

Praktikum Klassische Physik I

Versuchsvorbereitung: **P1-63, 64, 65: Schaltlogik**

Christian Buntin

Gruppe Mo-11

Karlsruhe, 26. Oktober 2009

Ausgehend von einfachen Logikgattern wird die Funktionsweise von Addierern, Speicherelementen, Multiplikatoren, Zähler und Digital-Analog-Wandlern beschrieben, welche die Grundlage der heutigen Elektronik bilden.

Bei diesen Logikschaltungen treten nur zwei Zustände auf: Entweder 1 (auch: *HIGH*, Wahr, +5 V) oder 0 (auch: *LOW*, Falsch, 0 V). Aus diesem Grund sollten die verwendeten Widerstände entsprechend hohe bzw. niedrige Werte haben, damit sich die auftretenden Spannungen eindeutig als *HIGH* oder *LOW* auffassen lassen.

Inhaltsverzeichnis

1	Diodengatter	3
1.1	AND-Gatter	3
1.2	NOT-Gatter	3
1.3	OR-Gatter	3
2	Gatter mit ICs	4
2.1	Inverter aus NAND- oder NOR-Gatter	5
2.2	XOR-Gatter	5
2.3	XOR-Gatter aus NAND-Gattern	7
3	Addierer	8
3.1	Halbaddierer	8
3.2	Volladdierer	8
3.3	Subtrahierer	9
4	Flip-Flops	11
4.1	RS-Flip-Flop	11
4.2	RST-Flip-Flop	12
4.3	JK-Master-Slave-Flip-Flop	14
5	Schieben, Multiplizieren, Rotieren	15
5.1	4-Bit-Schieberegister	15
5.2	4-Bit-Rotationsregister	16
6	Zähler	16
6.1	4-Bit-Asynchrone Zähler	16
6.2	Asynchrone Dezimalzähler	17
6.3	4-Bit-Synchrone Zähler	18
6.4	Synchrone Dezimalzähler	18
7	Digital-Analog-Wandlung	18

1 Diodengatter

1.1 AND-Gatter

Ein AND-Gatter gibt genau dann 1 aus, wenn beide Eingänge auf 1 gelegt sind (siehe Tabelle 1).

Es wird ein Dioden-AND-Gatter wie in der Vorbereitungsmappe beschrieben aufgebaut (siehe Abbildung 1).

Wenn A und/oder B auf 0 liegen, so fließt der Strom über die Dioden ab und am Widerstand R_1 fällt der Großteil der Spannung ab, sodass C auf 0 liegt. Wenn einer der Eingänge auf 1 und der andere auf 0 liegt, dann verhindern die Dioden einen Stromfluss zwischen den Eingängen. Dennoch fließt Strom über einen Eingang ab und C liegt auf 0. Wenn jedoch A und B zusammen auf 1 liegen, so liegen alle Eingänge und somit auch C auf 1.

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

Tabelle 1: Wahrheitstabelle eines AND-Gatters

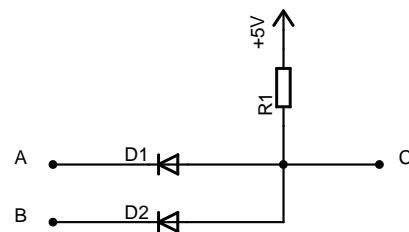


Abbildung 1: Schaltskizze eines Dioden-AND-Gatters

1.2 NOT-Gatter

Ein NOT-Gatter invertiert den Eingang (siehe Tabelle 2).

Es wird ein NOT-Gatter wie in der Vorbereitungsmappe beschrieben aufgebaut (siehe Abbildung 2).

Wenn A auf 0 liegt, so sperrt der Transistor und B liegt somit auf 1. Wenn A auf 1 liegt, schaltet der Transistor durch und an R_3 fällt fast die ganze Spannung ab, der Strom fließt quasi an B „vorbei“, weshalb B auf 0 liegt.

NAND-Gatter

Ein NAND-Gatter gibt die invertierte Ausgabe des AND-Gatters aus (siehe Tabelle 3).

