

# Versuchsvorbereitung P1-63: Digitale Elektronik, Schaltlogik

Michael Walz  
Gruppe 10

28. Oktober 2007

## Inhaltsverzeichnis

|          |   |           |
|----------|---|-----------|
| <b>0</b> | <b>Vorwort</b>                            | <b>3</b>  |
| <b>1</b> | <b>Gatter aus diskreten Bauelementen</b>  | <b>3</b>  |
| 1.1      | AND-Gatter . . . . .                      | 3         |
| 1.2      | NOT- und NAND-Gatter . . . . .            | 4         |
| 1.2.1    | NOT-Gatter . . . . .                      | 4         |
| 1.2.2    | NAND-Gatter . . . . .                     | 4         |
| 1.3      | OR-Gatter . . . . .                       | 5         |
| <b>2</b> | <b>Logische Funktionen mit ICs</b>        | <b>5</b>  |
| 2.1      | Inverter . . . . .                        | 5         |
| 2.1.1    | NOR-Gatter . . . . .                      | 5         |
| 2.2      | EXOR . . . . .                            | 6         |
| 2.3      | EXOR nur mit NAND-Gattern . . . . .       | 6         |
| <b>3</b> | <b>Addierer</b>                           | <b>6</b>  |
| 3.1      | Halbaddierer . . . . .                    | 6         |
| 3.2      | Volladdierer . . . . .                    | 7         |
| 3.3      | Subtrahierer . . . . .                    | 7         |
| <b>4</b> | <b>Flip-Flops als Speicherelemente</b>    | <b>8</b>  |
| 4.1      | RS-Flip-Flop . . . . .                    | 8         |
| 4.2      | Getaktetes RS-Flip-Flop . . . . .         | 9         |
| 4.2.1    | D-FF . . . . .                            | 9         |
| 4.3      | JK-MS-FF . . . . .                        | 9         |
| 4.3.1    | Takt springt auf 1 . . . . .              | 10        |
| 4.3.2    | Takt springt auf 0 . . . . .              | 10        |
| <b>5</b> | <b>Schieben, Multiplizieren, Rotieren</b> | <b>10</b> |
| 5.1      | 4-Bit-Schieberegister . . . . .           | 10        |
| 5.2      | Rotationsregister . . . . .               | 11        |
| <b>6</b> | <b>Zähler</b>                             | <b>11</b> |
| 6.1      | 4-Bit-Asynchrnzähler . . . . .            | 11        |
| 6.2      | Asynchrner Dezimalzähler . . . . .        | 11        |
| 6.3      | 4-Bit-Synchrnzähler . . . . .             | 11        |
| 6.4      | Synchrner Dezimalzähler . . . . .         | 12        |
| <b>7</b> | <b>Digital-Analog-Wandlung</b>            | <b>12</b> |

## 0 Vorwort

Dieser Versuch handelt von Schaltlogik, die mittels elektronischer Bauteile realisiert werden soll. Die beiden Werte der Booleschen Algebra werden durch zwei unterschiedliche Potentiale gegenüber der Erdung (0V durch Erdung und 5V) dargestellt. Dies gilt sowohl für Ein- wie auch für die Ausgänge der Schaltungen. Dabei gelten die folgenden Äquivalenzen für die Booleschen Werte, die auch nebeneinander gleichwertig gebraucht werden:

**Erdung:** falsch, 0, Low (L), nicht gesetzt

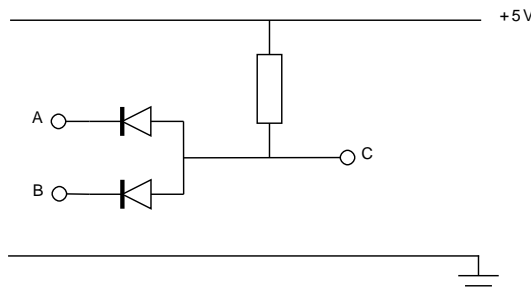
**5V:** wahr, 1, High (H), gesetzt

Dabei wirkt bei den verwendeten ICs<sup>1</sup> ein nicht angeschlossener Eingang so, als wäre er auf High gesetzt.

