

Zen mit ein oder zwei Lächeln –  
Informationsverarbeitung in IT-Systemen (IIS)  
vormals : Digitaltechnik für BG (DT)  
Danke Günter und Katrin

<http://www.freewebs.com/gtslehrer/> - [mailto: gtslehrer @ yahoo.com](mailto:gtslehrer@yahoo.com)

9. November 2007

## Inhaltsverzeichnis

<b>1 Analoge vs. Digitale Signale</b>	<b>2</b>
1.1 Bewertung . . . . .	2
1.2 Was ist analog bzw. digital? . . . . .	2
1.3 Vorteile und Nachteile . . . . .	4
<b>2 Dualzahlen</b>	<b>5</b>
2.1 Aufbau und Wandlung nach Dezimal . . . . .	5
2.2 Wandeln Dezimal-Binär . . . . .	5
2.2.1 Übung . . . . .	6
2.3 Addieren . . . . .	6
2.3.1 Übung . . . . .	6
2.4 Subtrahieren . . . . .	7
2.4.1 Übung . . . . .	7

<i>INHALTSVERZEICHNIS</i>	2
2.5 Multiplizieren . . . . .	7
2.5.1 Übung . . . . .	8
2.6 Dividieren . . . . .	8
2.6.1 Übung . . . . .	9
<b>3 Binäre Brüche [optional]</b>	<b>10</b>
3.1 vorwärts . . . . .	10
3.2 rückwärts . . . . .	10
3.3 0,1, 0,00001 etc. . . . .	10
<b>4 Hexadezimal</b>	<b>11</b>
4.1 Übersicht . . . . .	11
4.2 Ziffern . . . . .	11
4.3 dez - hex . . . . .	11
4.4 hex - dez . . . . .	12
4.4.1 Übung . . . . .	12
4.5 dual - hex und zurück, aber fix . . . . .	12
<b>5 Oktal</b>	<b>13</b>
5.1 Ziffern . . . . .	13
5.2 dez - oct . . . . .	13
5.3 oct - dez . . . . .	13
5.3.1 Übung . . . . .	13
<b>6 Negative Zahlen (nur bin)</b>	<b>14</b>
6.0.2 Übung . . . . .	14
6.1 Überschreitung des Zahlenbereiches . . . . .	14
6.1.1 Übung Zahlenbereich . . . . .	15
6.1.2 Übung Günter . . . . .	15

<b>7 Logische Bausteine</b>	<b>16</b>
7.1 TTL-Level . . . . .	16
7.2 Schaltalgebra (Boole'sche Algebra) in Wahrheitstabellen . . . . .	16
7.3 Gesetze . . . . .	17
7.4 Konvertierung zu NICHT-UND . . . . .	17
7.5 vollständige Normalform . . . . .	18
7.6 Vereinfachung nach Boolescher Algebra . . . . .	18
7.7 Übung: XOR mit 3 Eingängen aus NAND-Bausteinen . . . . .	19
7.8 Maschinenlast . . . . .	19
7.9 Übung: 7-Segment-Anzeige . . . . .	19
7.10 Elektronik-Bastelkasten . . . . .	21
<b>8 Addierer</b>	<b>22</b>
<b>9 Gray-Code</b>	<b>23</b>
<b>10 Karnaugh-Veitch-Diagramme (KV-Diagramme) (4 Bit only)</b>	<b>24</b>
10.1 Wie's geht . . . . .	24
10.2 7-Segment vereinfachen. . . . .	26
10.3 Wandler 4-Bit-Gray nach bin vereinfachen. . . . .	26
10.4 Wandler 4-Bit-Gray nach BCD vereinfachen. . . . .	26
10.5 Pseudo-Tetraden-Anzeiger vereinfachen. . . . .	26
10.6 Halbaddierer und Volladdierer vereinfachen. . . . .	26
10.7 Maschine, die durch 4 teilbare Zahlen anzeigt, vereinfachen. . . . .	26
10.8 Hardware-Multiplizierer für 2 Zahlen von je 2-Bit bauen. . . . .	26
<b>11 Speicherzellen</b>	<b>27</b>
11.1 1-Bit-Flipflop (FF) . . . . .	27
11.2 Taktflankengesteuertes FF . . . . .	27

11.3	Master-Slave-Set-Reset-FF (MS-RS-FF)	29
11.3.1	Prellende Schalter	29
11.4	MSRSFF	29
11.4.1	Duke oder Count?	29
11.5	Rückwärtszähler	30
11.5.1	Move oder Shift?	30
<b>12</b>	<b>Zen oder die Kunst, Digitaltechnik anzuwenden</b>	<b>31</b>
12.1	Größenwahn: 2-stelliger Dezimalzähler	31
12.2	Umschaltbarer Zähler	31
12.3	Noch größer: Kursarbeit 24-Stunden-Uhr HHMMSS	31
12.4	Ampel an Einmündung	31
12.5	Schrittmotor	31
<b>13</b>	<b>Klausur-Ideen</b>	<b>33</b>
13.1	Umrechnen	33
13.2	Rechnen	33
13.3	Wertebereich	33
13.4	Exclusive-Oder	33
13.5	4-Bit-Hardware-Multizierer	33
13.6	4-Bit-Gucker: A gleich ungleich größer kleiner B	33

### **Disclaimer**

Wissen ist zum Teilen da. Ich teile mein Wissen mit Ihnen, lieber Kollege.

Ich bin aber nicht perfekt. Unter [gtslehrer@yahoo.com](mailto:gtslehrer@yahoo.com)  
nehme ich dankbar Ihre Verbesserungsvorschläge entgegen.

\*

**Legal Blurb:** Alle Informationen in diesem Dokument sind falsch, unvollständig, irreführend, irrelevant und / oder  
funktionieren einfach nicht.

Wenn Sie es trotzdem benutzen, und es geht dabei etwas kaputt, ist das Ihr Problem, nicht meins.