Es wird das AND-Gatter mit dem NOT-Gatter verbunden (siehe Abbildung 3).

1.3 OR-Gatter

Ein OR-Gatter gibt genau dann 1 aus, wenn einer der Eingänge (oder beide) auf 1 gelegt sind (siehe Tabelle 4).

A	$\bar{A} = C$
0	1
1	0

Tabelle 2: Wahrheitstabelle eines NOT-Gatters

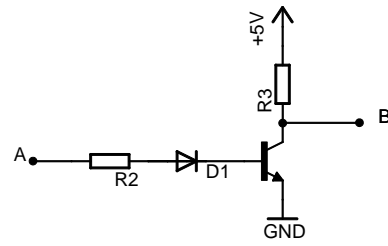


Abbildung 2: Schaltskizze eines NOT-Gatters

A	B	$A \wedge B$	$\overline{A \wedge B} = C$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Tabelle 3: Wahrheitstabelle eines NAND-Gatters

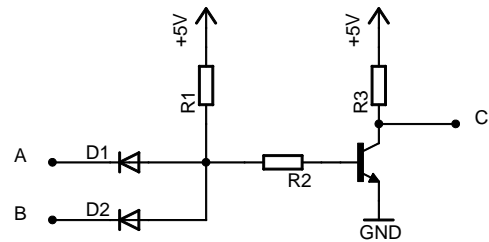


Abbildung 3: Schaltskizze eines NAND-Gatters

Es wird ein OR-Gatter wie in der Vorbereitungsmappe beschrieben aufgebaut (siehe Abbildung 4).

Wenn A und B auf 0 liegen, liegt in der Schaltung nur der *LOW*-Zustand vor, also liegt auch C auf 0. Liegt A oder B (oder beide) auf 1, so liegt damit auch C auf 1, da an R1 fast die gesamte Spannung abfällt. Wie auch schon beim AND-Gatter verhindern hier die Dioden einen Stromfluss zwischen den Eingängen.

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

Tabelle 4: Wahrheitstabelle eines OR-Gatters

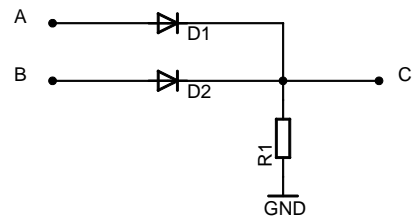


Abbildung 4: Schaltskizze eines Dioden-OR-Gatters

2 Gatter mit ICs

Da die im letzten Kapitel verwendeten Dioden auch in Durchlassrichtung immer noch einen Widerstand besitzen, fällt dort immer etwas Spannung ab. Dadurch erhöht sich beim AND-Gatter die *LOW*-Schwelle und beim OR-Gatter verringert sich die *HIGH*-Schwelle ein wenig. Wenn man dann mehrere dieser Gatter hintereinander schaltet, kann der Spannungsabfall so hoch werden, dass der Zustand des Ausgangs, *HIGH* oder *LOW*, nicht mehr eindeutig ist. Daher werden im Folgenden ICs statt Dioden verwendet, in denen die Gatter bereits samt Schaltung zur Vermeidung des Spannungsabfalls integriert sind. Bei den verwendeten ICs muss darauf geachtet werden, dass freie Eingänge auf 1 gelegt sind.

2.1 Inverter aus NAND- oder NOR-Gatter

A	B	$A \vee B$	$\overline{A \vee B} = C$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Tabelle 5: Wahrheitstabelle eines NOR-Gatters

Zunächst werden die Wahrheitstabellen des NAND- und des NOR-Gatters untersucht.

Wenn man beim NAND-Gatter (siehe Tabelle 3) nur A und C vergleicht, so ergibt sich für die erste, zweite und vierte Zeile die Wahrheitstabelle eines NOT-Gatters. Dabei muss für die zweite und vierte Zeile B auf 1 gelegt werden, für die erste und vierte Zeile müssen A und B zusammenschaltet werden.

