

Prak.: P1 Semester: WS19/20 Wochentag: Di Gruppennr.: 11

Name: Hohenstein Vorname: Michael

Name: Michel Vorname: Alexis

Emailadresse(n): alexis.michel@gmx.de, michael.hohenstein@gmx.de

Versuch: Schaltlogik (P1-2) Fehlerrech.: Nein

Betreuer: Ida Göbel Durchgeführt am: 29.10.19

Wird vom Betreuer ausgefüllt.

1. Abgabe am: \_\_\_\_\_

Rückgabe am: \_\_\_\_\_ Begründung:

2. Abgabe am: \_\_\_\_\_

Ergebnis: + / 0 / - Fehlerrechnung: Ja / Nein

Datum: \_\_\_\_\_ Handzeichen: \_\_\_\_\_

Bemerkungen:



# Inhaltsverzeichnis

<b>1</b>	<b>Aufgabe: Gatter aus diskreten Bauelementen</b>	<b>1</b>
1.1	AND-Gatter . . . . .	1
1.2	NOT- und NAND-Gatter . . . . .	2
1.3	OR-Gatter . . . . .	4
<b>2</b>	<b>Aufgabe: Weitere einfache Gatter, realisiert mit ICs</b>	<b>6</b>
2.1	Inverter (NOT-Gatter) aus NAND- oder NOR-Gatter . . . . .	6
2.2	XOR-Gatter . . . . .	7
2.3	XOR mit NAND-Gattern . . . . .	8
<b>3</b>	<b>Aufgabe: Addierer</b>	<b>9</b>
3.1	Halbaddierer . . . . .	9
3.2	Volladdierer . . . . .	9
<b>4</b>	<b>Aufgabe: Speicherelemente:</b>	<b>11</b>
4.1	RS-Flip-Flop . . . . .	11
4.2	Entprellen eines Schalters . . . . .	12
4.3	Getaktetes RS-Flip-Flop . . . . .	13
4.4	D-Flip-Flop . . . . .	14
4.5	JK-Master-Slave-Flip-Flop . . . . .	14
<b>5</b>	<b>Aufgabe: Schieben, Multiplizieren, Rotieren</b>	<b>17</b>
5.1	4-Bit-Schieberegister . . . . .	17
5.2	4-Bit-Rotationsregister . . . . .	18
<b>6</b>	<b>Aufgabe: Zähler</b>	<b>19</b>
6.1	4-Bit-Asynchrone Zähler . . . . .	19
<b>7</b>	<b>Durchführung</b>	<b>20</b>

# Abbildungsverzeichnis

1.1	Schaltplan Dioden And-Gatter . . . . .	1
1.2	Schaltsymbol And-Gatter. Links befinden sich die Eingangspins A und B und rechts der Ausgangspin C . . . . .	2
1.3	Schaltplan Transistor Not-Gatter . . . . .	3

1.4	Schaltsymbol Not-Gatter. Links befindet sich der Eingangspin A und rechts der Ausgangspin B . . . . .	3
1.5	Schaltplan Nand-Gatter aus einem And- und einem Not-Gatter . . . . .	4
1.6	Schaltnetz Nand-Gatter aus einem And- und einem Not-Gatter . . . . .	4
1.7	Schaltsymbol Nand-Gatter. Links befinden sich die Eingangspins A und B rechts der Ausgangspin C . . . . .	4
1.8	Schaltplan Dioden OR-Gatter . . . . .	4
1.9	Schaltsymbol OR-Gatter. Links befinden sich die Eingangspins A und B und rechts der Ausgangspin C . . . . .	5
2.1	Not-Gatter aus Nand- oder Nor-Gattern . . . . .	7
2.2	XOR-Gatter aus Nand-, And- und Or-Gattern . . . . .	8
2.3	XOR-Gatter aus Nand-Gattern . . . . .	8
3.1	Schaltnetz Halbaddierer . . . . .	9
3.2	Schaltnetz Volladdierer mit einem dritten Eingang C = (Ü-1) . . . . .	10
4.1	Schaltnetz eines RS-Flip-Flops . . . . .	11
4.2	Schaltnetz eines Entprellers aus einem RS-Flip-Flop . . . . .	12
4.3	Schaltnetz eines RST-Flip-Flops . . . . .	13
4.4	Schaltnetz eines D-Flip-Flops . . . . .	14
4.5	Schaltnetz eines JK-MS-Flip-Flops . . . . .	15
5.1	Schaltnetz eines 4-Bit-Schieberegisters . . . . .	17
5.2	Schaltnetz eines 4-Bit-Rotationsregisters . . . . .	18
6.1	Schaltnetz eines 4-Bit-Asynchrnzählers . . . . .	19

## Tabellenverzeichnis

1.1	Wahrheitstabelle AND-Gatter . . . . .	1
1.2	Wahrheitstabelle NOT-Gatter . . . . .	2
1.3	Wahrheitstabelle NAND-Gatter . . . . .	3
1.4	Wahrheitstabelle OR-Gatter . . . . .	4
2.1	Tabelle der verwendeten ICs anstelle der Gatter . . . . .	6
2.2	Wahrheitstabellen Not-, Nand- und Nor-Gatter . . . . .	6
2.3	Wahrheitstabelle XOR-Gatter . . . . .	7
3.1	Wahrheitstabelle Halbaddierer . . . . .	9
3.2	Wahrheitstabelle Volladdierer . . . . .	10
4.1	Zustandstabelle RS-Flip-Flop . . . . .	11
4.2	Zustandstabelle RS-Flip-Flop als Entpreller . . . . .	13
4.3	Zustandstabelle RST-Flip-Flop . . . . .	13
4.4	Zustandstabelle D-Flip-Flop . . . . .	14
4.5	Zustandstabelle JK-MS-Flip-Flop . . . . .	16

# 1. Aufgabe: Gatter aus diskreten Bauelementen

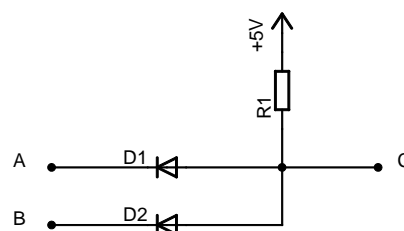
## 1.1 AND-Gatter

Bei dem ersten zu realisierenden Gatter handelt es sich um ein And-Gatter. Dieses Verundet zwei Eingangssignale A und B. Die Wahrheitstabelle für das Und-Gatter ist in Tabelle 1.1 definiert.

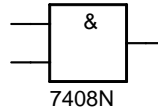
**Tabelle 1.1:** Wahrheitstabelle AND-Gatter

A	B	$C=A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Technisch wird das And-Gatter über zwei Dioden realisiert. Der dazugehörige Schaltplan wird in Abbildung 1.1 dargestellt. Dieser funktioniert dadurch, dass durch den Pull-Up Widerstand  $R_1$  der Ausgangspin C auf eine logische 1 gezogen wird, was über ein Anlegen einer positiven Spannung realisiert wird. In diesem Versuch wurde für eine logische 1 eine Spannung von 5 V verwendet. Man spricht in diesem Fall auch von einem HIGH-Signal, wenn eine positive Spannung anliegt. Liegt an beiden Eingangspins eine 1 an, so liegt zwischen den Diodenenden kein Potentialunterschied vor. Die für die Verwendung der Diode erforderliche Schleusenspannung liegt nicht vor und die Diode sperrt. Einzig und allein der Pull-Up Widerstand sorgt nun für die Ausgabe einer 1. Wird wiederum an beiden Eingangspins eine 0 angelegt, so bricht die Spannung an dem Stromknoten, an dem C verbunden ist, auf 0 V zusammen. In diesem Fall spricht man auch von einem LOW-Signal. Dies liegt daran, dass die Dioden in Durchlassrichtung betrieben werden und somit der Ausgangspin C mit der Erdung kurzgeschlossen wird. Die komplette Spannung



