 3 \times 7$ (and) $+ 6$ (or) $= 27$

$k_2 = 9$ (and) $+ 3$ (or) $= 12$

$k_1 - k_2 = 15$

c).

| | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | | |
| 01 | | 1 | | 1 |
| 11 | | 1 | | 1 |
| 10 | | | 1 | |

## **Aufgabe Ü4:** Einfachauswahlaufgabe: Boolesche Algebra

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n"). Nennen Sie dazu in Ihrer Abgabe explizit die jeweils ausgewählte Antwortnummer ((i), (ii), (iii) oder (iv)). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

a) Bei welcher Belegung $(x_1, x_2, x_3, x_4)$ ergibt die Boolesche Funktion $f(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee \overline{x_3}) \vee \overline{x_4}$ den Wert 1?

| (i) $(1,0,0,0)$ | (ii) $(0,1,1,0)$ | (iii) $(0,0,1,1)$ | (iv) $(0,1,0,1)$ |
|---|---|---|---|
| | | | |

b) Bei welcher Belegung $(a, b)$ ergibt der NOR–Operator ($\downarrow$) den Wert 1?

| (i) $(0,1)$ | (ii) $(1,1)$ | (iii) $(1,0)$ | (iv) $(0,0)$ |
|---|---|---|---|
| | | | |

c) Welche der folgenden Wertetabellen beschreibt die NOR-Funktion ($y = \overline{a+b}$)?

(i)

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

(ii)

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

(iii)

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

(iv)

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

d) Welche der folgenden Wertetabellen beschreibt die NAND-Funktion ($y = \overline{a \cdot b}$)?

(i)

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

(ii)

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

(iii)

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

(iv)

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

e) Ein Encoder besitzt die umgekehrte Funktionalität bezüglich eines Decoders. Angenommen ein Encoder hat $2^n$ Eingänge, von denen zu jedem Zeitpunkt genau einer mit einer 1 belegt ist. Wie viele Ausgänge muss der Encoder zur Umsetzung seiner Funktionalität besitzen?

| (i) $n$ | (ii) $\frac{2^n}{2}$ | (iii) $2 \cdot n$ | (iv) $2^n$ |
|---|---|---|---|
| | | | |

## Aufgabe 1: Einfachauswahlaufgaben

(12 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n"). Nennen Sie dazu in Ihrer Abgabe die jeweils ausgewählte Antwortnummer ((i), (ii), (iii) oder (iv)). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

| 1) Wie viele Bits werden zur byte-weisen Adressierung eines 2 Gigabyte großen Speichers benötigt? | | | |
|---|---|---|---|
| (i) 31 | (ii) 30 | (iii) 21 | (iv) 20 |
| [redacted] | [redacted] | [redacted] | [redacted] |

| 2) Welche Dualzahl entspricht dem hexadezimalen Wert B37? | | | |
|---|---|---|---|
| (i) 100001100001 | (ii) 101100111110 | (iii) 101100110111 | (iv) 101000110 |
| [redacted] | [redacted] | [redacted] | [redacted] |

| 3) Bei welcher Belegung $(a, b)$ ergibt der OR–Operator ($+$ oder $\vee$) den Wert 0? | | | |
|---|---|---|---|
| (i) $(1,1)$ | (ii) $(0,0)$ | (iii) $(0,1)$ | (iv) $(1,0)$ |
| [redacted] | [redacted] | [redacted] | [redacted] |

| 4) Bei welcher Belegung $(a, b)$ ergibt der NAND–Operator ($y = \overline{a \cdot b}$) den Wert 0? | | | |
|---|---|---|---|
| (i) $(0,1)$ | (ii) $(1,0)$ | (iii) $(1,1)$ | (iv) $(0,0)$ |
| [redacted] | [redacted] | [redacted] | [redacted] |

| 5) Welcher Speichertyp steht in der Speicherhierarchie am weitesten unten, da die Zugriffszeiten darauf am größten sind? | | | |
|---|---|---|---|
| (i) Register | (ii) Cache | (iii) Hauptspeicher | (iv) Hintergrundspeicher |
| [redacted] | [redacted] | [redacted] | [redacted] |

| 6) Welche Komponente ist gewöhnlich nicht an die South Bridge eines Mainboard-Chipsatzes angebunden? | | | |
|---|---|---|---|
| (i) USB-Schnittstellen | (ii) Audioausgang | (iii) Festplatten | (iv) Hauptspeicher |
| [redacted] | [redacted] | [redacted] | [redacted] |