\*

**Bitte teilen Sie meine Web-Adresse nicht Ihren Schülern mit.**

## 1 Analoge vs. Digitale Signale

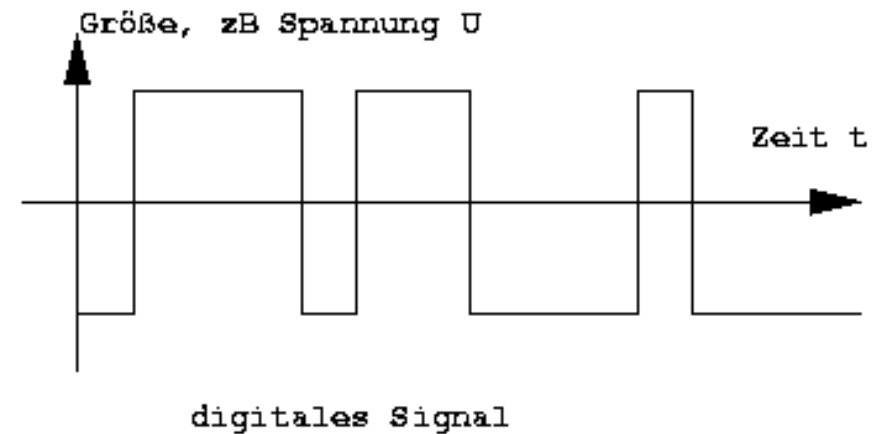
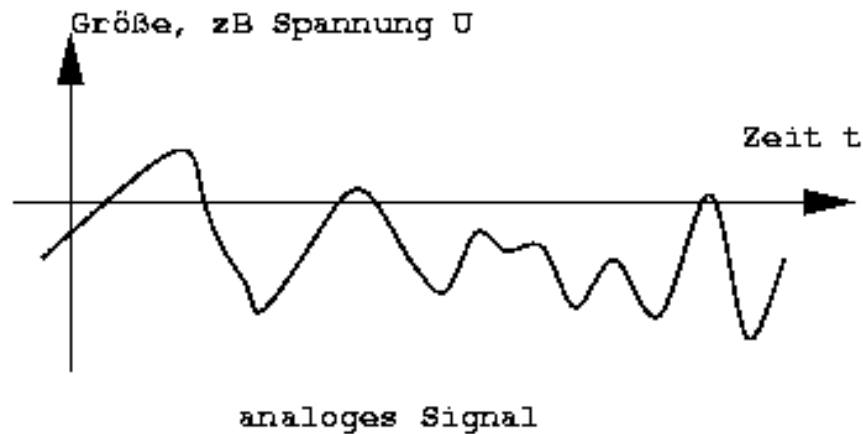
Informationsverarbeitung in IT-Systemen findet digital statt.

### 1.1 Bewertung

- 2 Klausuren
- 2 mündliche Noten nach `leitfaden.pdf` incl. Sozialverhalten und Teamarbeit

### 1.2 Was ist analog bzw. digital?

Analoge Signale haben unendlich viele Zwischenzustände.  
Digitale Signale haben einen begrenzten Bereich von Zuständen, üblicherweise 2.  
Mit mehr als 1 Bit, zB  $n$  Bit, gibt es  $2^n$  mögliche Zustände. Die Bits werden dabei nacheinander (sequentiell) übertragen.



Beispiel Stereoton<sup>1</sup>:

- Spannungsunterschiede = Tonsignal in 2 Cinch-Kabeln vom Cassettenrecorder zum Verstärker
- Spannung AN oder AUS in Kupferkabel. Licht AN oder AUS im Lichtwellenleiter, Magnet AN oder AUS auf Festplatte.

Beispiel Geschwindigkeitsmessung im Auto

- biegsame Welle vom Getriebeausgang zum Tacho
- hin- und herfließende Bits zwischen Standard-Sensor, sowieso vorhandenem Netzwirkabel (zB CAN-Bus) und elektronischer Anzeige

<sup>1</sup>CD-Signal üblicherweise 44100 Hz, 2 Kanäle, 16 Bit pro Kanal : 176 kByte/s

### 1.3 Vorteile und Nachteile

#### analog

- verliert beim Kopieren und Übertragen an Qualität
- + anschaulich und reparabel
- manipulierbar
- Genauigkeit begrenzt

#### digital

- + beim Kopieren und Übertragen : Qualität = *const.*
- + Fehler erkennbar und korrigierbar
- manipulierbar
- komplex, binär, softwareabhängig
- + mit geringstem Aufwand kunden-anpaßbar
- braucht A/D- und D/A-Wandler,
- + die aber in Massen zuverlässig und billig gefertigt werden.
- Genauigkeit begrenzt

## 2 Dualzahlen

Schreibweise:

- o Index D für Dezimal
- o Index B für Binär

### 2.1 Aufbau und Wandlung nach Dezimal

Wiederholung:

$$123D = 3 * 10^0 + 2 * 10^1 + 1 * 10^2 = 3 * 1 + 2 * 10 + 1 * 100 = 123$$

Binärzahl:

$$100101010B =$$

$$= 0*2^0 + 1*2^1 + 0*2^2 + 1*2^3 + 0*2^4 + 1*2^5 + 0*2^6 + 0*2^7 + 1*2^8 =$$

$$= 0 + 2 + 0 + 8 + 0 + 32 + 0 + 0 + 256 = 298D$$

### 2.2 Wandeln Dezimal-Binär

Wandle Zahl Z durch ganzzahliges Teilen durch 2

Rest = Binärstelle, Quotient = neue Zahl

bis Quotient = 0

Beispiel: 56D = ?B

$$56 : 2 = 28 \text{ R } 0 // \text{ dies ist die Einer-Stelle}$$

$$28 : 2 = 14 \text{ R } 0$$

$$14 : 2 = 7 \text{ R } 0$$

$$7 : 2 = 3 \text{ R } 1$$

$$3 : 2 = 1 \text{ R } 1$$

$$1 : 2 = 0 \text{ R } 1 // \text{ dies ist die höchste Stelle}$$

$$\text{Ergo : } 56D = 111000B \text{ Probe: } 111000B = 8 + 16 + 32 = 56D \text{ o.k.}$$

Hinweise:

- o Durch 10 teilbare Dezimalzahlen haben eine 0 hinten.
  - o Durch 2 teilbare Binärzahlen haben eine 0 hinten.
  - o Durch 100 teilbare Dezimalzahlen haben 00 hinten.
  - o Durch 4 teilbare Binärzahlen haben 00 hinten.
  - o Durch 1000 teilbare Dezimalzahlen haben 000 hinten.
  - o Durch 8 teilbare Binärzahlen haben eine 000 hinten.
- Probe : 56D ist durch 8D teilbar und hat als 111000B eine 000 hinten.