Beim NOR-Gatter ergibt sich dieses Verhalten für A und C für die erste, dritte und vierte Zeile der Wahrheitstabelle (siehe Tabelle 5). Dabei muss für die erste und dritte Zeile B auf 0 gelegt werden. Für die erste und vierte Zeile müssen A und B zusammenschaltet werden.

Somit ergeben sich vier verschiedene Möglichkeiten, ein NOT-Gatter auf diese Weise zu konstruieren.

2.2 XOR-Gatter

Um die disjunktive Normalform zu erhalten, wird aus der Wahrheitstabelle abgelesen, für welche Bedingungen C wahr ist. Dies ist bei A und nicht B oder bei B und nicht A der Fall:

$$(A \wedge \bar{B}) \vee (\bar{A} \wedge B) \quad (1)$$

Dies ist mit der Schaltung aus Abbildung 6 realisiert, wobei die NAND-Gatter als Inverter fungieren.

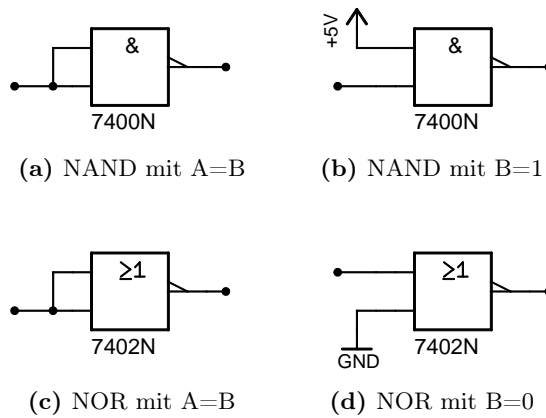


Abbildung 5: Verschiedene NOT-Schaltungen

A	B	$A \underline{\vee} B = C$
0	0	0
0	1	1
1	0	1
1	1	0

Tabelle 6: Wahrheitstabelle eines XOR-Gatters

2.3 XOR-Gatter aus NAND-Gattern

Die Gleichung 1 wird umgeformt, um zu einen einfacheren Aufbau zu gelangen:

$$(A \wedge \bar{B}) \vee (\bar{A} \wedge B) \quad (2)$$

$$= (A \wedge \bar{B}) \vee (\bar{A} \wedge B) \vee (A \wedge \bar{A}) \vee (B \wedge \bar{B}) \quad (3)$$

$$= (A \wedge (\bar{A} \vee \bar{B})) \vee (B \wedge (\bar{A} \vee \bar{B})) \quad (4)$$

$$= \left(A \wedge \overline{(\bar{A} \vee \bar{B})} \right) \vee \left(B \wedge \overline{(\bar{A} \vee \bar{B})} \right) \quad (5)$$

$$= \left(A \wedge \overline{(A \wedge B)} \right) \vee \left(B \wedge \overline{(A \wedge B)} \right) \quad (6)$$

$$= \overline{\overline{(A \wedge \overline{(A \wedge B)}) \vee (B \wedge \overline{(A \wedge B)})}} \quad (7)$$

$$= \overline{\overline{(A \wedge \overline{(A \wedge B)})} \wedge \overline{\overline{(B \wedge \overline{(A \wedge B)})}}} \quad (8)$$

$$= \overline{\overline{A \bar{A} B} \quad \overline{\overline{B \bar{A} B}}} \quad (9)$$

Von Gleichung 2 nach 3 wird mit dem Komplement $A \wedge \bar{A} = 0$ und $B \wedge \bar{B} = 0$ so erweitert, dass die Identität $x \vee 0 = x$ vorliegt. In Gleichung 4 wurde das Distributivgesetz angewendet und A und B ausgeklammert. In Gleichung 5 wurden die inneren Klammern zweimal negiert, was auch wieder der Identität entspricht.