## 1 Gatter aus diskreten Bauelementen

### 1.1 AND-Gatter

Schaltplan:



Schaltsymbol:

Für ein AND-Gatter sollte folgende Wahrheitstabelle gelten:

| A | B | C = A ∧ B |
|---|---|-----------|
| 0 | 0 | 0         |
| 0 | 1 | 0         |
| 1 | 0 | 0         |
| 1 | 1 | 1         |

Für das obige Schaltbild ist dies erfüllt. Wenn sowohl an A und an B 5V (logisch HIGH) anliegen, so fließt kein Strom durch die Dioden, womit ihr Widerstand groß im Vergleich zum realen Widerstand wird. Fast die gesamte Spannung fällt damit an den Dioden ab. Am realen Widerstand fällt praktisch keine Spannung ab, womit der Punkt C nahe am 5V-Potential, also wahr, liegt.

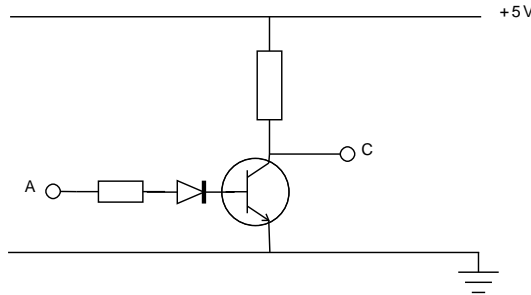
Wenn man A oder B (oder auch beide) erdet (logisch LOW), wird entsprechende Diode leitend und damit verschwindet ihr Widerstand im Vergleich zum realen Widerstand. Damit fällt die gesamte Spannung am realen Widerstand ab. Der Punkt C ist damit geerdet (logisch LOW).

<sup>1</sup>Integrated Circuits

## 1.2 NOT- und NAND-Gatter

### 1.2.1 NOT-Gatter

Schaltplan:



Schaltsymbol:

Für ein NOT-Gatter gilt die folgende Wahrheitstabelle:

| A | C = $\neg A$ |
|---|--------------|
| 0 | 1            |
| 1 | 0            |

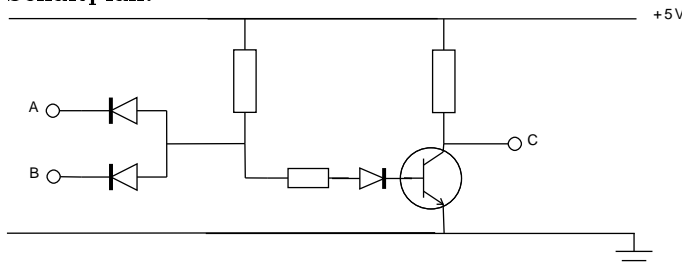
Für das obige Schaltbild ist dies erfüllt. Wenn A geerdet (logisch LOW) ist, dann ist die Spannungsdifferenz zwischen Basis und Emittor des Transistors Null, sodass dieser sperrt. Die gesamte Spannung fällt also am Transistor ab und nicht am realen Widerstand. Somit liegt an C das 5V-Potential (logisch HIGH) an.

Im umgekehrten Fall (A auf logisch HIGH) schaltet der Transistor in den leitenden Betrieb, sodass die gesamte Spannung am realen Widerstand abfällt. C liegt damit an der Erdung und damit auf logisch LOW.

Der Vorwiderstand links vom Transistor schützt die empfindliche Basis des Transistors vor der Zerstörung.

### 1.2.2 NAND-Gatter

Schaltplan:



Schaltsymbol:

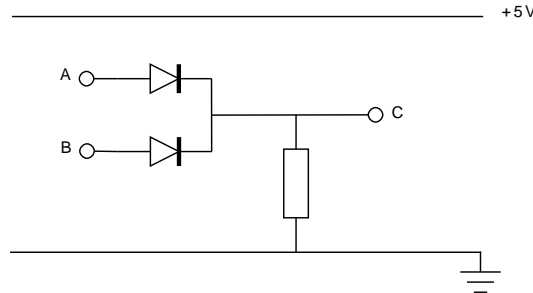
Für ein NAND-Gatter gilt die folgende Wahrheitstabelle:

| A | B | C = $\neg (A \wedge B)$ |
|---|---|-------------------------|
| 0 | 0 | 1                       |
| 0 | 1 | 1                       |
| 1 | 0 | 1                       |
| 1 | 1 | 0                       |

Für das obige Schaltbild ist dies erfüllt. Der Ausgang der AND-Schaltung wird einfach als Eingang für das NOT-Gatter benutzt. Da alle anderen logischen Schaltung aus Kombinationen von NAND-Gattern realisiert werden können, kann dieser Baustein als Standardbaustein bezeichnet werden.