### 2.2.1 Übung

Wandeln Sie nach Binär und zurück nach Dezimal: 5 6 34 72 181 255 256.

### 2.3 Addieren

Wiederholung:  $5 + 6 = (1, 1 \text{ gemerkt}) = 11$ .

```

  101B
+ 110B
-----
  1      gemerkt
=====
 1011B      Probe: 1 + 2 + 8 = 11 o.k.

```

Hinweis: 0+0 gibt 0. 1+0 gibt 1, 0 gemerkt. 1+1 gibt 0, 1 gemerkt.  
 Auch der Übertrag muß im Dualsystem erfolgen. Ist doch logisch.

### 2.3.1 Übung

Addieren Sie: 5+6, 20+32, 100+255, 1023+756 dezimal und binär, mit Probe.

**2.4 Subtrahieren****Obere Zahl muß größer sein als untere Zahl!!!**

$123 - 67 = ?$

7 bis 13 (eins gemerkt) = 6

$6+1 \text{ bis } 12 = 5$

Ergebnis:  $123 - 67 = 56$

$10010B - 1001B = ?$

10010

- 1001

-----

1

=====

1001 ----> nanu,  $1001B * 10B = 10010B$ . Natürlich.

Probe:  $18D - 9D = 9D$  o.k.

**2.4.1 Übung**

Subtrahieren Sie die Summen aus der vorigen Übung, mit Probe.

**2.5 Multiplizieren**Wiederholung:  $123*45 = 123*40 + 123*5$ 

123 \* 45

-----

4920 &lt;--- diese 0 wird nicht geschrieben

615

-----

5535

```

1001001 * 10010
-----
  1001001      <--- drei Nullen nicht geschrieben
   1001001     <--- multiplizieren mit 0 oder 1 ist easy
-----
 10100100010

```

Probe:  $73 * 18 = 1314$  o.k.

### 2.5.1 Übung

$5*6$ ,  $13*14$ ,  $18*19$ , alles mit Probe.

### 2.6 Dividieren

Wiederholung:

$1326 : 3 = ?$

1 durch 3 geht nicht

13 (eigentlich: 1300) durch 3 ist 4 (eigentlich: 400) bleiben 1

12 (eigentlich: 120) durch 3 sind 4 (eigentlich: 40) bleibt 0

3 durch 3 ist 1.

$1326 : 3 = 441$ .

$10100100010 : 10010 = 1001001$

10010

-----

```

  101
 1010      <--- paßt nicht - 0 runterziehen
 10100     <--- paßt nicht - 0 runterziehen
- 10010
-----

```

```
100      <--- paßt nicht - 0 runterziehen
1001     <--- paßt nicht - 1 runterziehen
10010    <--- paßt nicht - 0 runterziehen
-----
      0
```

Probe:  $1314 : 18 = 73$  was zu erwarten war.

### 2.6.1 Übung

Dividieren Sie alles aus der Multiplikations-Übung, mit Probe.

### 3 Binäre Brüche [optional]

#### 3.1 vorwärts

0,25D = ???B

Geht im Prinzip genauso, bloß mit 2 multiplizieren:

"0," hinschreiben,  $0,25 * 2 = 0,5$

"0" hinschreiben,  $0,5 * 2 = 1$ .

"1" hinschreiben, fertig.

#### 3.2 rückwärts

Die Nachkommastellen setzen die Reihe der Vorkommastellen lückenlos fort:

$$2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, \dots = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \dots$$

#### 3.3 0,1, 0,00001 etc.

1 nach binär überführen.

2 Überlegen, ob  $\infty$  Stellen in eine 256-bit-zahl passen.

3 lernen:

**Normale Rechner rechnen ungenau mit Dezimalzahlen.**

## 4 Hexadezimal

### 4.1 Übersicht

Was denn	Basis 10	Basis 2	Basis 8	Basis 16
Index vorangestellt	D	B	O	H
Ziffern die ersten 17 Zahlen	nichts 10 0, 1, 2, 3, 4, 5, 6, 7, ...	nichts, Index benutzen 2	0 oder Index 8	0x oder 0X oder Index 16

### 4.2 Ziffern

Hex Zahlen werden gekennzeichnet:

- durch ein nachgestelltes H oder
- durch ein vorgestelltes 0x oder 0X.

Für Hex braucht man 16 Ziffern, die die Werte 0D bis 15D haben. Außer 0 bis 9 nimmt man A bis F.

Die Stellen haben, von rechts gelesen, die Wertigkeiten  $16^0, 16^1, 16^2, 16^3, \dots = 1, 16, 256, 4096, \dots$

### 4.3 dez - hex

Geht im Prinzip wie bei Dualsystem, nur daß man den Rest in hex-Ziffern schreibt und durch 16 teilt.

#### 4.4 hex - dez

0.Stelle (das ist die 1. von rechts) \* 1 +

1. Stelle (das ist die 2. von rechts) \* 16 +

2. Stelle (das ist die erste von rechts) \* 16 \* 16 +

usw. ...

Wir rechnen nicht mit hex.

##### 4.4.1 Übung

Wandeln Sie nach Hex und zurück nach Dezimal: 5 6 34 72 181 255 256.

#### 4.5 dual - hex und zurück, aber fix

Schreiben Sie die obigen Zahlen in bin und in hex nebeneinander. Erkennen und formulieren Sie die einfache, elegante Umwandlungsregel.