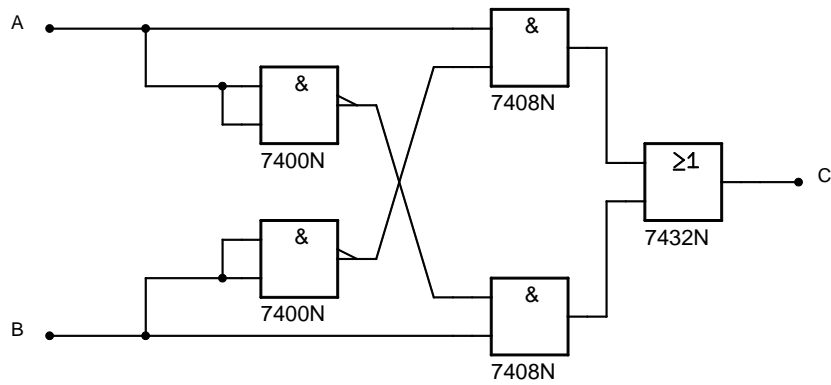


Abbildung 6: Schaltskizze eines XOR-Gatters aus der disjunktiven Normalform

Auf diese Klammern wird in Gleichung 6 das DeMorgan'sche Gesetz angewendet: $\overline{x \vee y} = \overline{x} \wedge \overline{y}$. Nun wird die gesamte Gleichung zweimal negiert (Gleichung 7) und wieder das DeMorgan'sche Gesetz angewendet (Gleichung 8). Dies entspricht der geforderten Form in Gleichung 9.

Mithilfe dieses Ergebnisses lässt sich ein XOR-Gatter somit auch aus vier NAND-Gatter aufbauen (Abbildung 7). Da nur noch ein Gattertyp verwendet wird, wird der Aufwand geringer, da ein IC bereits 4 NAND-Gatter beinhaltet. Dies spart den Aufwand, den man mit verschiedenen Bauteilen hätte, weshalb so vor allem bei großen Schaltungen die Kosten gesenkt werden können.

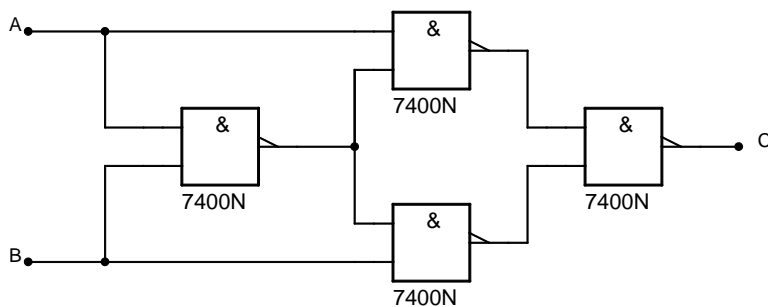


Abbildung 7: Schaltskizze eines XOR-Gatters aus NAND-Gattern

3 Addierer

3.1 Halbaddierer

Ein Halbaddierer addiert zwei Binärzahlen und gibt das Ergebnis als Summe und Übertrag aus (Tabelle 7). Für die disjunktive Normalform ergibt sich für S: Exklusiv

A	B	\ddot{U}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Tabelle 7: Wahrheitstabelle eines Halbaddierers

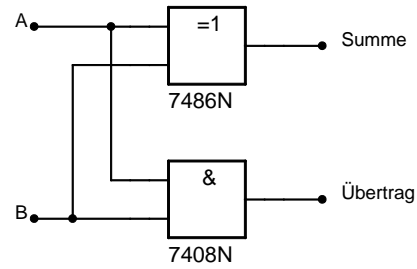


Abbildung 8: Schaltskizze eines Halbaddierers

A oder B. Für \ddot{U} : A und B.

$$S = A \vee B \qquad \ddot{U} = A \wedge B$$

Dies lässt sich also mit einem AND und einem XOR-Gatter wie in Abbildung 8 bewerkstelligen.