**Abbildung 1.1:** Schaltplan Dioden And-Gatter



**Abbildung 1.2:** Schaltsymbol And-Gatter. Links befinden sich die Eingangspins A und B und rechts der Ausgangspin C

fällt am Widerstand  $R_1$  ab, weil der Pull-Up Widerstand  $R_1$  einen viel größeren Widerstand als die offenen Dioden hat (Widerstand der Dioden ist nahe  $0\Omega \ll R_1$ ). Wird nun an nur einem der beiden Eingänge eine 1 angelegt, während an dem anderen eine 0 angelegt wird, so sperrt die Diode an dem Eingang mit der anliegenden 1 und die Diode an dem Eingang mit der anliegenden 0 befindet sich in Durchlassrichtung. Es geschieht dasselbe wie in dem Fall, in dem an beiden Ausgängen eine 0 anliegt, weil der Ausgangspin C mit der Erdung kurzgeschlossen wird, während das andere Eingangssignal aufgrund der sperrenden Diode keinen Einfluss auf das Ausgangssignal nehmen kann. Wären die Dioden nicht vorhanden, so wären die Eingänge A und B kurzgeschlossen und es würde bedingt durch den Potentialunterschied ein Strom fließen. Dieser ist unerwünscht und kann je nach Spannungsquelle, die an den Eingängen anliegt, mit einer großen Stromstärke ausfallen und elektronische Bauteile beschädigen.

Außerdem sind die Dioden wichtig, um unerwünschte Ströme am Ausgang des Gatters zu verhindern. Falls am Eingang A oder B eine Spannung  $> 5\text{ V}$  anliegt und keine Diode vorhanden wäre, so würde der Ausgang C ebenfalls diese Spannung ausgeben. Dadurch könnten elektronische Bauteile die an dem Ausgang angeschlossen sind, beschädigt werden.

Vereinfacht kann für das And-Gatter auch das Schaltsymbol in Abbildung 1.2 verwendet werden.

Ein And-Gatter lässt sich auch mit beliebig vielen Eingängen realisieren, indem weitere Dioden parallel zu den bestehenden hinzugefügt werden. Dementsprechend wird das Schaltsymbol auch um weitere Eingänge erweitert.

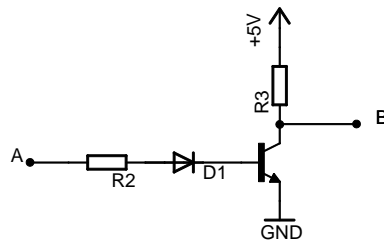
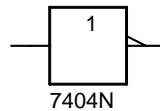
## 1.2 NOT- und NAND-Gatter

Ein logisches Nicht-Gatter gibt das Inverse seines Eingangssignals aus. Die Definition und Wahrheitstabelle des Nicht-Gatters kann in Tabelle 1.2 gefunden werden.

**Tabelle 1.2:** Wahrheitstabelle NOT-Gatter

A	$B = \neg A$
0	1
1	0

Ein Not-Gatter kann über einen NPN-Transistor realisiert werden. Wird an der Basis des Transistors ein HIGH-Signal angelegt, so lässt dieser den Strom vom Emitter zum Kollektor fließen (physikalische Stromrichtung). Dabei benötigt die Basis des Transistors einen Vorwiderstand, weil diese sonst aufgrund von zu großen Strömen kaputt geht. Liegt an der Basis ein LOW-Signal an, so sperrt der Transistor den Strom zwischen Emitter und Kollektor. Der Transistor kann am Kollektor das Inverse des an der Basis anliegenden Signals ausgeben. Wenn man den Emitter des Transistors erdet, so wie es für den Betrieb eines NPN-Transistors notwendig ist und den Kollektor mit einem Pull-Up Widerstand, ähnlich dem beim Dioden And-Gatter, auf HIGH zieht, kann man diese Schaltung als

**Abbildung 1.3:** Schaltplan Transistor Not-Gatter**Abbildung 1.4:** Schaltsymbol Not-Gatter. Links befindet sich der Eingangspin A und rechts der Ausgangspin B

Not-Gatter verwenden. Dabei liegt an dem Kollektor das Ausgangssignal des Gatters an. Abbildung 1.3 zeigt einen Schaltplan von der Schaltung. Dabei befindet sich bei Abbildung 1.3 zwischen dem Vorwiderstand und der Basis des Transistors eine Diode. Diese ist optional, weil der Transistor auch einen Strom vom Emitter zur Basis zulässt (physikalische Stromrichtung) aber nicht in die andere Richtung.

Vereinfacht kann für das Not-Gatter auch das Schaltsymbol in Abbildung 1.4 verwendet werden.

Das Not-Gatter kann mit dem And-Gatter kombiniert werden. Wenn der Ausgang des And-Gatters mit dem Eingang des Not-Gatters verbunden wird, ergibt sich daraus ein Nand-Gatter. Die Wahrheitstabelle eines Nand-Gatters wird in Tabelle 1.3 abgebildet. Ihr Ausgangssignal ist das Inverse des Ausgangssignals eines And-Gatters, dessen Wahrheitstabelle in Tabelle 1.1 dargestellt ist.

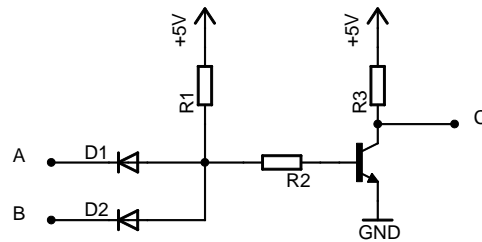
**Tabelle 1.3:** Wahrheitstabelle NAND-Gatter

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

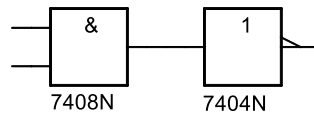
Der Zusammenschluss aus einem And- und einem Not-Gatter als Schaltplan, wird noch einmal in Abbildung 1.5 gezeigt. Da hier eine fehlerhafte Beschaltung der Basis des Transistors nicht möglich ist, wurde auf die Diode zwischen dem Vorwiderstand und der Basis des Transistors verzichtet.

Anstelle des Schaltplans kann zur Darstellung auch das Schaltnetz verwendet werden. Dieses ist in Abbildung 1.6 zu finden.

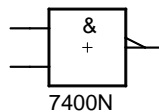
Vereinfacht kann für das Nand-Gatter auch das Schaltsymbol in Abbildung 1.7 verwendet werden.



**Abbildung 1.5:** Schaltplan Nand-Gatter aus einem And- und einem Not-Gatter



**Abbildung 1.6:** Schaltnetz Nand-Gatter aus einem And- und einem Not-Gatter



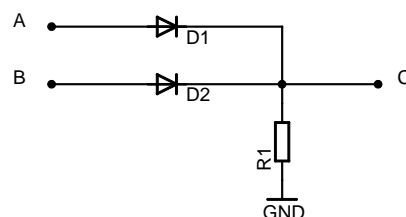
**Abbildung 1.7:** Schaltsymbol Nand-Gatter. Links befinden sich die Eingangspins A und B rechts der Ausgangspin C

### 1.3 OR-Gatter

Ein Oder-Gatter dient dazu, zwei Eingangssignale zu verodern. Die dazugehörige Wahrheitstabelle ist Tabelle 1.4 zu finden.