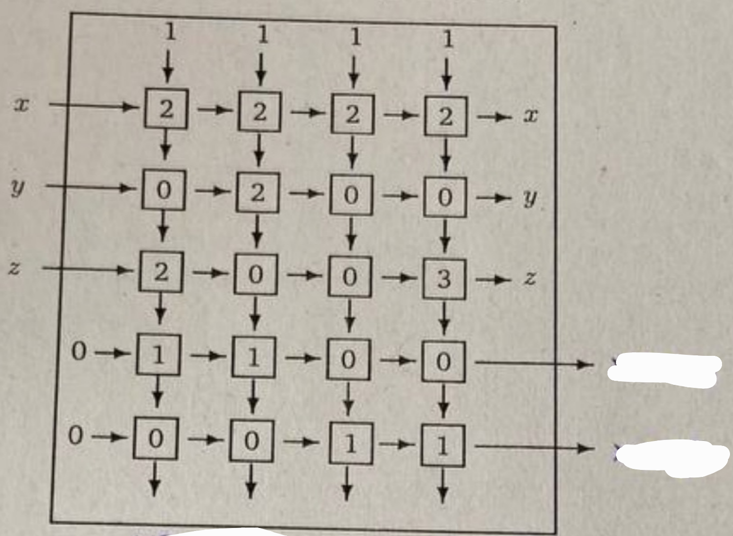
7) Sei folgende Wahrheitstafel einer Booleschen Funktion $f : B^2 \to B$ gegeben. Welcher Ausdruck entspricht der konjunktiven Normalform (KNF)?

| $i$ | $x_1$ | $x_2$ | $f(x_1, x_2)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 2 | 1 | 0 | 0 |
| 3 | 1 | 1 | 0 |

| | | | |
|---|---|---|---|
| (i) $f(x_1, x_2) = (\overline{x}_1 + x_2) \cdot (x_1 + \overline{x}_2) \cdot (\overline{x}_1 + \overline{x}_2)$ | (ii) $f(x_1, x_2) = \overline{x}_1 + x_2$ | (iii) $f(x_1, x_2) = (\overline{x}_1 + x_2) \cdot (\overline{x}_1 + \overline{x}_2)$ | (iv) $f(x_1, x_2) = \overline{x}_1\overline{x}_2 + \overline{x}_1 x_2$ |
| [redacted] | [redacted] | [redacted] | [redacted] |

| 8) Wie lautet die Zweierkomplementdarstellung von −81 unter Verwendung von 8 Bit? | | | |
|---|---|---|---|
| (i) 00010001 | (ii) 00110011 | (iii) 10101111 | (iv) 11001101 |
| [redacted] | [redacted] | [redacted] | [redacted] |

9) Welche Boolesche Funktion realisiert folgendes PLA?




| | | | |
|---|---|---|---|
| (i) $f(x, y, z) = (x\,\overline{z} + xyz + \overline{z}, \overline{x}yz + \overline{y}\,\overline{z})$ | (ii) $f(x, y, z) = (xz + xy, x + x\overline{z})$ | (iii) $f(x, y, z) = (x\,\overline{y}, \overline{x}yz + z)$ | (iv) $f(x, y, z) = (\overline{x}\,z + xy, x + x\overline{z})$ |
| [redacted] | [redacted] | [redacted] | [redacted] |

| | | | |
|---|---|---|---|
| [redacted] | [redacted] | [redacted] | [redacted] [redacted] |

10) Gegeben Sei folgende Zeile in SPIM Code: `var: .word 10, 11, 12, 13`
Welcher Befehl lädt den Wert 11 in das Register $t0?

| (i) | (ii) | (iii) | (iv) |
|---|---|---|---|
| `lw var, $t0+4` | `la $t0, var+4` | `lw $t0, var` | `lw $t0, var+4` |
| [redacted] | [redacted] | [redacted] | [redacted] [redacted] |

11) Wie lautet das dezimale Ergebnis der Addition der folgenden in Zweierkomplementdarstellung (unter Verwendung von 8 Bit) gegebenen Binärzahlen?

| | |
|---|---|
| | 11100010 |
| + | 01111100 |
| Übertrag | [redacted] |
| Ergebnis | [redacted] |

[redacted]

| (i) 01011111 | (ii) 01001110 | (iii) 101011110 | (iv) 01011110 |
|---|---|---|---|
| [redacted] | [redacted] | [redacted] | [redacted] [redacted] |

12) Wie bezeichnet man die Kombination mehrerer Quantenbits?

| (i) N-Qubit | (ii) Qubostat | (iii) Multi-Qubit | (iv) Quantenregister |
|---|---|---|---|
| [redacted] | [redacted] | [redacted] | [redacted] |

| a) Wie viele Bit enthält ein Byte? | | | |
|---|---|---|---|
| (i) 16 | (ii) 8 | (iii) 64 | (iv) 32 |
| | | | |
| b) Welche Binärzahl entspricht dem hexadezimalen Wert `C`? | | | |
| (i) `1110` | (ii) `0111` | (iii) `0110` | (iv) `1100` |
| | | | |
| c) Welche Komponente ist gewöhnlich nicht an die South Bridge eines Mainboard-Chipsatzes angebunden? | | | ① 输入输出 ② |
| (i) USB-Schnittstellen | (ii) Audio-Ausgang | (iii) Hauptspeicher | (iv) Festplatten |
| | | | |
| d) Wie lautet die höchste Speicheradresse bei einer Adressbreite von $n$ Bit? | | | |
| (i) $2+n-1$ | (ii) $2^n-1$ | (iii) $2/n-1$ | (iv) $2*n-1$ |
| | | | |
| e) Was ist keine Komponente der ~~Prozessorgrundstruktur~~? | | | |
| (i) Drucker | (ii) Operanden-register | (iii) Arithmetisch-logische Einheit | (iv) Befehlsregister |