3.2 Volladdierer

A	B	\ddot{U}_{-1}	\ddot{U}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
1	0	0	0	1
0	1	1	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Tabelle 8: Wahrheitstabelle eines Volladdierers

Ein Volladdierer besteht aus zwei hintereinander geschalteten Halbaddierern. Zur Summe von A und B des ersten Addierers wird der Übertrag \ddot{U}_{-1} eines eventuell vorhergehenden Addierers addiert und diese Summe an S ausgegeben. Wenn vom ersten oder vom zweiten Addierer ein Übertrag kommt, wird dieser an \ddot{U} ausgegeben. Somit lassen sich drei 1-Bit-Zahlen addieren (Tabelle 8).

Der Halbaddierer aus Kapitel 3.1 wird mit einem weiteren Halbaddierer zu einem Volladdierer wie in Abbildung 9 vervollständigt. Dabei kann, um Bauteile zu sparen, statt dem OR-Gatter auch ein XOR-Gatter benutzt werden, da die beiden Ausgänge des ersten Halbaddierers niemals gleichzeitig auf 1 liegen werden (Siehe Tabelle 7).

Um größere Zahlen addieren zu können, werden mehrere solche Volladdierer miteinander verbunden. Dabei wird der Übertrag mit dem \ddot{U}_{-1} -Eingang des nächsten

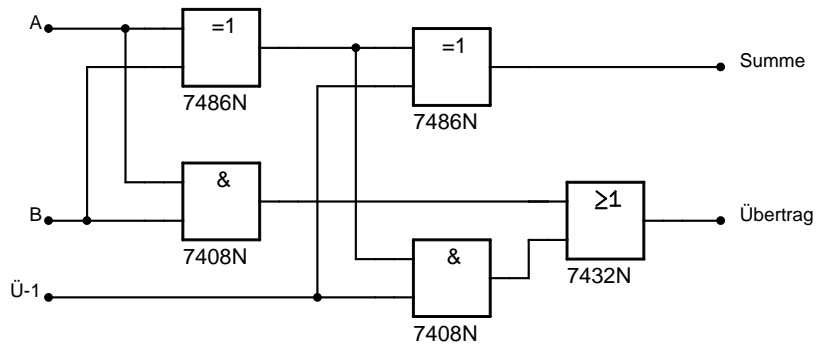


Abbildung 9: Schaltskizze eines Volladdierern aus zwei Halbaddierern

Addierers verbunden (Siehe Abbildung 10). Über die Eingänge werden die Stellen der Binärzahlen eingegeben, an den Summe-Ausgängen und am Übertrag wird die Summe der Zahlen ausgegeben.

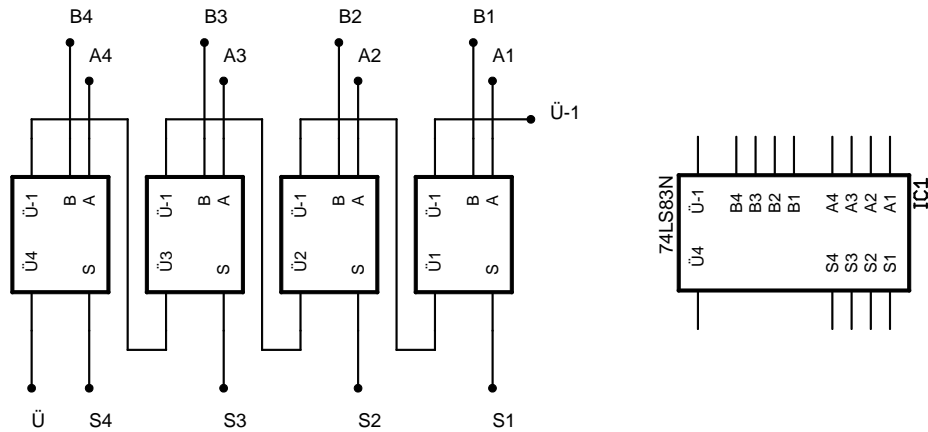


Abbildung 10: Ein 4 Bit Volladdierer

3.3 Subtrahierer

Es wird ein Subtrahierer wie in der Vorbereitungsmappe beschrieben aufgebaut (siehe Abbildung 11).