### 1.3 OR-Gatter

Schaltplan:



Schaltsymbol:

Für ein OR-Gatter gilt die folgende Wahrheitstabelle:

| A | B | $C = A \vee B$ |
|---|---|----------------|
| 0 | 0 | 0              |
| 0 | 1 | 1              |
| 1 | 0 | 1              |
| 1 | 1 | 1              |

Für das obige Schaltbild ist dies erfüllt. Wenn beide Eingänge an der Erdung liegen (logisch LOW), gibt es zwischen den beteiligten Punkten (A, B, C und Erdung) kein Spannungsgefälle, sodass alle – und damit auch der Ausgang C – auf logisch LOW liegen.

Sobald an A oder B (oder an beiden) das 5V-Potential (logisch HIGH) angelegt wird, beginnen der Dioden zu leiten. Ihr Widerstand ist im Vergleich zum realen Widerstand zu vernachlässigen. An letzterem fällt daher die gesamte Spannung ab, sodass der Ausgang C auf logisch HIGH liegt.

## 2 Weitere einfache logische Funktionen, realisiert mit ICs

Im Folgenden befinden sich alle Schaltpläne und Schaltsymbole von Hand erstellt im Anhang. Die Kenntnis der Schaltbilder wird vorausgesetzt, auch wenn nicht ausdrücklich auf sie verwiesen wird.

### 2.1 Inverter

Ein Inverter lässt sich aus einem NAND- oder NOR-Gatter ganz einfach realisieren, indem der zu negierende Wert einfach an beide Eingänge gelegt wird. Dies sieht man ganz einfach aus den Wahrheitstabellen von NAND- und NOR-Gatter.

Alternativ kann auch ein Eingang fest auf einen bestimmten Wert gelegt werden. (Beim NAND-Gatter auf 1; beim NOR-Gatter auf 0) Dann entscheidet der verbliebene Eingang über den Zustand des Ausgangs, der genau komplementär zum verbliebenen Eingang ist.

#### 2.1.1 NOR-Gatter

Für das hier erstmal verwendete NOR-Gatter gilt die folgende Wahrheitstabelle:

| A | B | $C = \neg(A \vee B)$ |
|---|---|----------------------|
| 0 | 0 | 1                    |
| 0 | 1 | 0                    |
| 1 | 0 | 0                    |
| 1 | 1 | 0                    |

## 2.2 EXOR

Die EXOR-Funktion beschreibt ein exklusives Oder. Sie ist dann wahr, wenn genau einer der Eingänge wahr ist (und der andere falsch). Für die EXOR-Funktion gilt die folgenden Wahrheitstabelle:

| A | B | $C = A \oplus B$ |
|---|---|------------------|
| 0 | 0 | 0                |
| 0 | 1 | 1                |
| 1 | 0 | 1                |
| 1 | 1 | 0                |

Die disjunktive Normalform ist eine Form, in der alle Teilterme aus den konjunktiv verknüpften Variablen (oder deren Negation) bestehen. Diese Teilterme werden dann disjunktiv zur Normalform zusammengesetzt. Die disjunktive Normalform lässt sich aus der Wahrheitstabelle ablesen, indem man nur die Zeilen betrachtet, in denen der Ausgang C der Wert 1 annimmt. Für XOR lautet die disjunktive Normalform:

$$\text{XOR: } C = (\bar{A} \wedge B) \vee (A \wedge \bar{B})$$

## 2.3 EXOR nur mit NAND-Gattern

Oben wurden die NAND-Gatter bereits als Standardbausteine der digitalen Elektronik bezeichnet. Nun werden wir die disjunktive Normalform für EXOR so umformen, dass die entstehende Form sich sofort durch NAND-Gatter darstellen lässt.

$$\begin{aligned}
 C &= (\bar{A} \wedge B) \vee (A \wedge \bar{B}) && \text{disjunktive Normalform} \\
 &= (\bar{A} \wedge B) \vee (A \wedge \bar{B}) \vee (A \wedge \bar{A}) \vee (B \wedge \bar{B}) && \text{Erlaubt, da: } A = A \vee 0 \\
 &= (A \wedge (\bar{A} \vee \bar{B})) \vee (B \wedge (\bar{A} \vee \bar{B})) && \text{A bzw. B wird ausgeklammert} \\
 &= (A \wedge \overline{(A \wedge B)}) \vee (B \wedge \overline{(A \wedge B)}) && \text{DeMorgan'sche Regel} \\
 &= \overline{(A \wedge \overline{(A \wedge B)})} \wedge \overline{(B \wedge \overline{(A \wedge B)})} && \text{DeMorgan'sche Regel}
 \end{aligned}$$

Diese Formel lässt sich nun mit vier NAND-Gattern realisieren.