## 5 Oktal

### 5.1 Ziffern

Oktale Zahlen werden gekennzeichnet durch ein vorangestelltes O oder 0.

Für oct braucht man 8 Ziffern, die die Werte 0D bis 7D haben.

Die Stellen haben, von rechts gelesen, die Wertigkeiten  $8^0, 8^1, 8^2, 8^3, \dots = 1, 8, 64, 512, \dots$

### 5.2 dez - oct

Geht im Prinzip wie bei Dualsystem, nur daß man den Rest in hex-Ziffern schreibt und durch 16 teilt.

### 5.3 oct - dez

0.Stelle (das ist die 1. von rechts) \* 1 +

1. Stelle (das ist die 2. von rechts) \* 8 +

2. Stelle (das ist die erste von rechts) \* 8 \* 8 +

usw. ...

Wir rechnen nicht mit oct.

Es gibt keinen schnellen Weg hex-oct-hex, aber einen schnellen bin-oct-bin. Finden Sie ihn.

#### 5.3.1 Übung

Wandeln Sie nach Oct und zurück nach Dezimal: 5 6 34 72 181 255 256.

## 6 Negative Zahlen (nur bin)

Beispiel: ein fünfstelliger mechanischer Tacho rollt von 99999 nach 100000, zeigt aber nur 00000. Die erste 1 schmeißt er weg.  
Bei Binärzahlen nehmen wir das erste Bit als Vorzeichen. Wir verwenden nur 4- und 8stellige bin-Zahlen.

00000000 = 0D

00000001 = 1D

Zu welcher Zahl muß ich 1 dazuzählen, damit 0 herauskommt? ...

... zu -1D, 11111111B. 11111111B ist also -1.

11111111 = -1D

11111110 = -2D

**Man kommt zur negativen Binärzahl, indem man alle Bits umkippt und 1 addiert.**

Beispiel:  $-0 = 11111111B + 1B = 00000000$  was zu erwarten war.

Beispiel:  $-5 = -(0101B) + 1B = 1010B + 1B = 1011B = 11D$ .  $11D + 5D = 16D$ , und das sieht genauso aus wie 0D.

**Übung:** Erzeugen Sie -8 und -28.

**Positive Zahl + Negative Zahl klappt immer.**

**Wenn bei Positiv+Positiv etwas Negatives herauskommt, oder bei Negativ+Negativ etwas Positives, dann muß man das CF dazu-nehmen.**

### 6.0.2 Übung

**Übung:** Berechnen Sie in 8 Bit:

$+5+6$   $+5+(-6)$   $(-5)+6$   $(-5)+(-6)$  und genauso  $\pm 60 \pm 90$ .

### 6.1 Überschreitung des Zahlenbereiches

**Übung:** Welcher Zahlenbereich ist mit 4 bzw. 8 Bit signed darstellbar?

-16 bis +15 bzw. -256 bis +255.

### 6.1.1 Übung Zahlenbereich

Berechnen Sie mit 4 Bit alle Kombinationen von  $+13$   $+14$  (das überschreitet den Zahlenbereich).

Halten Sie schriftlich fest, welche Ergebnisse richtig sind und wo man das Carry-Bit (die 1 an der 5. Stelle) ignorieren darf.

### 6.1.2 Übung Günter

Günter sagt: mit  $n$  Bit kann ich *signed* Zahlen von  $-2^n$  bis  $+2^n$  darstellen.

Probieren Sie für 4, 8, 16, 32 und 64 Bit, und korrigieren Sie die Formel.

## 7 Logische Bausteine

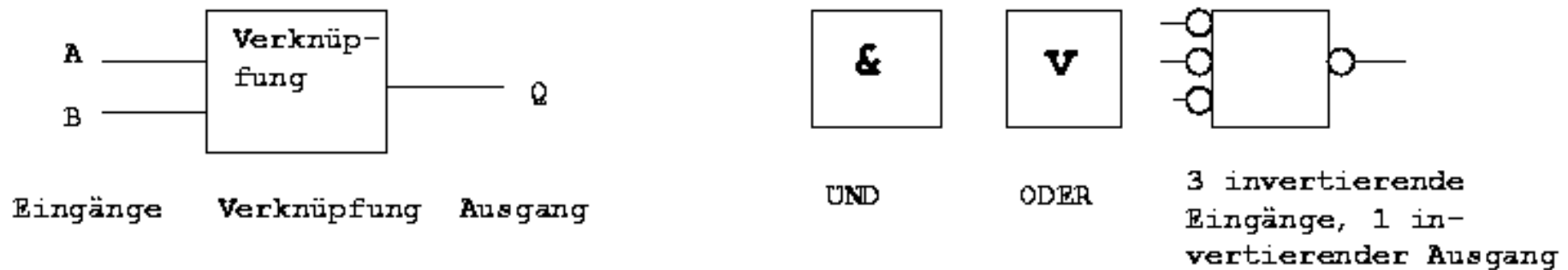
### 7.1 TTL-Level

Def.: 0..1 V = 0. 2,4..5 V = 1. Dazwischen = nicht definiert.

### 7.2 Schaltalgebra (Boole'sche Algebra) in Wahrheitstabellen

Kurze Schreibeise: 0 = falsch und 1 = wahr.

Übung: Legen Sie eine Tabelle an. Es gebe die Eingänge A und B, mit allen Kombinationen. Füllen Sie die anderen Spalten aus: A & B; A v B; A XOR B; A NAND B; A NOR B; A = B; NOT (NOT A AND NOT B); NOT (NOT A OR NOT B).



Ausgänge darf man nicht kurzschließen, sondern nur verknüpfen.  
Ausgänge darf man auftrennen.

Invertierte Signale schreiben wir mit Überstrich:  $\bar{A}$ .

Das UND-Zeichen lassen wir meistens weg. Das rechte Beispiel aus dem Bild ist  $\overline{\bar{A} \bar{B} \bar{C}}$ .