| a) Bei welcher Belegung $(a,b)$ ergibt der OR–Operator ($+$ oder $\vee$) den Wert 0? | | | |
|---|---|---|---|
| (i) $(0,0)$ | (ii) $(0,1)$ | (iii) $(1,0)$ | (iv) $(1,1)$ |
| | | | |
| b) Bei welcher Belegung $(a,b)$ ergibt der AND–Operator ($\cdot$ oder $\wedge$) den Wert 1? | | | |
| (i) $(0,0)$ | (ii) $(0,1)$ | (iii) $(1,0)$ | (iv) $(1,1)$ |
| | | | |
| c) Eine Funktion $f: B^n \to B$ heißt n–stellige Boolesche Funktion ($B=\{0,1\}$). Wie viele n–stellige Boolesche Funktionen gibt es für jedes beliebige $n \in N$ mit $n \geq 1$? | | | |
| (i) $2^n$ | (ii) $2^{2^n}$ | (iii) $2 \cdot 2^n$ | (iv) $2^{2\cdot n}$ |
| | | | |
| d) Bei welcher Belegung $(x_1,x_2,x_3,x_4)$ ergibt die Boolesche Funktion $f(x_1,x_2,x_3,x_4)=(x_1 \cdot x_2 \cdot \overline{x_3})+(x_3 \cdot x_4)+\overline{x_2}$ den Wert 1? | | | |
| (i) $(1,1,1,0)$ | (ii) $(0,1,1,0)$ | (iii) $(0,1,0,1)$ | (iv) $(0,0,0,0)$ |
| | | | |
| e) Wie wird die Anzahl der benötigten Steuereingänge $s$ für einen $n$-Eingaben Multiplexer berechnet? | | | |
| (i) $s=n$ | (ii) $s=2*n$ | (iii) $s=log_n n$ | (iv) $s=log_2 n$ |

| a) Jedes MIPS-Register hat eine feste Breite. Sie beträgt: | | | |
|---|---|---|---|
| (i) 16 Bit | (ii) 32 Bit | (iii) 64 Bit | (iv) 8 Bit |
| | | | |
| b) Der MIPS Prozessor besitzt die folgende Architektur: | | | |
| (i) MISC | (ii) Stack | (iii) RISC | (iv) CISC |
| | | | |
| c) Welche Aussage bei SPIM ist falsch? Der Stack... | | | |
| (i) ...arbeitet nach dem FIFO Prinzip | (ii) ...arbeitet nach dem LIFO Prinzip | (iii) ...wächst in Richtung der Speicheradresse 0 | (iv) ...hat eine variable Größe |
| | | | |
| d) In der MIPS Architektur steht ein Wort für... | | | |
| (i) ...die größte adressierbare Informationseinheit. | (ii) ...die maximale Datengröße, die in einem Rechenschritt verarbeitet werden kann. | (iii) ...die kleinste adressierbare Informationseinheit. | (iv) ...die Größe einer Speicherzelle. |
| | | | |
| e) Gegeben sei folgende Zeile in SPIM Code: `var: .word 10, 11, 12, 13` Welcher Befehl lädt den Wert 13 in das Register $t0? | | | |
| (i) `lw var, $t0+4` | (ii) `la $t0, var` | (iii) `lw $t0, var+12` | (iv) `lw $t0, var+8` |
| | | | |

a) Sei folgende Wahrheitstafel einer Booleschen Funktion gegeben. Was ist die Menge der einschlägigen Indizes?

| $i$ | $x_1$ | $x_2$ | $f(x_1, x_2)$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 0 |

| | | | |
|---|---|---|---|
| (i) $\{0\}$ | (ii) $\{2\}$ | (iii) $\{0, 1\}$ | (iv) $\{0, 1, 3\}$ |

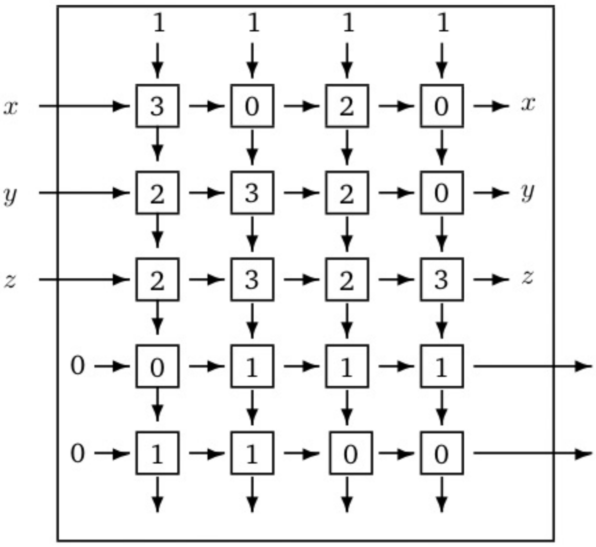
b) Welche der folgenden Mengen an Booleschen Funktionen ist nicht funktional vollständig?