# 3 Addierer

## 3.1 Halbaddierer

Für einen Halbaddierer, der die beiden einstelligen Binärzahlen A und B addiert, gilt für die Summe (S) und den Übertrag (C; von engl.: carry) die folgenden Wahrheitstabelle. Der Übertrag ist notwendig, da das Ergebnis der Addition von  $1 + 1$  sich nicht mehr mit einem einzelnen Bit darstellen lässt.

| A | B | S (Summe) | C (Übertrag) |
|---|---|-----------|--------------|
| 0 | 0 | 0         | 0            |
| 0 | 1 | 1         | 0            |
| 1 | 0 | 1         | 0            |
| 1 | 1 | 0         | 1            |

Wir erkennen aus den Wahrheitswerten, dass sich für die Summe ein EXOR- und für den Übertrag ein AND-Gatter verwendet lässt.

### 3.2 Volladdierer

Um einen Volladdierer zu bauen, müssen wir nur noch einen evtl. vorhandenen Übertrag aus dem vorherigen Rechenschritt berücksichtigen. Wir lassen erstmal A und B mit dem Halbaddierer addieren (in der Tabelle als „A+B (C/S)“ dargestellt) und überlegen nun, wie aus dem Ergebnis des ersten Halbaddierers und des ankommenden Übertrags  $C_{in}$  das Ergebnis bestehend aus Summe S und Übertrag  $C_{out}$  berechnet werden kann.

| A | B | $C_{in}$ | A+B (C/S) | $C_{out}$ | S |
|---|---|----------|-----------|-----------|---|
| 0 | 0 | 0        | 0/0       | 0         | 0 |
| 0 | 0 | 1        | 0/0       | 0         | 1 |
| 0 | 1 | 0        | 0/1       | 0         | 1 |
| 0 | 1 | 1        | 0/1       | 1         | 0 |
| 1 | 0 | 0        | 0/1       | 0         | 1 |
| 1 | 0 | 1        | 0/1       | 1         | 0 |
| 1 | 1 | 0        | 1/0       | 1         | 0 |
| 1 | 1 | 1        | 1/0       | 1         | 1 |

Man erkennt, dass S genau dann 1 wahr ist, wenn entweder  $C_{in}$  oder das Summe-Bit des ersten Halbaddierers 1 waren. Dies entspricht genau dem EXOR-Baustein, der für die Summe des Halbaddierers verwendet wurde (vgl. vorherige Aufgabe). Wir schließen also den zweiten Halbaddierer an den ankommenden Übertrag  $C_{in}$  und das Summe-Bit des ersten Halbaddierers an. Nun müssen wir noch die beiden möglichen Überträge aus dem ersten und zweiten Halbaddierer zusammenführen. Da nie beide gleichzeitig gesetzt sein können, genügt dazu ein einfaches OR-Gatter.

**Anmerkung:** Da ein Volladdierer auch als „Addierer für drei einstellige Dualzahlen“ verstanden werden kann, ist die Unterscheidung am Eingang zwischen den Zahlen (A, B) und dem ankommenden Übertrag unnötig. Alle drei Eingänge können beliebig vertauscht werden.

### 3.3 Subtrahierer

Der 4-Bit-Subtrahierer wird wie in Abbildung 4 auf dem Aufgabenblatt aufgebaut. Der Baustein SN7483 besteht aus vier Volladdierern, wobei der Übertragsausgang des einen Volladdierers mit dem Übertragseingang des nächsthöheren verbunden ist. Dadurch ist nur der Übertragseingang des kleinsten (Port 13) und der Übertragsausgang des größten (Port 14) von außen erreichbar.

Zur Berechnung wird benutzt, dass im Binärsystem  $A + \bar{A} = 1111 = 10000 - 1$  gilt.

$$\begin{aligned}
 -A &= \bar{A} + 1 - 10000 \\
 B - A &= B + (-A) = B + \bar{A} + 1 - 10000
 \end{aligned}$$

Das Invertieren des Subtrahenden A wird durch die NAND-Gatter zwischen  $A_{1-4}$  und dem IC erledigt. Das Subtrahieren der 10000 wird vom SN7483 nicht geleistet und muss von der Elektronik herum ausgeführt werden.

Für den Fall einer positiven Differenz lautet das Ergebnis 1????. Die 1 liegt am Carry-Bit (Port 14) an, der mit dem Übertragseingang des ersten Volladdierers (Port 13) verbunden ist. Dadurch wird für positive Differenzen die oben beschriebene Addition um 1 erreicht. Da vom NOT-Gatter die „Carry-1“ in eine 0 invertiert wird, zeigt das SGN-Bit an, dass es sich um *keine* negative Zahl handelt. Die EXOR-Gatter für die Ausgabe für  $D_{1-4}$  bleiben in diesem Fall wirkungslos.