**Tabelle 1.4:** Wahrheitstabelle OR-Gatter

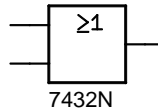
A	B	$C = A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1



**Abbildung 1.8:** Schaltplan Dioden OR-Gatter

Technisch wird das Or-Gatter über zwei Dioden realisiert. Der dazugehörige Schaltplan wird in Abbildung 1.8 dargestellt. Der Pull-down Widerstand  $R_1$  zieht den Ausgangspin auf LOW. Sind beide Eingangssignale LOW, so wird eine 0 ausgegeben, weil zwischen





**Abbildung 1.9:** Schaltsymbol OR-Gatter. Links befinden sich die Eingangspins A und B und rechts der Ausgangspin C

den Enden der Dioden keine Spannung anliegt und die Schleusenspannung nicht erreicht wird. Dadurch sperren die Dioden. Liegt stattdessen an mindestens einer der Dioden ein HIGH-Signal an, so befindet sich die Diode in Durchlassrichtung. Der Ausgangspin wird an das anliegende HIGH-Signal kurzgeschlossen und gibt eine logische 1 aus.

Die Dioden erfüllen hierbei dieselbe Funktion wie bei dem And-Gatter: Falls an den Eingängen unterschiedliche Spannungen anliegen, sollen Ströme zwischen den Eingängen untereinander blockiert werden. Außerdem sind die Dioden wichtig, um unerwünschte Ströme an den Eingängen des Gatters zu verhindern. Falls am Eingang A oder B eine negative Spannung  $< 0\text{ V}$  anliegt und keine Diode vorhanden wäre, so würde der Ausgang C ebenfalls diese Spannung ausgeben. Infolge dessen würde ein Strom von der Erdung zu den Eingängen fließen. Dieser unerwünschte Fehlerstrom kann elektronische Bauteile, die an dem Ausgang angeschlossen sind, beschädigen.

Vereinfacht kann für das Or-Gatter auch das Schaltsymbol in Abbildung 1.9 verwendet werden.

Ein Or-Gatter lässt sich auch mit beliebig vielen Eingängen realisieren, indem weitere Dioden parallel zu den bestehenden hinzugefügt werden. Dementsprechend wird das Schaltsymbol auch um weitere Eingänge erweitert.

## 2. Aufgabe: Weitere einfache Gatter, realisiert mit ICs

### 2.1 Inverter (NOT-Gatter) aus NAND- oder NOR-Gatter

Ab diesem Teil des Versuches werden die And-, Or-, Not-Gatter und Kombinationen daraus nicht mehr aus Dioden und Transistoren selber gebaut sondern in schon fertigen ICs verwendet. Dafür kommen ICs zum Einsatz, die in Tabelle 2.1 beschrieben werden.

**Tabelle 2.1:** Tabelle der verwendeten ICs anstelle der Gatter

IC	Gatter
7400	Nand-Gate
7402	Nor-Gate
7408	And-Gate
7410	Tripple-Nand-Gate
7432	Or-Gate

Das Ziel dieser Teilaufgabe bestand darin, ein Not-Gatter zu bauen und dafür nur Nand- oder Nor-Gatter zu verwenden. Um dies zu realisieren, wird zuallererst in Tabelle 2.2 ein Blick auf die Wahrheitstabellen der Gatter geworfen.

**Tabelle 2.2:** Wahrheitstabellen Not-, Nand- und Nor-Gatter

A	B	$\neg A$	$\neg(A \wedge B)$	$\neg(A \vee B)$
0	0	1	1	1
0	1	1	1	0
1	0	0	1	0
1	1	0	0	0

Aus Tabelle 2.2 lässt sich folgende Besonderheit erkennen. Sind bei einem Nand- oder Nor-Gatter die Eingänge A und B gleich, so ist der Ausgang C identisch mit  $C = \neg A = \neg B$ . Wenn man also bei einem Nand- oder einem Nor-Gatter die zwei Eingänge kurzschließt, so verringert sich die Anzahl an Eingängen auf einen einzelnen, weil dann nämlich gilt:  $A=B$ . Das Ergebnis ist ein Not-Gatter.

Eine andere Auffälligkeit in Tabelle 2.2 liegt darin, dass für den Fall  $B=1$  ein Nand-Gatter sich identisch zu  $C = \neg A$  verhält. Dasselbe gilt für ein Nor-Gatter, wenn  $B=0$ . Also kann man ein Not-Gatter auch bilden, indem man bei einem Nand-Gatter einen Ausgang dauerhaft auf HIGH setzt oder bei einem Nor-Gatter einen Ausgang dauerhaft auf LOW

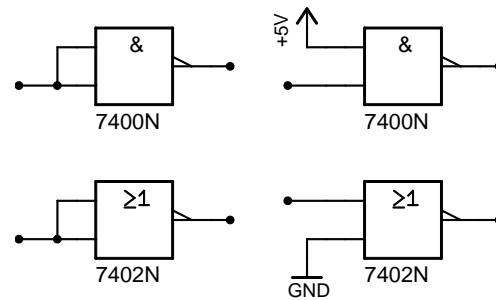


Abbildung 2.1: Not-Gatter aus Nand- oder Nor-Gattern

setzt. Man kann den Eingang des Nand-Gatters auch einfach offen lassen. Die ICs haben an allen Eingängen Pull-up Widerstände.

Die Ergebnisse werden noch einmal in Abbildung 2.1 dargestellt.

## 2.2 XOR-Gatter

Ein Xor-Gatter ist ein exklusives Or-Gatter. Man kann es beschreiben als ein „entweder, oder“ Gatter. Die dazugehörige Wahrheitstabelle wird in Tabelle 2.3 abgebildet.

Tabelle 2.3: Wahrheitstabelle XOR-Gatter

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

Das Ziel dieser Teilaufgabe ist es, ein XOR-Gatter aus Nand-, And- und Or-Gattern zu bauen. Dazu wird zu allererst die Wahrheitstabelle für ein Xor-Gatter in Tabelle 2.3 aufgestellt. Aus dieser kann abgelesen werden, dass der Ausgang des XOR-Gatters 1 ist, wenn entweder  $A=1$  und  $B=0$  oder wenn  $A=0$  und  $B=1$ . Daraus kann die disjunktive Normalform für das XOR-Gatter (Gleichung 2.1) aufgestellt werden.

$$A \vee B = (A \wedge \neg B) \vee (\neg A \wedge B) \quad (2.1)$$

Mithilfe der disjunktiven Normalform kann nun das Schaltnetz für das Xor-Gatter aufgestellt werden. Dazu braucht man ein Or-Gatter, zwei And-Gatter und zwei Not-Gatter, wie man aus Gleichung 2.1 ablesen kann. Da aber keine Not-Gatter zur Verfügung stehen, werden diese aus zwei Nand-Gattern gebaut. Wie dies realisiert werden kann, zeigt Abbildung 2.1. Das fertige Ergebnis ist in Abbildung 2.2 zu finden.