| | | | |
|---|---|---|---|
| (i) $\{OR, NOT\}$ | (ii) $\{AND, OR\}$ | (iii) $\{NAND\}$ | (iv) $\{AND, NOT\}$ |

c) Jede Boolesche Funktion $F : B^n \to B$ ist eindeutig darstellbar als...

| | | | |
|---|---|---|---|
| (i) ... Produkt der Minterme ihrer einschlägigen Indizes. | (ii) ... Summe der Minterme ihrer einschlägigen Indizes. | (iii) ... Summe der Minterme ihrer nichteinschlägigen Indizes. | (iv) ... Summe der Maxterme ihrer einschlägigen Indizes. |

d) Welche Boolesche Funktion realisiert folgendes PLA?




| | | | |
|---|---|---|---|
| (i) $f(x, y, z) = (yz + xy + \overline{z}, \overline{x}z + \overline{y}\,\overline{z})$ | (ii) $f(x, y, z) = (\overline{y}\,\overline{z}, xyz + \overline{y})$ | (iii) $f(x, y, z) = (\overline{y}\,\overline{z}, \overline{x}yz + \overline{y}z)$ | (iv) $f(x, y, z) = (\overline{y}\,\overline{z} + xyz + \overline{z}, \overline{x}yz + \overline{y}\,\overline{z})$ |

e) Welcher der folgenden Booleschen Terme ist äquivalent zu $(x_1 \cdot x_2) + x_1 + x_3$?

| | | | |
|---|---|---|---|
| (i) $(\overline{x_1}x_2x_3) + (x_1\overline{x_2}x_3)$ | (ii) $(x_1x_2x_3) \cdot (x_1\overline{x_2}x_3)$ | (iii) $(x_1 + x_2 + x_3) + (x_1\overline{x_2}x_3)$ | (iv) $(x_1 + x_2 + x_3) \cdot (x_1 + \overline{x_2} + x_3)$ |

| a) Bei welcher Belegung $(x_1, x_2, x_3)$ ergibt die Boolesche Funktion $f(x_1, x_2, x_3) = (x_1 \cdot \overline{x}_2) + (x_2 \cdot \overline{x_3})$ den Wert 1? | | | |
|---|---|---|---|
| (i) $(0, 0, 1)$ | (ii) $(0, 1, 0)$ | (iii) $(1, 1, 1)$ | (iv) $(0, 1, 1)$ |
| | | | |

| b) Was bewirkt der Spim-Befehl `li $v0 5`: | | | |
|---|---|---|---|
| (i) Es wird ein Integer von der Konsole eingelesen | (ii) Es wird eine Zahl vom Typ double von der Konsole eingelesen | (iii) Der Wert 5 wird in das Register `$v0` geladen | (iv) Es wird ein Integer auf der Konsole ausgegeben |
| | | | |

| c) Wofür steht CISC im Zusammenhang mit Mikroprozessoren? | | | |
|---|---|---|---|
| (i) Complex Instruction Set Computer | (ii) Complex Instruction Set Calculator | (iii) Controversy Instruction Set Computer | (iv) Constructive Instruction Set Computer |
| | | | |

| d) Welche Aussage ist korrekt? MIPS ist eine... | | | |
|---|---|---|---|
| (i) Last-in-First-Out-Architektur | (ii) Stack-Architektur | (iii) Load-Store-Architektur | (iv) Heap-Architektur |
| | | | |

| e) Welche Aussage ist falsch? Die Funktion `syscall`... | | | |
|---|---|---|---|
| (i) beendet das Programm sofort | (ii) erwartet die Nummer der auszuführenden Funktion in `$v0` | (iii) besitzt selbst keine Parameter | (iv) führt eine Funktion des Betriebssystems aus |
| | | | |

| a) Angenommen ein Multiplexer hat `512` (Nutz-)Eingänge. Wie viele Steuereingänge werden benötigt, um die (Nutz-)Eingänge einzeln selektieren zu können? | | | |
|---|---|---|---|
| (i) `3` | (ii) `9` | (iii) `256` | (iv) `512` |
| | | | |

| b) Wie viele Felder enthält das Karnaugh-Diagramm einer Booleschen Funktion $f : B^3 \to B$? | | | |
|---|---|---|---|
| (i) 4 | (ii) 8 | (iii) 1 | (iv) 2 |
| | | | |

| c) Es kann sein, dass nicht alle $2^n$ Argumente einer Booleschen Funktion $f : B^n \to B$ $(n \geq 1)$ auftreten können. Wie bezeichnet man die Argumente einer solchen partiellen Funktion $f$, für die der Funktionswert nicht festgelegt ist? | | | |
|---|---|---|---|
| (i) Don't Worry | (ii) Don't Panic | (iii) Don't Know | (iv) Don't Cares |
| | | | |