Für negative Differenz erhalten wir das Ergebnis in Zweierkomplementdarstellung, da wir die 10000 nicht abgezogen haben. Wir wollen aber in diesem Fall den Betrag der Differenz angeben und nur über das SGN-Bit angeben, dass die Differenz negativ ist. Die Betragsbildung erreichen wir durch erneutes Komplementieren und der Addition von 1. Da wir die negative Zahl in der Zweierkomplementdarstellung betrachten, entfällt das Abziehen der 10000.

$$\begin{aligned}
 |B - A| &= |B + \overline{A} + 1| \\
 &= \overline{B + \overline{A} + 1} + 1 \\
 &= \overline{B + \overline{A}} + (-1) + 1 \\
 &= \overline{B + \overline{A}}
 \end{aligned}$$

Den Betrag der Differenz erhalten wir also, indem wir das Ergebnis des SN7483 nochmals komplementieren. Da nun das Carry-Bit nicht gesetzt ist, wird dieses benutzt, um die Invertierung über die EXOR-Gatter zu erreichen. Nebenbei wird sowohl das SGN-Bit richtig gesetzt, wie auch sichergestellt, dass die 1 nicht mehr am Eingangsübertrag des ersten Volladdierers anliegt.

## 4 Flip-Flops als Speicherelemente

Als Flip-Flops (FF) bezeichnet man Schaltung mit zwei stabilen Zuständen. Hier bleibt die Information auch erhalten, wenn man die Eingangsinformation wieder entfernt. Sie werden als Speichersteine genutzt.

### 4.1 RS-Flip-Flop

Dieses Flip-Flop wird durch zwei NAND-Gatter realisiert. Die Schaltlogik für  $Q$  und  $\overline{Q}$  ist demnach:

$$Q = \neg(S \wedge \overline{Q}) = \neg S \vee \neg \overline{Q} \quad \overline{Q} = \neg(R \wedge Q) = \neg R \vee \neg Q$$

Die Funktionstabelle sieht folgendermaßen aus:

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

Unter der Voraussetzung, dass am Ausgang  $\overline{Q}$  wirklich immer das Inverse von  $Q$  anliegt, folgt:

(Dies gilt über die Rückkopplung von  $Q$  und  $\overline{Q}$  auch immer, es sei denn S und R sind gleichzeitig gesetzt.  $\Rightarrow$  Verbotener Zustand)

| S | R | Q | $\overline{Q}$ | Beschreibung       |
|---|---|---|----------------|--------------------|
| 0 | 0 | 1 | 1              | Verbotener Zustand |
| 0 | 1 | 1 | 0              | Setzen (S)         |
| 1 | 0 | 0 | 1              | Reset (R)          |
| 1 | 1 | Q | $\overline{Q}$ | Speicherzustand    |

Das RS-Flip-Flop speichert also seine Information solange es sich im Speicherzustand (S=R=1) befindet. Durch Absenken des Potentials an Set (S) oder Reset (R) wird das Q-Bit gesetzt oder gelöscht. Kritisch ist nur der Zustand, dass sowohl S und R auf LOW liegen. Als Folge davon enthalten  $Q$  und  $\overline{Q}$  beide den Wert 1, sodass die Information im Flip-Flop flöten gegangen ist.

## 4.2 Getaktetes RS-Flip-Flop

Ein Getaktetes RS-Flip-Flop (RST-FF) wird nach Abbildung 5 aufgebaut. Bis auf den verbotenen Zustand gilt wieder, dass  $Q = \overline{Q}$  ist. Die Schaltlogik für  $Q$  und  $\overline{Q}$  kann man aus dem Schaltplan des RST-FF ablesen:

$$\begin{aligned} Q &= \overline{\overline{Q} \wedge \overline{S} \wedge \overline{T}} \\ &= Q \vee (S \wedge T) \\ \overline{Q} &= \overline{Q \wedge R \wedge T} \\ &= \overline{Q} \vee (R \wedge T) \end{aligned}$$