Bei diesem Subtrahierer werden alle Bits des Subtrahenten Negiert, es wird die Komplementärzahl erzeugt: $\bar{A} = 15 - A$. Der Addierer in der Mitte rechnet also:

$$S = B + \bar{A} = B - A + 15$$

Zusätzlich wird das Übertragsbit hinzuaddiert. Bei dieser Rechnung müssen drei Fälle unterschieden werden:

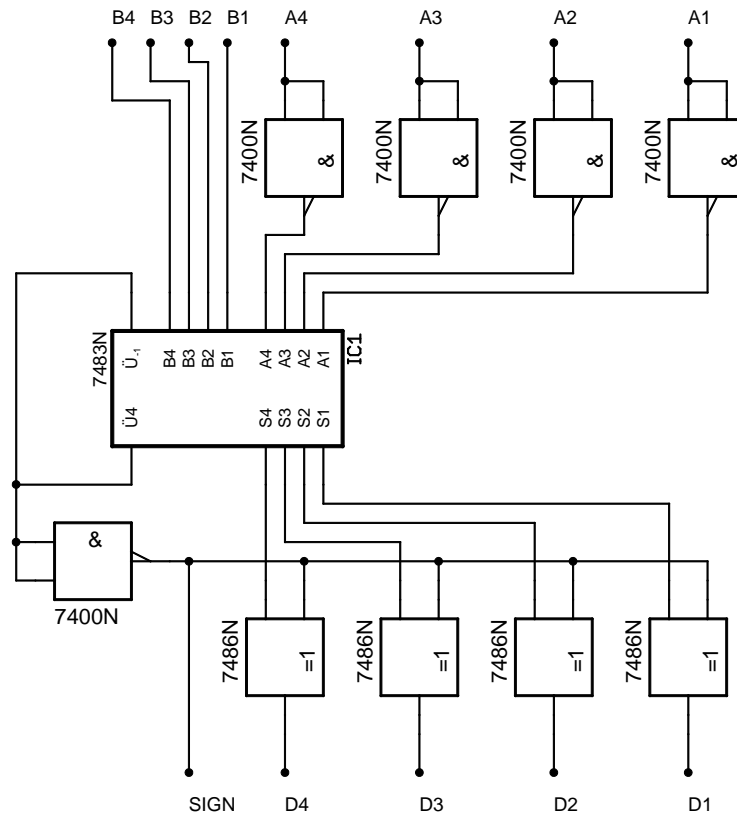


Abbildung 11: Schaltskizze eines Subtrahierers

- i) $A < B$: S wird größer als 15, also wird das Übertragsbit gesetzt, welches somit noch hinzuaddiert wird, damit mit modulo 16 bestimmt werden kann, um wieviel S größer als 16 ist. Da nur die ersten vier Bits ausgegeben werden, wird der Wert $D = S \bmod 16$ ausgegeben, was eben dieser Differenz entspricht.

Das Übertragsbit entspricht dem Vorzeichen der Differenz, wobei *HIGH* für + steht. Somit ist das Ergebnis:

$$D = +(B - A + 15 + 1) \bmod 16 = +(B - A)$$

Die XOR-Gatter spielen hier keine Rolle, da deren zweiter Eingang auf 0 liegt.

- ii) $A > B$: S bleibt kleiner als 16, also wird das Übertragsbit auf 0 gelegt. Diese 0 wird negiert und dann zum einen als Minus-Vorzeichen ausgegeben und zum anderen zum Aktivieren der XOR-Gatter als Inverter genutzt, da das Ergebnis kleiner Null ist. Somit wird folgendes gerechnet:

$$D = -S = \overline{(B - A + 15)} = -(15 - (B - A + 15)) = -(A - B) = B - A$$

- iii) $A = B$: Wenn das Übertragsbit von einer vorhergehenden Rechnung noch auf 1 liegt, wird folgendes gerechnet:

$$S = B + \bar{A} + 1 = B - A + 15 + 1$$

Also wird $S \bmod 16 = B - A = 0$ mit positivem Vorzeichen ausgegeben.