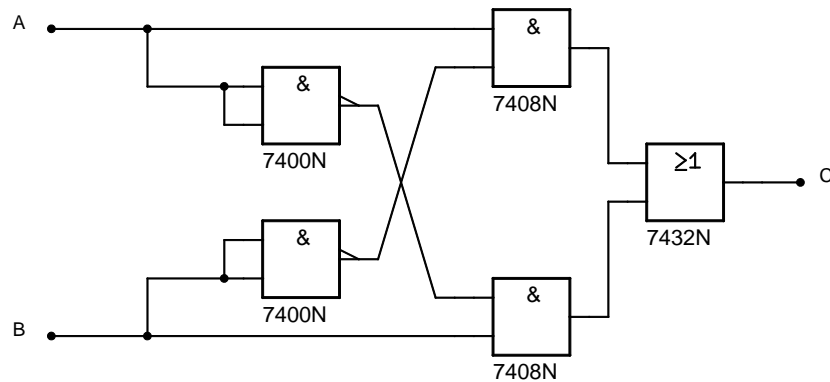


Abbildung 2.2: XOR-Gatter aus Nand-, And- und Or-Gattern

### 2.3 XOR mit NAND-Gattern

Nun geht es darum, ein XOR-Gatter zu bauen und dafür nur Nand-Gatter zu verwenden. Dafür muss die disjunktive Normalform des Xor-Gatters erst einmal umgeformt werden. Um Umformungen an Schaltnetzen mit Nand- und Nor-Gattern durchzuführen, sind die DeMorganschen Regeln sehr hilfreich:

$$\neg(A \wedge B) = (\neg A) \vee (\neg B) \quad (2.2)$$

$$\neg(A \vee B) = (\neg A) \wedge (\neg B) \quad (2.3)$$

Mithilfe dieser Regeln kann die Gleichung 2.1 in die Nand-Schreibweise umgeformt werden.

$$\begin{aligned} A \vee B &= (A \wedge \neg B) \vee (\neg A \wedge B) \\ &= (A \wedge \overline{A}) \vee (A \wedge \overline{B}) \vee (\overline{B} \wedge A) \vee (B \wedge \overline{B}) \\ &= (A \wedge (\overline{A} \vee \overline{B})) \vee (B \wedge (\overline{A} \vee \overline{B})) \\ &= (A \wedge \overline{A \wedge B}) \vee (B \wedge \overline{A \wedge B}) \\ &= \overline{\overline{A \wedge B} \overline{B \wedge A}} \\ &= \overline{\overline{A \wedge B} \overline{B \wedge A}} \end{aligned}$$

Diese Gleichung für das Xor besteht nur aus Nand-Gattern. Der Vorteil daran ist, dass man für die Realisierung der Schaltung mit deutlich weniger ICs zurecht kommt.

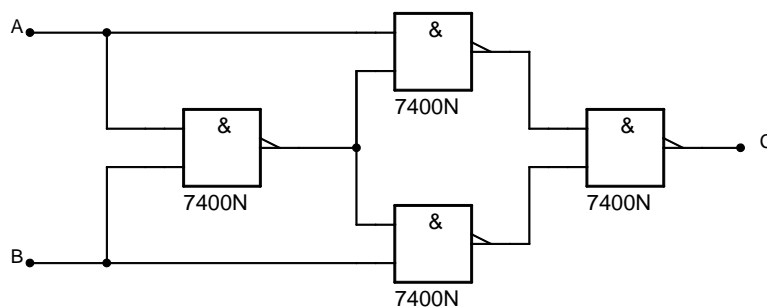


Abbildung 2.3: XOR-Gatter aus Nand-Gattern

## 3. Aufgabe: Addierer

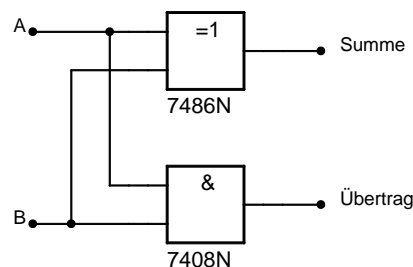
### 3.1 Halbaddierer

Ein Halbaddierer ist eine Schaltung, die in der Lage ist, zwei einstellige Binärzahlen zu addieren. Dabei repräsentiert die logische 0 in der elektronischen Schaltung eine binäre 0 und die logische 1 eine binäre 1. Die Wahrheitstabelle dieser Schaltung, welche die binären Zahlen A und B addieren soll und die Summe C sowie den Übertrag D ausgeben soll, wird in Tabelle 3.1 aufgeführt.

**Tabelle 3.1:** Wahrheitstabelle Halbaddierer

A	B	Summe	Übertrag
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Dabei fällt auf, dass die Wahrheitstabelle der Summe identisch zu der des Xor-Gatters ist und die Wahrheitstabelle des Übertrags identisch zu der des And-Gatters ist. Daraus kann man das Schaltnetz für den Halbaddierer generieren. Abbildung 3.1 zeigt das Ergebnis.



**Abbildung 3.1:** Schaltnetz Halbaddierer

### 3.2 Volladdierer

Der Halbaddierer ist nur in der Lage, zwei einstellige binäre Zahlen zu addieren. Möchte man mehr erreichen, so ist ein Volladdierer von Nöten. Dieser soll in der Lage sein, drei einstellige Binärzahlen zu addieren. Die dazugehörige Wahrheitstabelle ist Tabelle 3.2.

Die daraus resultierende disjunktive Normalform für den Übertrag U und die Gleichung für die Summe S lauten:

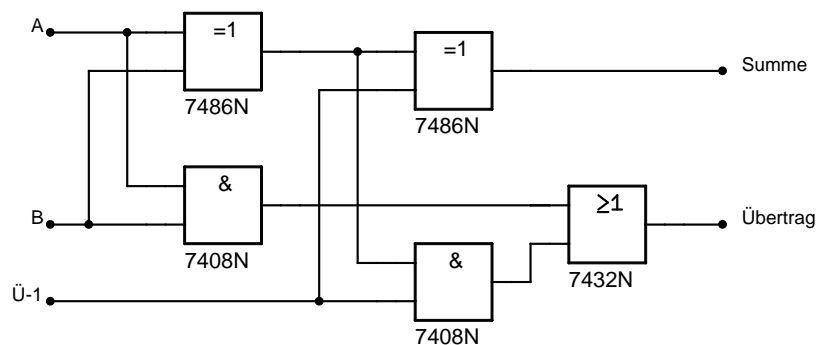
$$U = (\bar{A} \wedge B \wedge C) \vee (A \wedge \bar{B} \wedge C) \vee (A \wedge B \wedge \bar{C}) \vee (A \wedge B \wedge C) \quad (3.1)$$

$$S = A \vee B \vee C \quad (3.2)$$

**Tabelle 3.2:** Wahrheitstabelle Volladdierer

A	B	C	Summe	Übertrag
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

Die gesamte Schaltung lässt sich realisieren, indem man zwei Halbaddierer miteinander verbindet, sodass zuerst die Zahlen A und B addiert werden. Daraus resultieren dann ein Übertrag und eine Summe. Ein zweiter Halbaddierer addiert die Summe mit dem dritten Eingang C. Die daraus resultierende Summe ist die Summe der gesamten Rechnung. Nun müssen nur noch die zwei übrig gebliebenen Überträge miteinander verbunden werden. Ein Blick in Tabelle 3.2 zeigt, dass der Übertrag 1 ist, wenn der Übertrag aus A+B 1 ist, oder wenn der Übertrag aus  $((A+B)+C)$  1 ist. Daraus folgt, dass man die Überträge miteinander verodern kann. Das fertige Schaltnetz ist in Abbildung 3.2 zu sehen.

**Abbildung 3.2:** Schaltnetz Volladdierer mit einem dritten Eingang C = (Ü-1)

Mit einem Volladdierer kann man zwei Binärzahlen addieren, die aus mehr als einer Ziffer bestehen. Dafür muss man die Addition ziffernweise durchführen. Als dritte Zahl, die für die Addition eingegeben wird, wird der Übertrag der Addition der 2 niedrigerwertigen Ziffern eingegeben. Mithilfe so einer Kaskadierung von Volladdierern kann man mehrziffrige Binärzahlen addieren.

## 4. Aufgabe: Speicherelemente:

### 4.1 RS-Flip-Flop

Ein Flip-Flop ist ein Schaltnetz, welches in der Lage ist, Zustände zu speichern. Zuallererst wird in den Versuchen das RS-Flip-Flop untersucht. Sein Aussehen wird in Abbildung 4.1 dargestellt.