Die Zustände von  $Q$  und  $\overline{Q}$  bleiben also erhalten, es sei denn, man legt S bzw. R gleichzeitig mit dem Taktsignal auf 1. Wenn  $T = 0$  ist, so bleibt die Information unabhängig von R und S erhalten. Die entsprechende Funktionstabelle lautet:

| S    | R | T | $Q$ | $\overline{Q}$ | Beschreibung                    |
|------|---|---|-----|----------------|---------------------------------|
| egal |   | 0 | $Q$ | $\overline{Q}$ | Zustand gespeichert (Takt = 0)  |
| 0    | 0 | 1 | $Q$ | $\overline{Q}$ | Zustand gespeichert (wie RS-FF) |
| 0    | 1 | 1 | 0   | 1              | Reset                           |
| 1    | 0 | 1 | 1   | 0              | Setzen                          |
| 1    | 1 | 1 | 1   | 1              | Verbotener Zustand              |

**Anmerkung:** Während beim RS-FF durch Absenken des Potentials (LOW) ein Setzen und Löschen geschah, arbeitet der RST-FF nun durch Abheben des Potentials.

### 4.2.1 D-FF

Es gibt immer noch das Problem des verbotenen Zustandes, bei dem die Information im Flip-Flop zerstört wird. Dies geschieht, wenn  $S=R=1$  ist. Da der Zustand  $S = R = 0$  zum Speichern nicht mehr benötigt wird<sup>2</sup>, kann man mit Hilfe eines Inverters  $R = \overline{S}$  erzwingen. In der Schaltung geschieht das, indem man A mit R verbindet. Der Eingang S ist dann neben dem Takt T der einzig verbliebene Eingang und wird nun D genannt. Diese Art des Flip-Flops nennt man D-FF. Seine Funktionstabelle ist deutlich kompakter.

| D    | T | $Q$ | $\overline{Q}$ | Beschreibung        |
|------|---|-----|----------------|---------------------|
| egal | 0 | $Q$ | $\overline{Q}$ | Zustand gespeichert |
| 0    | 1 | 0   | 1              | Reset               |
| 1    | 1 | 1   | 0              | Setzen              |

## 4.3 JK-Master-Slave-Flip-Flop

Der JK-Master-Slave-Flip-Flop (JK-MS-FF) besteht grob aus zwei hintereinander geschalteten RST-FF mit komplementären Takt, wobei der erste Master und der zweite Slave genannt wird. Der Master bekommt seine Daten über die Eingänge J und K (vergleichbar mit S und R bei RST-FF). Seine Ausgänge ( $q$  und  $\overline{q}$ ) sind mit den Eingängen des Slaves verbunden. Da am Slave allerdings (im Vergleich zum Master) der invertierte Takt anliegt, arbeitet das JK-MS-FF immer im Taktzyklus 0-1-0. Beim Umschalten des Taktes auf die 1 liest der Master die anliegenden Informationen von J und K ein. Da am Slave nun aber  $\overline{1} = 0$  anliegt, verarbeitet dieser bisher keine Daten. Erst beim Umschalten des Taktes auf 0 liest der Slave nun die Daten des Masters ein und speichert sie. Sie liegen nun an den Ausgängen  $Q$  und  $\overline{Q}$  an.

<sup>2</sup>Wenn  $T = 0$  ist, so bleibt die Information unabhängig von R und S erhalten.

Die Eingänge Preset P und Clear C arbeiten wie die Eingänge R und S beim RS-FF. Sie sind so geschaltet, dass sie immer Vorrang haben. Anhand der folgenden Schalttable sehen wir, das auch hier ein verbotener Zustand ( $P = C = 0$ ) existiert, der vermieden werden muss. Ansonsten lässt sich über  $P = 0$  der Ausgang  $Q$  setzen und über  $C = 0$  löschen.

| P | C | $Q$ und $q$  | $\bar{Q}$ und $\bar{q}$  | Beschreibung       |
|---|---|--------------|--------------------------|--------------------|
| 0 | 0 | 1            | 1                        | Verbotener Zustand |
| 0 | 1 | 1            | 0                        | Setzen (S)         |
| 1 | 0 | 0            | 1                        | Reset (R)          |
| 1 | 1 | $Q$ bzw. $q$ | $\bar{Q}$ bzw. $\bar{q}$ | Speicherzustand    |

Im Normalfall liegen P und C auf logisch HIGH, sodass die Eingänge keine Wirkung haben. Im Folgenden werden wir erörtern, was beim Umschalten des Taktes passiert. (Immer unter der Voraussetzung, dass  $P = C = 1$  gilt.)