Ist das Übertragsbit auf 0 gesetzt, so sind die XOR-Gatter als Inverter aktiv und es wird folgendes gerechnet:

$$S = \overline{B + \bar{A}} = \overline{B - A + 15} = 15 - B + A - 15 = A + B = 0$$

Da allerdings das Übertragsbit auf 0 liegt, wird ein negatives Vorzeichen ausgegeben: $D = -S = -0$.

4 Flip-Flops

4.1 RS-Flip-Flop

Ein RS-Flip-Flop besteht aus zwei NAND-Gattern, deren Ausgang jeweils mit einem Eingang des anderen Gatters verbunden ist (siehe Abbildung 12).

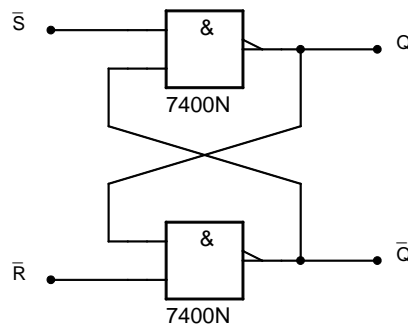


Abbildung 12: Schaltskizze eines RS-Flip-Flops

Wird S auf 1 gelegt, dann liegt Q auf 1. Auch die Eingänge des anderen Gatters liegen somit auf 1, weshalb \bar{Q} auf 0 liegt. Da diese 0 auch an einem Eingang des ersten Gatters anliegt, bleibt Q auf 1, auch wenn S wieder auf 0 gesetzt wird. Die 1 wird gespeichert. Wird \bar{R} auf 1 gelegt, so springt \bar{Q} auf 1, was zur Folge hat, dass Q auf 0 liegt. Dieser Zustand bleibt auch gespeichert (Siehe Tabelle 9). Somit eignet sich das RS-Flip-Flop zur Speicherung eines Bit.

Der Zustand $R = S = 1$ führt zur Aussage $Q = \bar{Q}$. Da dies logisch keinen Sinn ergibt, ist dieser Zustand verboten.

S	R	Q_n	\bar{Q}_n	
0	0	Q_{n-1}	\bar{Q}_{n-1}	Speicherzustand
1	0	1	0	Q wird auf 1 gesetzt
0	1	0	1	Q wird auf 0 gesetzt
1	1	1	1	Verbotener Zustand

Tabelle 9: Funktionstabelle eines RS-Flip-Flops

4.2 RST-Flip-Flop

Eine Erweiterung des RS-Flip-Flops ist das RST-Flip-Flop. Bei diesem Typ ist vor jeweils einen Eingang des RS-Flip-Flops ein NAND-Gatter geschaltet, welche nur dann ein Schaltsignal am R- oder S-Eingang durchlassen, wenn der Takteingang T auf 1 liegt (siehe Abbildung 13 und Tabelle 10).

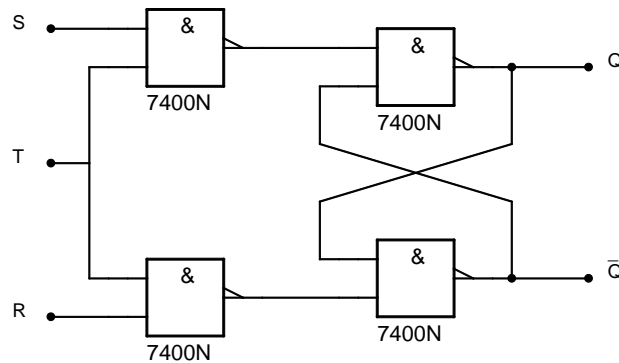


Abbildung 13: Schaltskizze eines getakteten RS-Flip-Flops

S	R	T	Q_n	\bar{Q}_n	
0	0	0	Q_{n-1}	\bar{Q}_{n-1}	keine Änderung
1	0	0	Q_{n-1}	\bar{Q}_{n-1}	keine Änderung
0	1	0	Q_{n-1}	\bar{Q}_{n-1}	keine Änderung
1	1	0	Q_{n-1}	\bar{Q}_{n-1}	keine Änderung
0	0	1	Q_{n-1}	\bar{Q}_{n-1}	keine Änderung
1	0	1	1	0	Q wird auf 1 gesetzt
0	1	1	0	1	Q wird auf 0 gesetzt
1	1	1	1	1	Verbotener Zustand