Übung : Wir stellen uns eine Und-Schaltung vor als zwei hintereinandergeschaltete Taster. Und veranschaulichen an diesem Beispiel das Kommutativgesetz für AND.

### 7.3 Gesetze

Finden Sie die folgenden Booleschen Gesetze, zB bei wikipedia:

1. Kommutativ
2. Assoziativ
3. Distributiv
4. Absorption
5. Neutrale Elemente
6. Komplementäre Elemente
7. DeMorgan'sche Theoreme. Beweisen Sie beide durch je eine Tabelle.

### 7.4 Konvertierung zu NICHT-UND

NAND ist elektronisch am einfachsten zu bauen: EIN Multi-Emitter-Transistor.

Na dann machen wir halt eine Soll-Tabelle und lesen daraus bequem die AND-Form ab. AND bauen wir dann aus NAND NAND.

Beispiel: XOR mit zwei oder drei Eingängen.

<b>A</b>	<b>B</b>	<b>A XOR B = Q</b>
----------	----------	--------------------

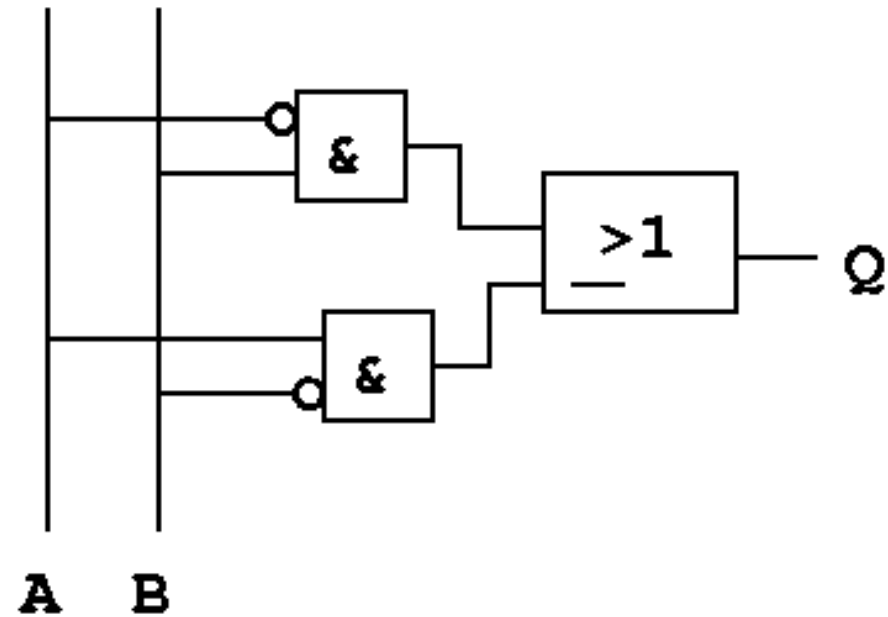
<b>0</b>	<b>0</b>	<b>0</b>
----------	----------	----------

<b>0</b>	<b>1</b>	<b>1</b>
----------	----------	----------

<b>1</b>	<b>0</b>	<b>1</b>
----------	----------	----------

<b>1</b>	<b>1</b>	<b>0</b>
----------	----------	----------

$$Q = \bar{A}B \vee A\bar{B}$$



### 7.5 vollständige Normalform

Jeder Ausgang wird als logische Verknüpfung der Eingänge angegeben.

### 7.6 Vereinfachung nach Boolescher Algebra

$$\bar{A}BCD \vee ABCD = ?$$

### 7.7 Übung: XOR mit 3 Eingängen aus NAND-Bausteinen

Mit Tabelle, Formel und Schaltung.

### 7.8 Maschinenlast

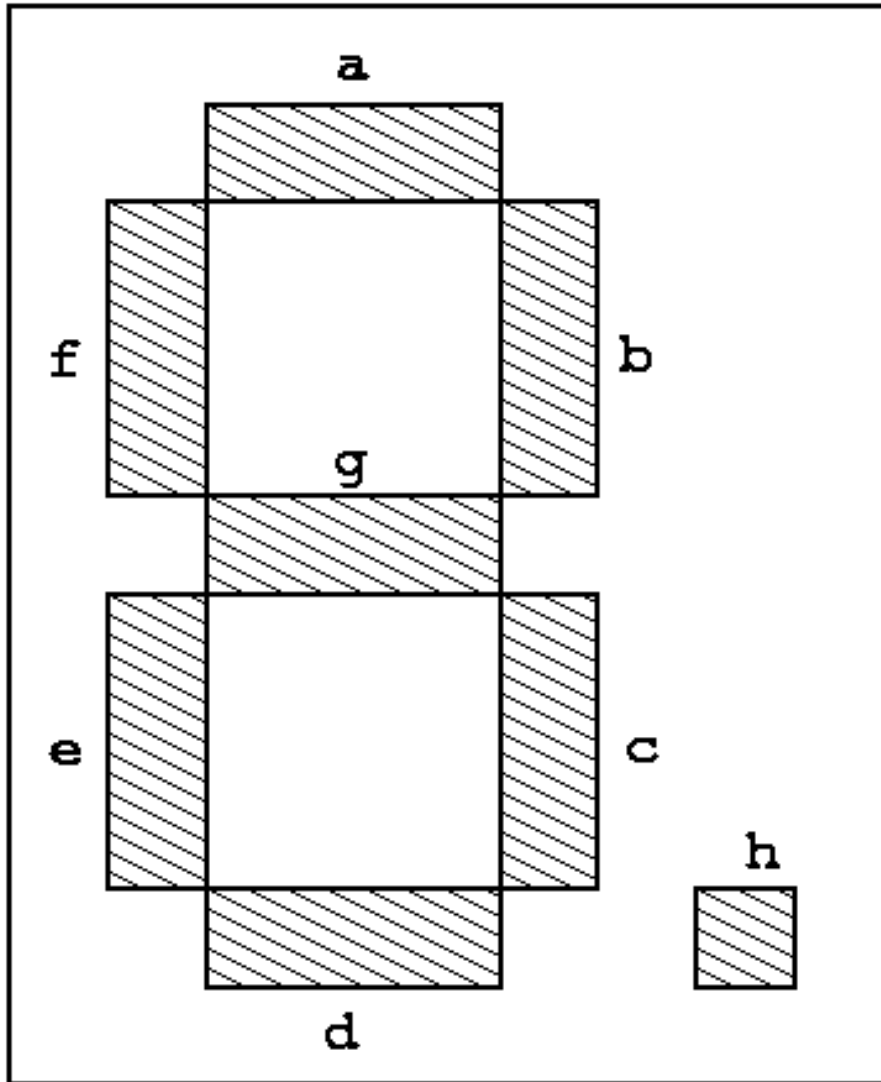
Gegeben sind 4 Signale. Sie bedeuten: Maschine A / B / C / D läuft mit 10 / 20 / 35 / 40 kW.