| d) Die Reihenfolge der Beschriftung eines Karnaugh-Diagramms erfolgt so, dass sich zwei zyklisch benachbarte Spalten oder Zeilen nur in... | | | |
|---|---|---|---|
| (i) keiner Komponente (Variable) unterscheiden. | (ii) genau einer Komponente (Variable) unterscheiden.. | (iii) zwei Komponenten (Variablen) unterscheiden | (iv) in allen Komponenten (Variablen) unterscheiden. |
| | | | |

| e) Wie lautet das Komplementärgesetz zur Manipulation logischer Gleichungen? | | | |
|---|---|---|---|
| (i) $a + \overline{a} = 1$ | (ii) $(a + b) + c = a + (b + c)$ | (iii) $a + b = b + a$ | (iv) $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ |
| | | | |

| a) Welche der folgenden Dezimalzahlen hat zwei Darstellungen in der Einerkomplementdarstellung? | | | |
|---|---|---|---|
| (i) 2 | (ii) 0 | (iii) 1 | (iv) -1 |
| | | | |

| b) Welche der folgenden Antworten entspricht der Einerkomplementdarstellung der Dezimalzahl -74 (unter Verwendung von 8 Bit)? | | | |
|---|---|---|---|
| (i) 10110110 | (ii) 11000011 | (iii) 10111100 | (iv) 10110101 |
| | | | |

| c) Wie lautet die kleinste Dezimalzahl, die in der Zweierkomplementdarstellung darstellbar ist, wenn 6 Bit zur Darstellung zur Verfügung stehen? | | | |
|---|---|---|---|
| (i) -32 | (ii) -31 | (iii) -64 | (iv) -63 |
| | | | |

?

| d) Welche der folgenden Antworten entspricht der Zweierkomplementdarstellung der Dezimalzahl -97 (unter Verwendung von 8 Bit) | | | |
|---|---|---|---|
| (i) 10110110 | (ii) 10011110 | (iii) 11111111 | (iv) 10011111 |
| | | | |

e) Wie lautet das dezimale Ergebnis der Addition der folgenden in Zweierkomplementdarstellung gegebenen Binärzahlen?

| | |
|---|---|
| | 10110001 |
| + | 00110100 |
| Übertrag | 11100101 |
| Ergebnis | |

$2^0+2^1+2^3+2^4$

11100100

00011011

1+2+8+16

11+16

=27

| (i) -1 | (ii) 1 | (iii) 27 | (iv) -27 |
|---|---|---|---|
| | | | |

| a) Bei welcher Belegung $(x_1, x_2, x_3, x_4)$ ergibt die Boolesche Funktion $f(x_1, x_2, x_3, x_4) = \overline{(x_1 \vee x_2 \vee \overline{x_3})} \vee \overline{x_4}$ den Wert 1? | | | |
|---|---|---|---|
| (i) $(1, 0, 0, 0)$ | (ii) $(0, 1, 1, 0)$ | (iii) $(0, 0, 1, 1)$ | (iv) $(0, 1, 0, 1)$ |
| | | | |

| b) Bei welcher Belegung $(a, b)$ ergibt der NOR–Operator ($\downarrow$) den Wert 1? | | | |
|---|---|---|---|
| (i) $(0, 1)$ | (ii) $(1, 1)$ | (iii) $(1, 0)$ | (iv) $(0, 0)$ |
| | | | |

c) Welche der folgenden Wertetabellen beschreibt die NOR-Funktion ($y = \overline{a + b}$)?

(i)

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

(ii)

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

(iii)

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

(iv)

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

d) Welche der folgenden Wertetabellen beschreibt die NAND-Funktion ($y = \overline{a \cdot b}$)?

(i)

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

(ii)

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

(iii)

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

(iv)

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

4 2

2

2

?

| e) Ein Encoder besitzt die umgekehrte Funktionalität bezüglich eines Decoders. Angenommen ein Encoder hat $2^n$ Eingänge, von denen zu jedem Zeitpunkt genau einer mit einer 1 belegt ist. Wie viele Ausgänge muss der Encoder zur Umsetzung seiner Funktionalität besitzen? | | | |
|---|---|---|---|
| (i) $n$ | (ii) $\frac{2^n}{2}$ | (iii) $2 \cdot n$ | (iv) $2^n$ |
| | | | |

| | | | |
|---|---|---|---|
| a) Wie lautet das dezimale Ergebnis der Addition der folgenden in Zweierkomplementdarstellung gegebenen Binärzahlen? | | | |

| | |
|---|---|
| | 10110001 |
| + | 00110100 |
| Übertrag | 11100101 |
| Ergebnis | |

00011010

$\Rightarrow 2^1 + 2^3 + 2^4$

$2 + 8 + 16$

26

| | | | |
|---|---|---|---|
| ~~(i)~~ -27 | (ii) 67 | (iii) 32 | (iv) -93 |
| | | | |

| | | | |
|---|---|---|---|
| b) Um wie viele Stellen verschiebt sich das Komma der normalisierten Mantisse einer 32 Bit IEEE 754 Gleitkommazahl, wenn der Exponent 10110011 lautet? | | | |
| (i) 3 | (ii) 52 | (iii) 17 | (iv) 31 |
| | | | |