#### 4.3.1 Takt springt auf 1

Der Slave-FF ist nun nicht in Betrieb, da an ihm der Takt 0 anliegt.  $Q$  und  $\bar{Q}$  bleiben also unverändert. Für den Master-FF müssen nur noch die folgenden 4 Fälle betrachtet werden, die fast genau der Schaltlogik des RST-FF entsprechen.

| J | K | $q$       | $\bar{q}$ | Beschreibung    |
|---|---|-----------|-----------|-----------------|
| 0 | 0 | 1         | 1         | Speicherzustand |
| 0 | 1 | 0         | 1         | Reset           |
| 1 | 0 | 1         | 0         | Setzen          |
| 1 | 1 | $\bar{Q}$ | $Q$       | invertieren     |

Der letzte Fall bedarf zusätzlicher Erklärung. Dieser Zustand war beim RST-FF der verbotene Zustand, der zur Zerstörung der Daten geführt hat. Da aber parallel zu den Eingängen J und K auch die Ausgänge des Slaves überkreuz mit den ersten NAND-Gattern verbunden sind, ergibt sich hier ein anderes Bild. Genau einer der Ausgänge  $Q$  oder  $\bar{Q}$  ist 0<sup>3</sup>. Damit wird nur einer der beiden Zustände  $q$  oder  $\bar{q}$  gesetzt. Durch die Überkreuz-Schaltung der Ausgänge wird hier eine Invertierung erreicht. Beim JK-MS-FF wird damit das Problem des verbotenen Zustandes weitgehend beseitigt.

#### 4.3.2 Takt springt auf 0

In diesem Fall sind die Eingänge J und K wirkungslos, da der Master keine Daten verarbeitet.  $q$  und  $\bar{q}$  bleiben unverändert. Der Slave übernimmt einfach die Daten vom Master.

$$Q = q \quad \text{und} \quad \bar{Q} = \bar{q}$$

## 5 Schieben, Multiplizieren, Rotieren

### 5.1 4-Bit-Schieberegister

Ein 4-Bit-Schieberegister wird nach Abbildung 7 aufgebaut. Es besteht aus vier JK-MS-FF, die alle am gleichen Takt angeschlossen sind. Die Ausgänge eines JK-MS-FF sind mit den Eingängen des nächsten verbunden. Dadurch übernimmt beim Durchlaufen eines Taktes 0-1-0 jedes FF den Zustand seines Vorgängers. Da das Auslesen der Eingänge zeitlich vor dem

<sup>3</sup>zumindest dann, wenn das JK-MS-FF einmal in einem validen Zustand war. Ein Beginnen mit  $J = K = 1$  ist zu vermeiden. In diesem Fall kann aber über P oder C immer noch einer der Zustände  $Q$  oder  $\bar{Q}$  gesetzt werden.

Schreiben der Ausgänge geschieht, werden die Informationen bei jedem Takt nur um einen FF weitergeben. Über C lassen sich alle 4 FFs gleichzeitig zurücksetzen. Über einen Taster lassen sich die Eingänge des ersten FF steuern:

Taster geschlossen:  $J=0$  und  $K=1$       Taster offen:  $J=1$  und  $K=0$

Die Zustände der einzelnen FF kann man parallel an den Punkten  $Q_{A-D}$  abgreifen. Die Schaltung kann deshalb auch als Seriell-Parallel-Wandler verstanden werden. Als Taktgeber kann man keinen normalen Schalter benutzen, da dieser immer etwas „prellt“. Wir werden ein kleines FF nach Abbildung 10 dazu benutzen.

## 5.2 Rotationsregister

Wenn man nun die Eingänge des ersten Flip-Flops mit den Ausgängen des letzten Flip-Flops verbindet, laufen die Bit mit jedem Takt einen Schritt im Kreis herum. Die Daten können gleichzeitig über die Preset-Eingänge der FF geschrieben werden. Man kann sie anschließend seriell an einem Ausgang  $Q_{A-D}$  seriell abgreifen. Die Schaltung kann deshalb auch als Parallel-Seriell-Wandler verstanden werden.

# 6 Zähler

## 6.1 4-Bit-Asynchrone Zähler

Nach Abbildung 8 wird ein 4-Bit-Asynchrone Zähler aufgebaut. Es werden vier JK-MS-FF hintereinander geschaltet. Die Ausgänge der FFs sind an den Takt des jeweils nächsten angeschlossen. Am ersten FF liegt ein externer Takt an. Die Eingänge J und K sind nie angeschlossen und liegen damit auf 1. Dadurch invertieren die FFs bei jedem Takt. Am Anfang müssen jedoch über C (Clear) alle FFs erstmal in einen definierten Zustand gebracht werden, um Inkonsistenzen zu vermeiden (vgl. JK-MS-FF). Jedes FF kippt, wenn sein Takt einmal den Zyklus 0-1-0 durchlaufen hat. Das erste Flip-Flop kippt bei jedem Zyklus. Das zweite nur bei jedem zweiten und das dritte erst bei jedem vierten etc., da sie ihren Takt über die Ausgänge der vorherigen FFs beziehen. Die Ausgänge  $Q_{A-D}$  zeigen also die Anzahl der Taktzyklen in binärer Form an. Dieser Zähler heißt asynchron, da die FFs nicht alle gleichzeitig schalten, da sich an verschiedenen Takten angeschlossen sind.