Gesucht ist eine Maschine mit drei Leuchtdioden. Grün leuchtet bei weniger oder gleich 30 kW. Rot leuchtet bei gleich oder mehr als 80 kW. Gelb leuchtet bei - ach finden Sie's selber heraus.

### 7.9 Übung: 7-Segment-Anzeige

Tabelle, Formel und Schaltung für alle Ausgänge.

Eine 7-Segment-Anzeige soll 4-Bit-Zahlen in hex anzeigen. Die Eingänge heißen A, B, C, D. Die Ausgänge heißen a, b, c, d, e, f, g.



1 2 3 4 5 6 7 8  
9 0 A b c d E F

### 7.10 Elektronik-Bastelkasten

Du sollst nicht an den Kabeln reißen.

Du sollst keine Ausgänge kurzschließen.

Du sollst die Kabel so im Deckel befestigen, daß die Stecker nicht beim Schließen des Deckels geklemmt werden.

## 8 Addierer

Wir können nun aus *jeder* Eingangs-Kodierung eine passende Ausgangs-Kodierung basteln. Drum wollen wir nun Binärzahlen addieren.

Rechte Stelle, Halbaddierer (HA): 2 Eingänge A B, 2 Ausgänge S (Summe) Ü (Übertrag).

Andere Stellen, Volladdierer (VA): 3 Eingänge A B Ü, 2 Ausgänge S Ü.

Übung: Tabelle, Formel und Schaltung.

## 9 Gray-Code

Wir wollen digital Längen und Winkel messen. Dazu drucken wir ein Bitmuster auf den Schlitten / Rotor und tasten den mit Fotozellen ab.  
Problem: Fotozellen und ihre Träger sitzen schief. Aus Übergang 0111-1000 wird ggf. 0111-1111-1011-1001-1000. Total doof.

Lösung:

**Beim Gray-Code wechselt pro Schritt nur 1 Bit.  
Es gibt zyklische und nicht-zyklische.**

Übung: Erfinden Sie zyklische 1-, 2-, 3- und 4-Bit-Graycodes.

Übung: 4-Bit-Gray nach 4-Bit-bin: Tabelle, Formel und Schaltung.

Lösung: Wenn ein n-Bit-Graycode gegeben ist, können wir daraus einen n+1-Bit-Graycode ermitteln wie folgt:

1. n-Bit-Graycode in umgekehrter Reihenfolge nochmal hinschreiben;
2. die erste Hälfte mit 0, die zweite mit 1 ergänzen.

Man erreicht so nur 2er-Potenzen als Anzahl der Schritte.

Man darf dabei stets am Anfang und am Ende m Zahlen streichen: das Ergebnis ist immer noch Graycode und zyklisch.

## 10 Karnaugh-Veitch-Diagramme (KV-Diagramme) (4 Bit only)

### 10.1 Wie's geht

Wir zeichnen ein 4x4 Raster. (Gibt 16 Felder, passend für 4 Bit.)

Spaltenköpfe: 2-Bit-Graycode für A und B.

Zeilenköpfe: 2-Bit-Graycode für C und D.

**2 Einsen nebeneinander oder untereinander darf man zusammenfassen.**

**4 Einsen in 1 Reihe oder im Quadrat darf man zusammenfassen.**

**Zusammenfassungsfelder dürfen überlappen.**

**Zusammenfassungsfelder dürfen über den Rand zum Anfang des Rasters gehen.**

**Allg.:  $2^n \times 2^m$  - Felder darf man zusammenfassen.**

Beispiel:

A	0	0	1	1		
B	0	1	1	0	C	D
	1	0	1	1	0	0
	1	0	0	0	0	1
	0	1	1	0	1	1
	1	1	0	1	1	0

Matrix g

$$g = \overline{BD} \vee \dots$$



das sind die vier Ecken

Dabei kommen maximal vereinfachte Formeln für jeden Ausgang heraus.

**10.2 7-Segment vereinfachen.**

Wieviele KV-Diagramme erhält man bei 4 Eingängen und 7 Ausgängen?

**10.3 Wandler 4-Bit-Gray nach bin vereinfachen.**

**10.4 Wandler 4-Bit-Gray nach BCD vereinfachen.**

BCD: Binär Codierte Dezimalzahlen. 0000B bis 1001B.

Der Elektronikkasten hat einen eingebauten Wandler BCD nach 7-Segment.

**10.5 Pseudo-Tetraden-Anzeiger vereinfachen.**

Pseudo-Tetraden sind die 4-Bit-Ziffern A bis F. Generieren Sie das Signal DasIstEinePseudoTetrade (PT).