$2^0+2^1+2^4+2^5+2^7$

$1+2+16+32+128$

32 4 128 8

50+129

| | | | |
|---|---|---|---|
| c) Durch welche der folgenden Booleschen Funktionen wird ein Halbaddierer mit den Eingängen x und y und den Ausgängen R (Resultat) und Ü (Übertrag) realisiert? | | | |
| (i) $R = xy + \overline{x}\,\overline{y}$ $= \overline{x+y}$ | (ii) $R = \overline{(\overline{x}+y)} + \overline{(x+\overline{y})}$ $= \overline{(\overline{x}+\overline{y})}$ | (iii) $R = \overline{xy}$ $= \overline{xy}$ | (iv) $R = \overline{(\overline{x}y)} + \overline{(x\overline{y})}$ $= \overline{(\overline{x}\,\overline{y})}$ |
| | | | |

| | | | |
|---|---|---|---|
| d) Ein Volladdierer (Addition zweier Binärziffern und eines Übertrags) lässt sich mit... | | | |
| (i) ...zwei Halbaddierern realisieren. | (ii) ...zwei Halbaddierern und einem AND-Gatter realisieren. | (iii) ...zwei Halbaddierern und einem OR-Gatter realisieren. | (iv) ...zwei Halbaddierern und einem NOT-Gatter realisieren. |
| | | | |

| | | | |
|---|---|---|---|
| e) Wie bezeichnet man ein Addiernetz, bei welchem z.B. bei der Addition einer 8-stelligen Dualzahl die 4 höherwertigen Input-Operanden zweimal addiert werden und zwar für den Fall, dass bei der Addition der niederwertigen Hälfte der Input-Operanden ein Übertrag auftritt oder nicht, um damit die Berechnungszeit der Gesamtaddition zu verkürzen? | | | |
| (i) Carry-Save | (ii) Carry-Look-Ahead | (iii) Ripple-Carry | (iv) Carry-Select |
| | | | |

| | | | |
|---|---|---|---|
| a) Welche Belegung der beiden Eingänge S (Set) und R (Reset) eines SR-Latch ist unzulässig? | | | |
| (i) $S = 1, R = 1$ | (ii) $S = 1, R = 0$ | ~~(iii)~~ $S = 0, R = 0$ | (iv) $S = 0, R = 1$ |
| | | | |
| b) Womit können die beiden NOR-Gatter eines SR-Latch ersetzt werden, um ebenfalls ein äquivalentes Verhalten eines 1-Bit-Speichers zu realisieren? | | | |
| (i) AND-Gatter | (ii) OR-Gatter | (iii) NAND-Gatter | (iv) NOT-Gatter |
| | | | |
| c) Eine Flip-Flop-Schaltung, die das Eingangssignal übernimmt, wenn der Taktgeber von 0 auf 1 übergeht bezeichnet man als... | | | |
| (i) übersteuert. | (ii) pegelgesteuert. | (iii) nicht gesteuert. | (iv) flanken-gesteuert. |
| | | | |
| d) Angenommen aus Kostengründen würden nur NAND-Gatter produziert werden. Wie kann damit die OR-Funktion $(a + b)$ realisiert werden? | | | |
| (i) (a NAND b) NAND (a NAND b) | (ii) a NAND a | (iii) a NAND b | (iv) (a NAND a) NAND (b NAND b) |
| | | | |
| e) Welcher Speichertyp steht in der Speicherhierarchie oberhalb des Caches (d.h. der Zugriff drauf ist schneller als auf den Cache)? | | | |
| (i) Hintergrund-speicher | ~~(ii)~~ Register | (iii) Bandlaufwerk | (iv) Arbeitsspeicher |
| | | | |

| a) Wie bezeichnet man ein klassisches Problem der Mathematik/Informatik, das darin besteht, die Reihenfolge für den Besuch mehrerer Orte zu bestimmen, so dass die gewählte Route den kleinstmöglichen Weg hat? | | | |
|---|---|---|---|
| (i) Traveling-Salesman-Problem | (ii) Knapsack-Problem | (iii) Gate-Assignment-Problem | (iv) Boolean-Satisfiability-Problem |
| | | | |

| b) Wie bezeichnet man eine stochastischen Optimierungsmethoden, bei der man Sprünge verschiedener Größe in der Lösungslandschaft durchführt, um eine Lösung mit möglichst geringen Kosten zu finden? | | | |
|---|---|---|---|
| (i) Simulated Annealing | (ii) Newton-Verfahren | (iii) Division | (iv) Scheduling |
| | | | |

| c) Wie bezeichnet man die beobachtete Grundregel in der Entwicklung neuer Computerchips nach der sich die Transistordichte auf Computerchips in etwa alle 12-18 Monate verdoppelt? | | | |
|---|---|---|---|
| (i) Zuse's Law | (ii) Moore's Law | (iii) Gordon's Law | (iv) Heisenberg's Law |
| | | | |