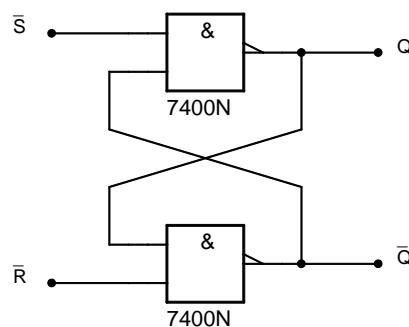
Das RS-Flip-Flop ist ungetaktet und besitzt zwei Eingänge: Set (S) und Reset(R). Der Name „RS-Flip-Flop“ ist eine Abkürzung für „Reset-Set-Flip-Flop“. Außerdem besitzt das RS-Flip-Flop zwei Ausgänge: Q und  $\bar{Q}$ . Dabei gilt:  $Q = \neg \bar{Q}$ . Bei der hier verwendeten Darstellung des RS-Flip-Flops sind die Eingänge  $\bar{R}$  und  $\bar{S}$  negiert.

Das Verhalten des RS-Flip-Flops in Tabelle 4.1 beschrieben.

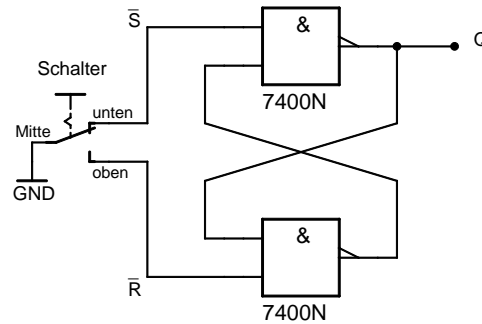
**Tabelle 4.1:** Zustandstabelle RS-Flip-Flop

R	S	$Q_n$	$\bar{Q}_n$	
0	0	$Q_{n-1}$	$\bar{Q}_{n-1}$	Keine Änderung (Speichern)
0	1	0	1	setze 0
1	0	1	0	setze 1
1	1	1	1	$Q_n = \bar{Q}_n$ verbotener Zustand

Durch anlegen einer 1 an den Eingang R und einer 0 an den Eingang S wird das RS-Flip-Flop zurückgesetzt und an dem Ausgang Q eine 0 ausgegeben. Geht man davon aus, dass davor  $Q = 1$  war und  $\bar{Q} = 0$  war, so schalten beide Nand-Gatter um, da am unteren Gatter jetzt eine 1 (vom Ausgang Q) und eine 0 ( $= \bar{R}$ ) anliegt, was nach der Wahrheitstabelle vom And-Gatter (Tabelle 1.3) einer 1 an  $\bar{Q}$  entspricht. Nun liegt an beiden Eingängen des oberen Nand-Gatters eine 1 an ( $\bar{Q}$  wird zu 1 und  $S = 0$ , also  $\bar{S} = 1$ ). Nach der Wahrheitstabelle



**Abbildung 4.1:** Schaltnetz eines RS-Flip-Flops



**Abbildung 4.2:** Schaltnetz eines Entprellers aus einem RS-Flip-Flop

des Nand-Gatters wird dann  $Q$  zu 0. Das ändert jetzt den Eingang des unteren Gatters wieder, jedoch sind an diesem nun beide Eingänge 0, was nach der Wahrheitstabelle des Nand-Gatters immer noch einem Ausgang von 1 entspricht. Nun kann auch  $T$  wieder auf 0 gesetzt werden; dann liegt am unteren Nand-Gatter eine 1 und eine 0 an, was immer noch die Ausgabe  $\bar{Q} = 1$  erzeugt. Wenn die Eingänge mit  $R = S = 0$  beschaltet werden, so behält das RS-Flip-Flop seinen Zustand bei  $Q$  und  $\bar{Q}$  bei, woraus auch folgt, dass wenn initial  $Q = 0$  und  $\bar{Q} = 1$  sind, sich nichts verändert.

Umgekehrt kann durch anlegen einer 1 an dem Eingang  $S$  eine 1 an dem Ausgang  $Q$  gesetzt werden (entspricht dem Setzen von  $Q$ ). Der Ausgang  $\bar{Q}$  passt sich dementsprechend an. Wegen der Symmetrie des Aufbaus ist die Begründung für das Setzen analog zu der Begründung für das Zurücksetzen; es reicht, oben und unten,  $S$  und  $R$ , und  $Q$  und  $\bar{Q}$  zu vertauschen.

Dies gilt aber nur für den allgemeinen Fall  $R = \bar{S}$ . Ein Problem ergibt sich aber, wenn die Eingänge mit  $R = S = 1$  belegt werden, so gibt das RS-Flip-Flop die Signale  $Q = \bar{Q} = 1$  aus, weil an jedem Nand-Gatter ein Eingang 0 ist, was wieder nach der Wahrheitstabelle des Nand-Gatters für die Ausgabe 1 sorgt. Dies wird als „verbotener Zustand“ bezeichnet, weil dann  $Q = \bar{Q}$  vorliegt. Aus diesem Grund muss der verbotene Zustand beim RS-Flip-Flop vermieden werden.

## 4.2 Entprellen eines Schalters

Mit einem modifizierten RS-Flip-Flop kann man das Signal eines Schalters entprellen. Unter dem Prellen eines Schalters versteht man das Rauschen, das während dem Umlegen des Schalters und kurz danach am Ausgangssignal des Schalters anliegt. Häufig wird dies mechanisch bedingt hervorgerufen, weil der Schalter beim Schließen und somit Annehmen eines neuen Zustandes mechanisch schwingt und daher im Inneren die elektrischen Kontakte nicht kontinuierlich berührt.

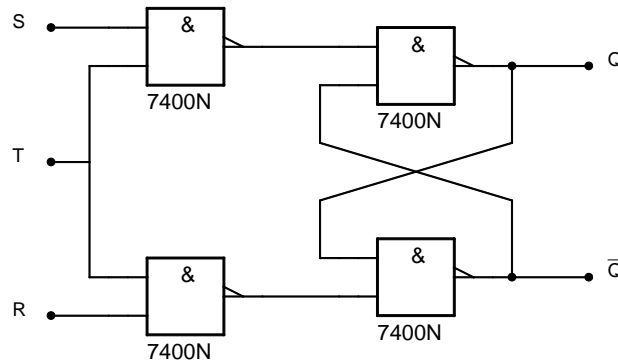
Wenn man ein RS-Flip-Flop an einen Schalter anschließt und dafür sorgt, dass der Schalter immer nur einen der Eingänge des RS-Flip-Flops auf LOW legt, so wird der andere Eingang durch den internen Pull-Up Widerstand immer auf HIGH gelegt. Abbildung 4.2 zeigt das Schaltnetz dieses Entprellers.

Der Entpreller funktioniert dadurch, dass die Ausgangssignale nach dem ersten Kontakt des Schalters mit dem neuen Zustand stehen bleiben. Die darauf folgenden durch Schwingungen bedingten Prellungen des Schalters bewirken keine Zustandsänderung und haben somit keinen Einfluss auf die Ausgänge der Flip-Flops. Das Verfahren wird noch einmal in Tabelle 4.2 beschrieben.



**Tabelle 4.2:** Zustandstabelle RS-Flip-Flop als Entpreller

Schalterstellung	Q
Oben	1
Unten	0

**Abbildung 4.3:** Schaltnetz eines RST-Flip-Flops

### 4.3 Getaktetes RS-Flip-Flop

Das getaktete RS-Flip-Flop, auch RST-Flip-Flop, ist ein taktgesteuerter Umbau des RS-Flip-Flops. Zusätzlich zu den Eingängen R und S besitzt das RST-Flip-Flop einen weiteren Eingang T, der den Takt vorgibt. Der Aufbau des RST-Flip-Flops wird in Abbildung 4.3 gezeigt und das Verhalten in Tabelle 4.3 erklärt.