## 6.2 Asynchroner Dezimalzähler

Wenn wir nun die Ausgänge  $Q_B$  und  $Q_D$  an ein NAND-Gatter und das Ergebnis am globalen Clear anschließen, erreichen wir aus der oberen Schaltung einen Dezimalzähler. Sobald das zweite und vierte Bit gesetzt sind, der Zähler also den Wert 10 ( $2^1 + 2^3 = 10$ ) erreicht hat, wird der Zähler gelöscht. Damit kann von 0-9 gezählt werden. Den Wert, der an Clear anliegt, könnte man bei einem weiteren Dezimalzähler als Takt anschließen und so den Zehnerübertrag realisieren.

## 6.3 4-Bit-Synchrone Zähler

Die Synchronität bedingt, dass alle 4 JK-MS-FF am gleichen Takt anliegen. Wir müssen nach anderen Möglichkeiten suchen, um das Umspringen der FFs zu steuern. Dies geschieht hier über die Eingänge J und K. Wenn das entsprechende FF umspringen soll, legen wir an beide Eingänge auf 1. Ansonsten liegen die Eingänge auf 0. Am Gesamteingang E muss wieder ein Taktzyklus 0-1-0 angelegt werden, um den Zähler hochzuzählen. Dazu wird wieder die Entprellungsschaltung aus Abbildung 10 benötigt. Ein Flip-Flop darf erst dann umkippen,

wenn das vorangehende FF schon gesetzt ist und wenn dieses FF auch kippen würde (also an dessen Eingang 1 anliegt). Deshalb verbinden wir einfach Ein- und Ausgang des vorherigen FF über ein AND-Gatter mit dem Eingang des folgenden FFs.

#### 6.4 Synchroner Dezimalzähler

Der synchroner Dezimalzähler nach Abbildung 9 funktioniert im Wesentlichen genauso, wie der 4-Bit-Synchronzähler. Es muss nur zusätzlich beachtet werden, dass nach der 9 (Binär: 1001) wieder die 0 (Binär: 0000) kommt. Dies erreichen wir, indem wir den zweiten JK-MS-FF am Schalten über ein AND-Gatter dann hindern, wenn das vierte Bit gesetzt ist. Durch Anschluss des Ausgangs des ersten FF am K-Eingang des vierten, erreichen wir, dass das vierte Bit immer gelöscht wird, wenn das erste Bit zurück auf 0 fällt. Dies ist aber erst bei 9 das erste Mal von Relevanz. Davor ist das vierte Bit sowieso nicht gesetzt.

## 7 Digital-Analog-Wandlung

Im Folgenden soll einer der obigen Dezimalzähler am Messgerät angeschlossen werden, um die Zahl des Zählers analog auszugeben. Die Ausgänge des Zählers werden über Widerstände am einen Anschluss der Messgerätes angeschlossen. Der andere Anschluss wird an das 0V-Potential gelegt. Dabei soll beim Zählerstand 9 gerade 90% des Vollausschlages ( $I_{\max} = 100\mu A$ ) angezeigt werden. Daraus folgt, dass aus den Ausgängen  $Q_{A-D}$  die Ströme  $10\mu A$ ,  $20\mu A$ ,  $40\mu A$  und  $80\mu A$  fließen müssen. Da an den gesetzten Ausgängen ein Potential von je  $U = 4V$  anliegt, ergibt die einfache Rechnung  $R = U/I$ :

$$\begin{aligned}R_{Q_A} &= 400k\Omega \\R_{Q_B} &= 200k\Omega \\R_{Q_C} &= 100k\Omega \\R_{Q_D} &= 50k\Omega\end{aligned}$$

Davon kann man nun noch den Innenwiderstand des Messgerätes  $R_i = 1k\Omega$  abziehen. Zusätzlich würde es sich anbieten, zur Sicherheit zwischen den Widerständen und dem Messgerät noch je eine Diode einzubauen, damit die Ausgänge  $Q_{A-D}$  auf keinen Fall miteinander wechselwirken, auch wenn dies wegen den großen Widerständen doch recht unwahrscheinlich ist.