| d) Wie bezeichnet man die Überlagerung zweier Zustände in der Quantenwelt? | | | |
|---|---|---|---|
| (i) Verdeckung | (ii) Gewichtung | (iii) Superposition | (iv) Linearität |
| | | | |

| e) Was ist der übliche Zeitwert für einen Annealing-Vorgang auf dem D-Wave Quantum Annealer? | | | |
|---|---|---|---|
| (i) 20 Minuten | (ii) 20 Sekunden | (iii) 20 Millisekunden | (iv) 20 Mikrosekunden |

| a) Wie viele Bit stehen im ursprünglichen ASCII-Code zur Kodierung eines Zeichens zur Verfügung? | | | |
|---|---|---|---|
| (i) 16 | (ii) 1 | ~~(iii)~~ 128 | (iv) 7 |
| | | | |

b) Die Dezimalzahl 16.909.060 (01020304 Hexadezimal) soll als 32-Bit-Integer-Wert (Wortbreite) ab Speicheradresse 0000 gespeichert werden. Dabei kommt die Little Endian Byte-Anordnung zum Einsatz. Welche Antwort entspricht der resultierenden Speicherbelegung?

(i)

| Adresse | Wert |
|---|---|
| 0000 | 04 |
| 0001 | 03 |
| 0002 | 02 |
| 0003 | 01 |

(ii)

| Adresse | Wert |
|---|---|
| 0000 | 01 |
| 0001 | 02 |
| 0002 | 03 |
| 0003 | 04 |

(iii)

| Adresse | Wert |
|---|---|
| 0000 | 03 |
| 0001 | 01 |
| 0002 | 02 |
| 0003 | 04 |

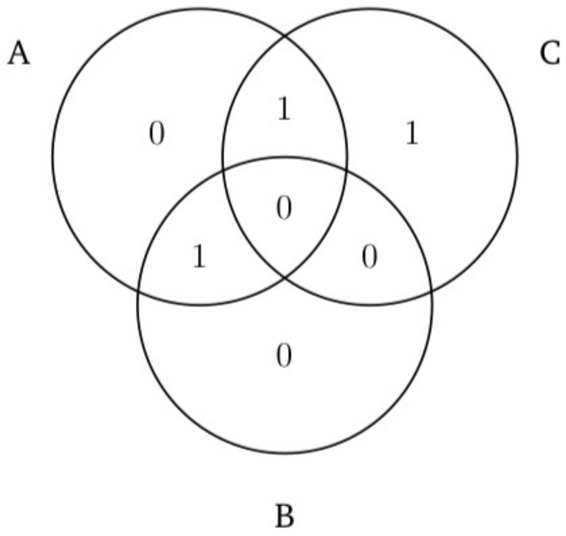
(iv)

| Adresse | Wert |
|---|---|
| 0000 | 02 |
| 0001 | 01 |
| 0002 | 04 |
| 0003 | 03 |

| c) Welche Operation kann auf zwei gleichlange Codewörter angewendet werden, um durch Zählen der 1en im Ergebnis den Hamming-Abstand der Codewörter zu bestimmen? | | | |
|---|---|---|---|
| (i) XOR | (ii) AND | (iii) NOR | (iv) OR |
| | | | |

1234567

d) Gehen Sie nun davon aus, dass Sie folgendes Code-Wort **1010001** empfangen haben. Es wurde **gerade Parität** verwendet. Bei der Übertragung ist ein einzelner Bitfehler aufgetreten. Welches Paritätsbit ist betroffen?




| | | | |
|---|---|---|---|
| (i) keins | (ii) $A$ | (iii) $B$ | (iv) $C$ |
| | | | |

e) Angenommen ein Speicherwort wird in einem kurzen Zeitintervall $k$ mal gelesen oder geschrieben und befindet sich nach dem ersten Zugriff im Cache. Wie berechnet sich die Trefferrate (Hit Ratio) $h$?

| | | | |
|---|---|---|---|
| (i) $h = (k-1) \cdot (k)$ | (ii) $h = \frac{k-1}{k}$ | (iii) $h = \frac{k}{k}$ | (iv) $h = \frac{k}{k-1}$ |
| | | | |

1) Welche Dualzahl entspricht dem hexadezimalen Wert `C9`?

| | | | |
|---|---|---|---|
| (i) `10000001` | (ii) `11001001` | (iii) `10111111` | (iv) `10101010` |
| | | | |

2) Wie lautet eine der De Morganschen Regeln?

| | | | |
|---|---|---|---|
| (i) $\overline{(a+b)} = \overline{a} \cdot \overline{b}$ | (ii) $a \cdot 0 = 0$ | (iii) $a + \overline{a} = 1$ | (iv) $a \cdot (b+c) = (a \cdot b) + (a \cdot c)$ |
| | | | |