**10.6 Halbaddierer und Volladdierer vereinfachen.**

**10.7 Maschine, die durch 4 teilbare Zahlen anzeigt, vereinfachen.**

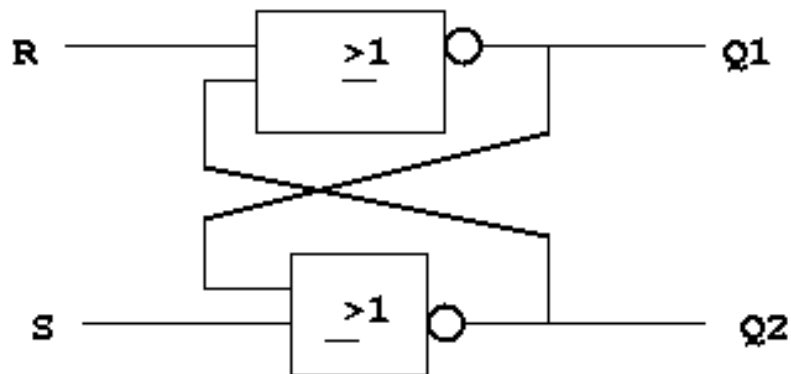
**10.8 Hardware-Multiplizierer für 2 Zahlen von je 2-Bit bauen.**

Wieviele Bits brauchen Sie für das Ergebnis???

## 11 Speicherzellen

### 11.1 1-Bit-Flipflop (FF)

Eine Schaltung, die sich etwas merken kann:



R	S	Q1
0	0	= Qalt = merkt sich was
0	1	0
1	0	1
1	1	unlogisch - BÖSE

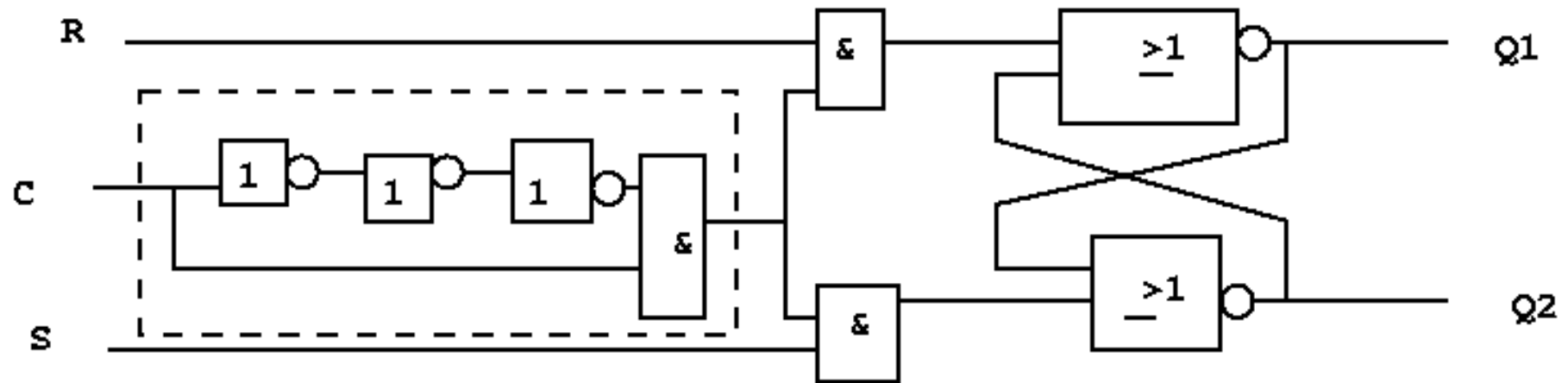
### 11.2 Taktflankengesteuertes FF

Problem: Das obige FF schaltet unerwünscht bei Knacksen auf R- oder S-Leitung.

Lösung1: Ein zweiter Eingang, eine Art "R bzw S ist jetzt gültig"-Leitung.

Lösung2: Clock und Flanke-nach-Impuls-Wandler.

Der Wandler funktioniert, weil die logischen Bausteine ("Gatter") Laufzeiten im ns-Bereich (Nano-Sekunden,  $10^{-9} \text{sec}$ ) haben.



**Pegel-nach-Impuls-Wandler**

Übung: Zeitdiagramm für Übergang C 0-1-0, Schalttabelle.

### 11.3 Master-Slave-Set-Reset-FF (MS-RS-FF)

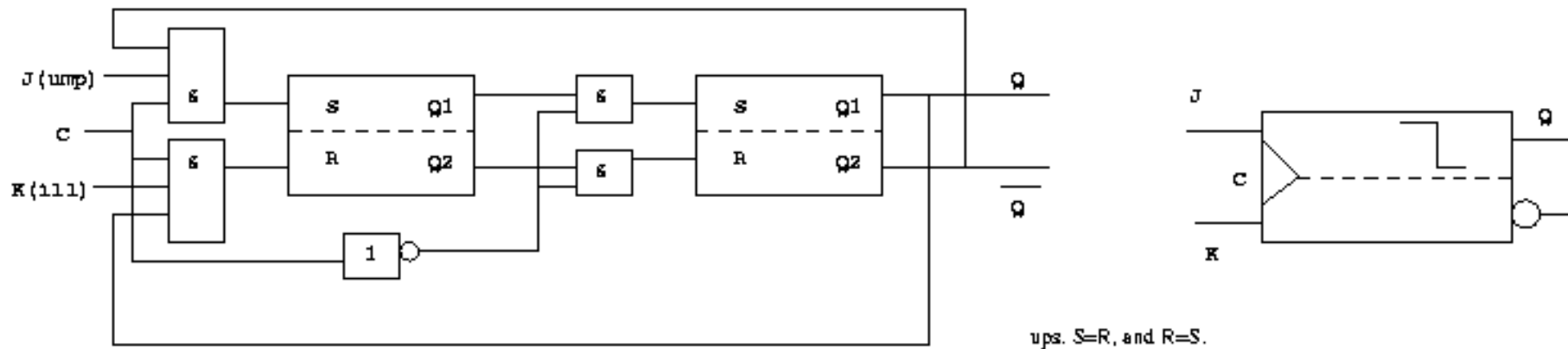
#### 11.3.1 Prellende Schalter

Problem: handbetätigte Schalter prellen.

Benutzer denkt: Jetzt habe ich Kontakt geöffnet. Kleine Sschalterzunge macht: Jetzt habe ich geöffnet-geschlossen-geöffnet-geschlossen-geöffnet-geschlossen ... hähähä.