**Tabelle 4.3:** Zustandstabelle RST-Flip-Flop

T	R	S	$Q_{n+1}$	$\overline{Q}_{n+1}$	
0	0	0	$Q_{n-1}$	$\overline{Q}_{n-1}$	kein Takt, keine Änderung
0	0	1	$Q_{n-1}$	$\overline{Q}_{n-1}$	kein Takt, keine Änderung
0	1	0	$Q_{n-1}$	$\overline{Q}_{n-1}$	kein Takt, keine Änderung
0	1	1	$Q_{n-1}$	$\overline{Q}_{n-1}$	kein Takt, keine Änderung
1	0	0	$Q_{n-1}$	$\overline{Q}_{n-1}$	keine Änderung, Speichern
1	0	1	0	1	setze 0
1	1	0	1	0	setze 1
1	1	1	1	1	$Q_n = \overline{Q_n}$ verbotener Zustand

Wenn ein Eingangssignal für den Takt mit  $T=0$  eingegeben wird, ist das RST-Flip-Flop nicht in der Lage, seinen Zustand zu ändern. Bei einem HIGH-Signal für den Takt mit  $T=1$  verhält sich das RST-Flip-Flop identisch zu dem RS-Flip-Flop. Das RST-Flip-Flop ist also ein taktzustandsgesteuertes Flip-Flop, welches nur bei einer eingeschalteten Takt-Flanke agiert.

Man kann beim RST-Flip-Flop den verbotenen Zustand umgehen, indem man über den Einsatz eines Not-Gatter die Situation  $S = \overline{R}$  dauerhaft erzwingt. Dann muss das Register folgendermaßen bedient werden: Um den Zustand beizubehalten, wird der Takt auf  $T=0$  gesetzt. In dieser Zeit kann ein neuer Zustand S und  $R = \overline{S}$  eingegeben werden. Dieser wird aber erst übernommen, wenn der Takt auf  $T=1$  gesetzt wird. Wird der Takt danach

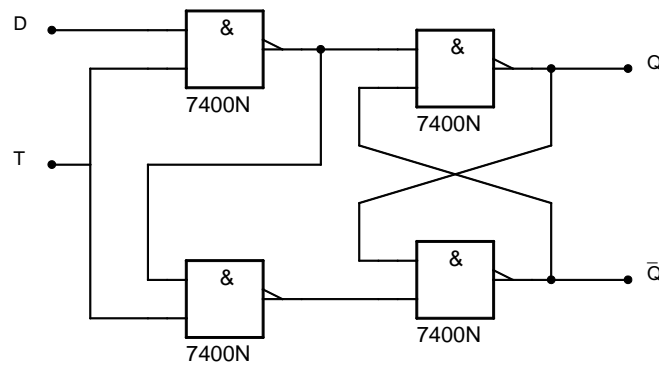


Abbildung 4.4: Schaltnetz eines D-Flip-Flops

wieder auf  $T=0$  gesetzt, behält das RST-Flip-Flop seine Ausgangszustände bei und es können neue Zustände eingegeben werden, die später bei einer HIGH-Flanke des Taktsignals wieder übernommen werden. In der Zeit, in der an den Eingängen  $T=0$  vorliegt, haben die Eingänge  $R$  und  $S = \bar{R}$  auf die Ausgänge des RST-Flip-Flops keine Auswirkung.

#### 4.4 D-Flip-Flop

Das D-Flip-Flop („Data-Flip-Flop“) ist eine Variation des RST-Flip-Flops. Im vorherigen Abschnitt wurde beschrieben, wie man das RST-Flip-Flop modifizieren kann, sodass der verbotene Zustand nicht mehr eintritt. Das D-Flip-Flop erreicht dies auf eine andere Art und Weise. Anstatt dass kontinuierlich das Signal  $R = \bar{S}$  angelegt wird, ist dies nur der Fall, wenn  $T=1$  anliegt. Abbildung 4.4 zeigt das Schaltnetz eines D-Flip-Flops.

Das D-Flip-Flop gibt es sowohl als taktzustands- als auch als taktflankengesteuertes Flip-Flop. Bei dem im Versuch und in Abbildung 4.4 verwendeten bzw. abgebildeten Flip-Flop handelt es sich um ein taktzustandsgesteuertes Flip-Flop. Das Verhalten des Flip-Flops wird in Tabelle 4.4 beschrieben.

Das taktflankengesteuerte D-Flip-Flop würde im Gegensatz zum taktzustandsgesteuerten D-Flip-Flop die neuen Zustände nicht übernehmen, wenn am Taktsignal eine logische 1 anliegt, sondern, wenn das Taktsignal von einer logischen 0 zu einer logischen 1 wechselt (oder umgekehrt, je nach Ausführung).

Tabelle 4.4: Zustandstabelle D-Flip-Flop

T	D	$Q_{n+1}$	$\bar{Q}_{n+1}$	
0	0	$Q_{n-1}$	$\bar{Q}_{n-1}$	kein Takt, keine Änderung
0	1	$Q_{n-1}$	$\bar{Q}_{n-1}$	kein Takt, keine Änderung
1	0	0	1	setze 0
1	1	1	0	setze 1

Das in Tabelle 4.4 beschriebene D-Flip-Flop verhält sich wie das im vorherigen Abschnitt beschriebene modifizierte RST-Flip-Flop ohne den verbotenen Zustand.

#### 4.5 JK-Master-Slave-Flip-Flop

Das JK-MS-Flip-Flop, „Jump-Kill-Master-Slave-Flip-Flop“ ist ein taktflankengesteuertes Flip-Flop. Das Schaltnetz eines JK-MS-Flip-Flops wird in Abbildung 4.5 dargestellt.