3) Wie lautet die Belegung von `$t2` nach Ausführung des folgenden SPIM-Codes?

```
.data
var: .word 8, 32, 17, 4, 9

.text
main: lw $t1, var
      lw $t2, var+4($t1)
```

| | | | |
|---|---|---|---|
| (i) `8` | (ii) `32` | (iii) `12` | (iv) `4` |
| | | | |

4) Sei folgende Wahrheitstafel einer Booleschen Funktion $f : B^2 \to B$ gegeben. Welcher Ausdruck entspricht **nicht** dieser Funktion?

| $i$ | $x_1$ | $x_2$ | $f(x_1, x_2)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 1 |

| (i) $f(x_1, x_2) = \overline{\overline{(x_1 \cdot x_1)} \cdot x_2}$ | (ii) $f(x_1, x_2) = x_1 + \overline{x}_2$ | (iii) $f(x_1, x_2) = \overline{(\overline{x}_1 \cdot x_2)}$ | (iv) $f(x_1, x_2) = (x_1 + \overline{x}_2) \cdot (\overline{x}_1 + \overline{x}_2)$ |
|---|---|---|---|
| | | | |

5) Ein Carry-Save-Addiernetz...

| | | | |
|---|---|---|---|
| (i) ...berechnet Teilergebnisse doppelt und selektiert eines davon. | (ii) ...dient der schnellen Addition von mehr als zwei Summanden. | (iii) ...lässt den endgültigen Übertrag (von rechts nach links) durch das Schaltnetz rieseln. | (iv) ...führt die Additionsoperation auf die Subtraktion zurück. |
| | | | |

6) Was ist kein Schritt im Pipelining einer klassischen RISC CPU?

| | | | |
|---|---|---|---|
| (i) Instruction Fetch | (ii) Instruction Decode | (iii) Erase | (iv) Execute (ALU) |

7) Um welche Art von Hazards handelt es sich, wenn Entscheidungen (z.B. Sprünge) erst basierend auf einem späteren Ausführungsschritt oder dem noch ausstehenden Ergebnis einer vorangegangenen Instruktion getroffen werden können und dadurch das Pipelining erschwert wird?

| | | | |
|---|---|---|---|
| (i) Control Hazards | (ii) Biohazards | (iii) Data Hazards | (iv) Structural Hazards |
| | | | |

8) Angenommen ein Decoder hat `4` Eingänge. Wie viele Ausgänge muss der Decoder zur vollständigen Umsetzung seiner Funktionalität besitzen?

| | | | |
|---|---|---|---|
| (i) `2` | (ii) `4` | (iii) `8` | (iv) `16` |
| | | | |

9) Wie lautet die Belegung von `$t2` nach Ausführung des folgenden SPIM-Codes?

```
.data
var: .word 12, 31, 9, 5, 4

.text
main: lw $t1, var
      lw $t2, var($t1)
```

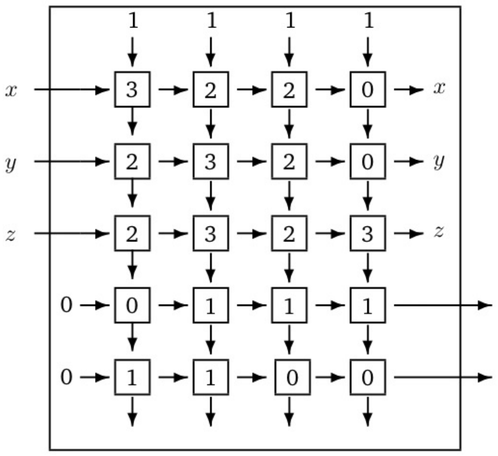
| | | | |
|---|---|---|---|
| (i) `12` | (ii) `9` | (iii) `5` | (iv) `4` |
| | | | |

10) Wie lautet das dezimale Ergebnis der Addition der folgenden in Zweierkomplementdarstellung (unter Verwendung von 8 Bit) gegebenen Binärzahlen?

| | |
|---|---|
| | `00001111` |
| + | `00110001` |
| Übertrag | |
| Ergebnis | |

| | | | |
|---|---|---|---|
| (i) `110` | (ii) `64` | (iii) `32` | (iv) `-19` |
| | | | |

11) Welche Boolesche Funktion realisiert folgendes PLA?




| | | | |
|---|---|---|---|
| (i) $f(x, y, z) = (\overline{yz} + xyz + \overline{z}, \overline{x}yz + \overline{yz})$ | (ii) $f(x, y, z) = (\overline{yz}, xyz + \overline{y})$ | (iii) $f(x, y, z) = (\overline{yz}, \overline{x}yz + \overline{y}z)$ | (iv) $f(x, y, z) = (x\overline{yz} + xyz + \overline{z}, \overline{x}yz + x\overline{y}\,\overline{z})$ |
| | | | |

12) Welche Belegung der beiden Eingänge S (Set) und R (Reset) eines SR-Latch ist unzulässig?

| | | | |
|---|---|---|---|
| (i) $S = 1, R = 1$ | (ii) $S = 1, R = 0$ | (iii) $S = 0, R = 1$ | (iv) $S = 0, R = 0$ |
| | | | |