Dasselbe beim Schließen. **Böse.** – Lösung:

#### 11.4 MSRSFF



Beobachten Sie einen kompletten Schaltvorgang für  $J=1$  und  $C=0-1-0$ .

Bei  $C=0-1$  schaltet das erste FF durch, bei  $C=1-0$  das zweite. Das entprellt alles.

Wir erstellen die Tabelle CJKQ mit Eingang 0xx, 100, 101, 110, 111 und erhalten für Q Qalt, 0, 1, NOT Qalt.

Damit haben wir im JK-RS-FF ein schönes Spielzeug, das zB Frequenzen durch 2 teilen kann. Oder 4. Oder 8.

#### 11.4.1 Duke oder Count?

Übung auf Papier: Nehmen Sie 10 Perioden einer Rechteckschwingung A, dividieren Sie sie mit 3 JKMSFF namens anton, berta, cesar, und analysieren Sie das Zeitdiagramm von A, Qanton, Qberta, Qcesar.

(Ist ein 4-Bit-Vorwärtszähler.)

Glaubt eh keiner, also basteln. Achtung: Im Bastelkasten sind keine R- und S-Eingänge, sondern NOT R und NOT S.

### **11.5 Rückwärtszähler**

Übung: Bauen Sie ihn zum Rückwärtszähler um, zuerst auf Papier, dann real.

#### **11.5.1 Move oder Shift?**

Verbinden Sie bei 4 aufeinanderfolgenden JKMSFF je Q-J und NOT Q-K, geben Sie allen denselben Uhr-Impuls, und setzen und löschen Sie das erste JKMSFF nach Belieben.

Zeitdiagramm. Ggf. basteln.

(Ist ein Lauflicht.)

## 12 Zen oder die Kunst, Digitaltechnik anzuwenden

### 12.1 Größenwahn: 2-stelliger Dezimalzähler

Problem: Durch 2 teilen ist fein, aber irgendwann langweilig. Feiner ist, wenn der Zähler von 0 bis 9 läuft.

Lösung: 4-Bit-Zähler bauen. Und Löschen, wenn er 10D (1010B) zeigt. (Ja, dann zeigt er ein winziges Moment halt 10. Na und.)

Basteln. Gutfinden. Glücklich sein.

### 12.2 Umschaltbarer Zähler

Auf Papier: Basteln Sie einen Zähler, bei dem man mit einem Schalter die Zählrichtung ändern kann: aufwärts oder abwärts.

### 12.3 Noch größer: Kursarbeit 24-Stunden-Uhr HHMMSS

Auf Papier: Basteln Sie Zähler für jede Stelle.

Basteln Sie die Reset-Logik. Sekunden und Minuten laufen bis 59. Stunden laufen bis 23.

Basteln: Sie brauchen 6 Kästen. Viel Glück und Viel Spaß.

### 12.4 Ampel an Einmündung

Gegeben ist eine Einbahn-Hauptstraße mit einer Einbahn-Einmündung, zwei Ampeln für die Autos und einer Fußgängerampel im weiteren Straßenverlauf.

Zunächst sollen die Autos von der Hauptstraße fahren, dann die von der Einmündung, dann die Fußgänger laufen. Dann von vorne.

Erstellen Sie ein vernünftiges 32-sec-Zeitschema für die roten, gelben und grünen LEDs der drei Ampeln. Namen: HSR (Hauptstraße Rot), HSG (Gelb), HSN (Grün), NSR, NSG, NSN, FGR, FGG.

Tabelle, ggf. KV-Diagramm, Schaltung, bauen, strahlen.

### 12.5 Schrittmotor

Ein Schrittmotor hat als Läufer einen Permanentmagneten und am Umfang viele kleine Spulen (vereinfacht: 4).

Spule 1 an: Läufer guckt nach oben.

Spule 1 und 2 an: Läufer dreht sich nach rechts oben.

Spule 2 an: Läufer dreht sich nach rechts.

usw.

Achtung: Die Kontakte sind auf Pin 1, 3, 5 und 7. Und sie sind *invertiert*.

Aufgabe: Drehen Sie den Motor mittels eines 3-Bit-Zählers.

Aufgabe: Stecken Sie zwei Kabel um, so daß sich der Motor andersrum dreht.

## 13 Klausur-Ideen

### 13.1 Umrechnen

Legen Sie eine Tabelle mit 5 Spalten an, Überschriften: HEX - BIN - DEC - OCT - GRM.

OCT steht für das Oktal-System mit 8 als Basis.

GRM steht für das gefürchtete 17er-System mit den Ziffern 0 bis 9 und G bis M.

Tragen Sie die folgenden Zahlen in die richtige Spalte ein und berechnen Sie die Darstellungen in den anderen Zahlensystemen:

$235_H$ ,  $56A_H$ ,  $7AF_H$ ,  $DEBB_H$ ,  $1010_B$ ,  $1010_D$ ,  $1010_O$ ,  $1010_{GRM}$

Schreiben Sie alle Nebenrechnungen auf. Ergebnisse ohne Nebenrechnung geben 0 Punkte.

### 13.2 Rechnen

Berechnen Sie die folgenden Aufgaben Dezimal und Binär. Benutzen Sie stets eine passende Darstellung von  $n * 8$  Bit Länge, dabei ist  $n$  eine ganze positive Zahl.

a)  $1023+255$       b)  $200-127$       c)  $200+(-127)$       d)  $16 * 16$       e)  $15 * 17$       f)  $255 : 3$

### 13.3 Wertebereich

Legen Sie eine Tabelle mit den Spalten (4Bit, 8Bit, 16Bit) und den Zeilen (signed, unsigned) an. Schreiben Sie sodann den Wertebereich der jeweiligen Integer hinein.

### 13.4 Exclusive-Oder

Beschreiben Sie in ein oder zwei Sätzen, was herauskommt, wenn Sie 4 Boole'sche Signale XOR-verknüpfen.

### 13.5 4-Bit-Hardware-Multizierer

### 13.6 4-Bit-Gucker: A gleich ungleich größer kleiner B