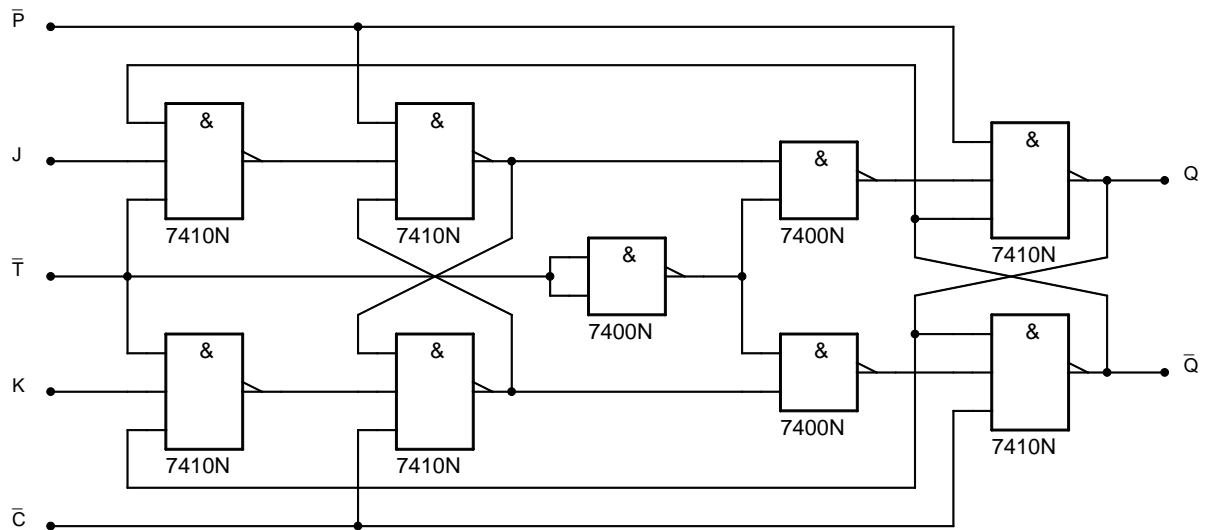


Abbildung 4.5: Schaltnetz eines JK-MS-Flip-Flops

Ein JK-MS-Flip-Flop besteht aus zwei RST-Flip-Flops: einem Master und einem Slave. Der Master wird mit dem Taktsignal getrieben, während der Slave mit dem negierten Taktsignal getrieben wird. Diese Kombination sorgt dafür, dass das JK-MS-Flip-Flop taktflankengesteuert ist, weil die einzelnen RST-Flip-Flops bei entgegengesetzter Taktflanke aktiviert werden. Die zwei einzelnen Flip-Flops sind taktzustandsgesteuert und werden im JK-MS-Flip-Flop abwechselnd aktiviert bzw. beschrieben. Wäre die Negierung nicht vorhanden, so wäre das Flip-Flop Taktzustandsgesteuert. Die zwei RST-Flip-Flops sind so miteinander verbunden, dass die Ausgänge  $Q$  und  $\bar{Q}$  des Masters mit den Eingängen  $R$  und  $S$  des Slaves verbunden sind. Beim JK-MS-Flip-Flop heißen diese Eingänge  $J$  und  $K$ .

Eine weitere Besonderheit am JK-MS-Flip-Flop liegt darin, dass zwei weitere Eingänge zur Verfügung stehen:  $\bar{P}$  und  $\bar{C}$ . Die Eingänge  $\bar{P}$ reset und  $\bar{C}$ lear ermöglichen es, die Ausgänge unabhängig vom Takt zu steuern.

Die Funktionsweise eines JK-MS-Flip-Flops lässt sich von der eines RST-Flip-Flops ableiten. Zuerst kann bei einer anliegenden Taktflanke von  $\bar{T} = 1$  in das Master Flip-Flop ein Zustand über die Eingänge  $J$  und  $K$  geschrieben werden. In dieser Zeit ist der Slave gesperrt. Der Master-Flip-Flop verhält sich ähnlich wie ein RST-Flip-Flop, dessen Zustandstabelle in 4.3 beschrieben wurde. Eine Besonderheit gibt es aber: es gibt keinen verbotenen Zustand. Durch die überkreuzte Rückkopplung des Slave-Flip-Flops ist es auch möglich, die Signale  $J=K=1$  einzugeben. Dies wird im nächsten Absatz genauer erklärt. Die Ausgänge aus dem Master Flip-Flop sind direkt mit den Eingängen vom Slave-Flip-Flop verbunden. Bei einer fallenden Taktflanke von  $\bar{T}$  wird nun das Master-Flip-Flop gesperrt, während das Slave-Flip-Flop die Ausgabe des Masters als Eingang übernimmt. Dies führt zu der in Tabelle 4.5 dargestellten Zustandstabelle.

Der Spezialfall „toggeln“ tritt ein, wenn die Eingänge mit  $J=K=1$  belegt werden. Durch die überkreuzte Rückkopplung aus dem Slave-Flip-Flop führt diese Ausgabe dazu, dass das Flip-Flop mit jeder fallenden Taktflanke von  $\bar{T}$  das Inverse des vorherigen Zustandes annimmt.

**Tabelle 4.5:** Zustandstabelle JK-MS-Flip-Flop

J	K	Taktflanke T	$Q_{n+1}$	$\overline{Q_{n+1}}$	
0	0	+	$Q_n$	$\overline{Q_n}$	Keine Änderung (Speichern)
0	0	-	$Q_n$	$\overline{Q_n}$	falsche Flanke $\Rightarrow$ keine Änderung
0	1	+	0	1	setze 0
0	1	-	$Q_n$	$\overline{Q_n}$	falsche Flanke $\Rightarrow$ keine Änderung
1	0	+	1	0	setze 1
1	0	-	$Q_n$	$\overline{Q_n}$	falsche Flanke $\Rightarrow$ keine Änderung
1	1	+	$\overline{Q_n}$	$Q_n$	toggeln
1	1	-	$Q_n$	$\overline{Q_n}$	falsche Flanke $\Rightarrow$ keine Änderung

## 5. Aufgabe: Schieben, Multiplizieren, Rotieren

### 5.1 4-Bit-Schieberegister

Eine wichtige Schaltaufgabe erfüllen ebenfalls Schieberegister. Sie ermöglichen es, serielle Signale (die auf einer Datenleitung übertragen werden) parallel (auf mehreren Datenleitungen) auszugeben. Das ist nützlich, um beispielsweise über einen zur Verfügung stehenden Pin mehrere Geräte (z.B. LEDs) anzusteuern. Ebenfalls kann man Schieberegister zum rechnen verwenden; ein Shift weiter entspricht einer Multiplikation mit 2, ein Shift zurück entspricht einer Division durch 2. In Abbildung 5.1 ist das Schaltnetz eines 4-Bit-Schieberegisters zu sehen. Es lässt sich erkennen, dass das Register grundsätzlich aus 4 JK-MS-Flip-Flops aufgebaut ist. Die Clear-Signale C (bzw. hier  $\bar{C}$ ) sind zusammenverbunden, um das gesamte Register auf einmal leeren zu können. Auch der Timer T (bzw.  $\bar{T}$ ) ist global für alle JK-MS-FFs verbunden.

Am Eingang D werden serielle Daten entgegengenommen. Dadurch, dass Q mit J des nächsten JK-MS-FFs verbunden ist, und  $\bar{Q}$  mit K verbunden ist, werden die Daten bei einem Taktpuls an das jeweils nächste Flip-Flop weitergegeben. Da am Anfangssignal noch kein invertiertes Signal zur Verfügung steht, wird an dieser Stelle ein Inverter benötigt, der  $\bar{D}$  bereitstellt. An den Ausgängen Q der Flip-Flops lassen sich die Daten parallel abgreifen. So kann über Aneinanderhängen von  $n$  JK-MS-FFs ein  $n$ -Bit-Schieberegister gebaut werden.

Bei diesem Versuch ist es auch wichtig, einen entprellten Taktgeber zu verwenden. Ist dieser nicht entprellt, kann es sein, dass statt einem Puls mehrere registriert werden, da die JK-MS-FFs bei der steigenden Taktflanke (bzw. hier bei der fallenden Taktflanke, da  $\bar{T}$  weitergegeben wird), schalten. So können nicht entprellte Taktgeber ungewollte Shifts hervorrufen. In diesem Fall wurde ein umgebautes RS-FF, welches einen einzigen Schalter als einzige Eingabe hatte, als entprellter Taktgeber verwendet.

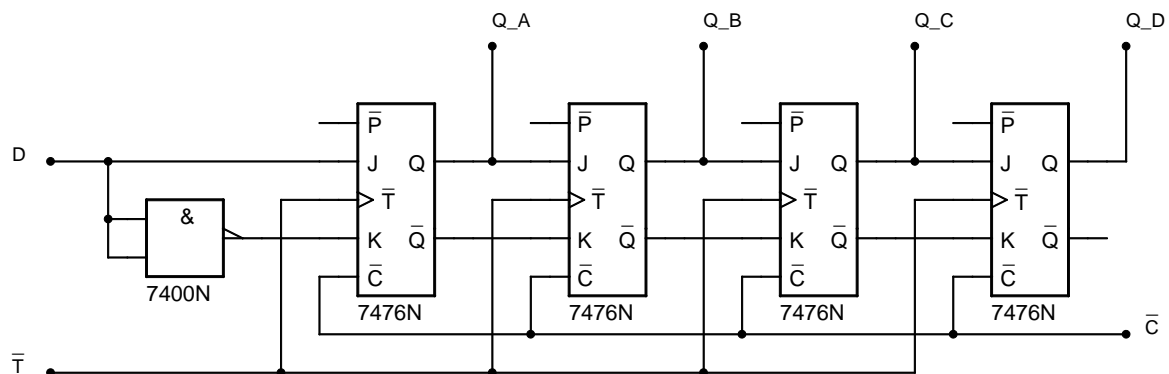


Abbildung 5.1: Schaltnetz eines 4-Bit-Schieberegisters

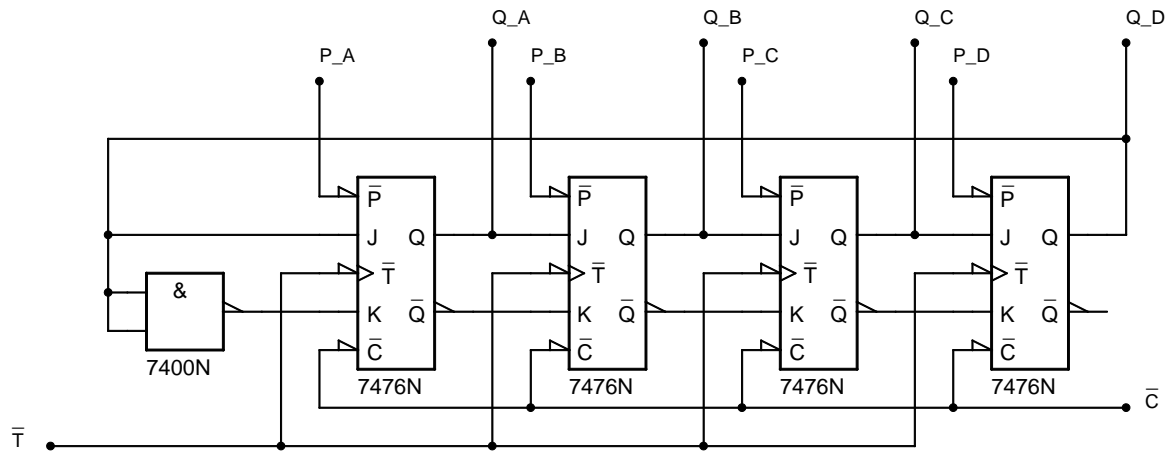


Abbildung 5.2: Schaltnetz eines 4-Bit-Rotationsregisters

## 5.2 4-Bit-Rotationsregister

Das 4-Bit-Rotationsregister ist grundsätzlich sehr ähnlich zum 4-Bit-Schieberegister. Wie der Name bereits impliziert, werden hier immer dieselben Daten 'im Kreis' rotiert. Das wird dadurch erreicht, dass der Ausgang des letzten Flip-Flops wieder mit dem Eingang des Ersten verbunden wird. In der Abbildung 5.2 ist zu sehen, wie ein 4-Bit-Rotationsregister aufgebaut ist. Da eine serielle Eingabe über das Daten-Signal D nicht mehr (einfach ohne Modifikation) möglich ist, werdend die Preset-Pins P der JK-MS-FFs verwendet. Über diese lassen sich die gewünschten Daten, die rotiert werden sollen, einstellen. Hier wurde das so realisiert, dass die P-Linien verkabelt, aber nirgends angeschlossen wurden; bei Notwendigkeit lassen sich diese mit HIGH oder LOW verbinden.

Der Inverter, der beim Schieberegister benötigt wird, um aus dem Eingang D das invertierte Signal  $\bar{D}$  zu machen, ist hier auch theoretisch nicht mehr notwendig: Da das letzte JK-MS-FF sowohl Q als auch  $\bar{Q}$  ausgibt, könnte man einfach den offenen Ausgang  $\bar{Q}$  mit dem Eingang K des ersten FFs verbinden.

## 6. Aufgabe: Zähler

### 6.1 4-Bit-Asynchronzähler

Eine weitere wichtige technische Möglichkeit ist es, eine Anzahl an Pulsen zu zählen. So kann beispielsweise die Anzahl an Pulsen eines Encoders bestimmt werden, oder allgemein bestimmt werden, wie oft ein beliebiger digitaler Prozess in einem gewissen Zeitfenster stattgefunden hat. In der Abbildung 6.1 ist ein 4-Bit-Asynchronzähler dargestellt. Wie ein Schieberegister besteht er grundsätzlich aus  $n = 4$  JK-MS-Flip-Flops mit der Bit-Anzahl  $n$ , die hier jedoch anders verschaltet werden. Hier gibt es kein klassisches Datensignal D mehr, sondern nur noch das Timer-Signal T (bzw.  $\bar{T}$ ), dessen Pulse gezählt werden. Das wird durch die Verschaltung der JK-MS-FFs realisiert. Da das K-Signal nicht verbunden ist, sorgen die Pull-Up-Widerstände in den ICs dafür, dass die Flip-Flops im Toggle-Modus arbeiten. Das bedeutet, dass die Flip-Flops abwechselnd an-und ausschalten. Da der Ausgang des einen Flip-Flops mit dem Timer-Eingang  $\bar{T}$  des jeweils nächsten FFs verbunden wird, und diese nur bei Fallendem  $\bar{T}$ -Signal schalten, schaltet jedes Flip-Flop mit der halben Frequenz seines Vorgängers und mit der doppelten Frequenz seines Nachfolgers; die Flip-Flops laufen alle asynchron.

Wird der Asynchronzähler mit den  $\bar{Q}$ -Ausgängen anstelle von den Q-Ausgängen angesteuert, so zählt er rückwärts. Dadurch werden die Takteingänge jedes Flip-Flops von dem jeweils negierten des Ausgangsignals des vorangestellten Flip-Flops geschaltet.

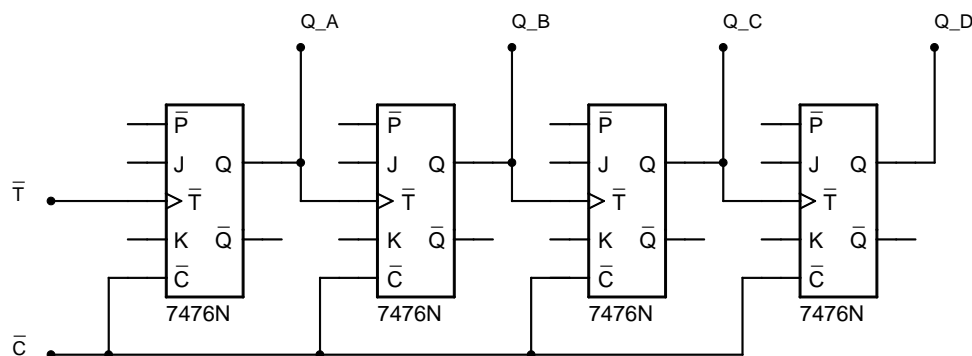


Abbildung 6.1: Schaltnetz eines 4-Bit-Asynchronzählers

## 7. Durchführung

Alle in diesem Protokoll beschriebenen Schaltungen wurden beim Durchführen der Experimente auf Anhieb erfolgreich nach dem jeweiligen Schaubild aufgebaut und haben erwartungsgemäß funktioniert.

Fast alle Schaltungen wurden ohne weitere Vorbereitungen aufgebaut, da diese relativ kompakt und übersichtlich waren, was insbesondere daran liegt, dass bei den späteren Aufgaben die Komplexität zunehmend in die ICs verlagert wurde. Lediglich beim Aufbau des JK-MS-FFs wurde auf einem Zettel mit dem Schaubild über die verbauten Kabel Buch geführt, da dieser Aufbau mit Abstand die größte Komplexität bei der Verschaltung hatte und deshalb schnell unübersichtlich geworden ist.