Tabelle 10: Funktionstabelle eines getakteten RS-Flip-Flops

Data-Flip-Flop

Verbindet man bei einem RST-Flip-Flop den R-Eingang mit dem negierten S-Eingang (hier mit dem Ausgang des NAND-Gatters) (siehe Abbildung 14), so speichert das Flip-Flop, wenn an T 1 anliegt, das *HIGH*- oder *LOW*-Signal am diesem neuen Eingang, dem Data-Eingang D. Durch die Invertierung ist S immer ungleich R, weshalb es hier keinen verbotenen Zustand gibt.

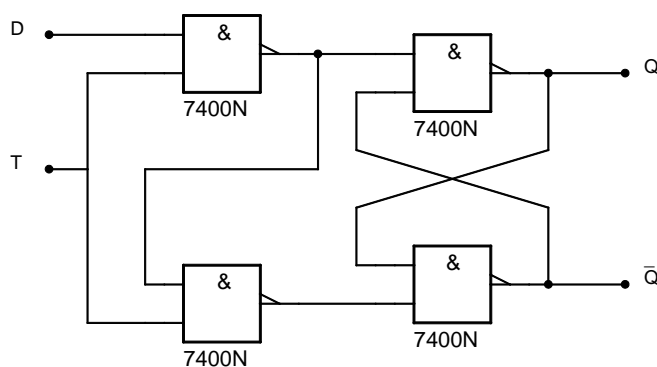


Abbildung 14: Schaltskizze eines Data-Flip-Flops

Entprellen eines Schalters

Bei einem mechanischen Schalter passiert es, dass beim Umschalten der Kontakt noch etwas nachschwingt und dieser dadurch mehrfach kurz unterbrochen wird. Auch wenn dieser Vorgang sehr schnell vonstatten geht, so schalten Flip-Flops, die auf Zustandsänderungen reagieren, trotzdem schnell genug, um auf dieses Prellen zu reagieren. Dies kann zu unerwünschten Nebeneffekten führen, wie das ungewollte Schalten eines Flip-Flops. Um diesen Effekt zu vermeiden, kann der Schalter mittels eines RS-Flip-Flops entprellt werden (siehe Abbildung 15). Solange der Kontakt nur unterbrochen und wiederhergestellt wird, ändert sich am Zustand des RS-Flip-Flops nichts. Erst wenn umgeschaltet wird, ändert sich der Zustand und bleibt auch trotz Prellens konstant. Daher wird diese Schaltung im Folgenden für die Schaltung des Taktsignals verwendet.

4.3 JK-Master-Slave-Flip-Flop

Es wird ein JK-Master-Slave-Flip-Flop wie in der Vorbereitungsmappe beschrieben aufgebaut (siehe Abbildung 16).

Diese Art von Flip-Flop besteht aus zwei hintereinander geschalteten RST-Flip-Flops, wobei das Taktsignal zwischen dem Master- (dem vorderen) und dem Slave-Flip-Flop (dem hinteren) negiert wird. Dadurch ist immer nur eines der Flip-Flops

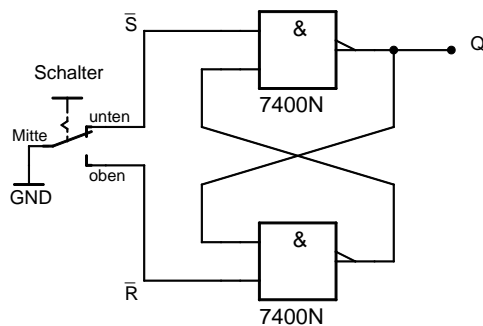


Abbildung 15: Schaltskizze eines RS-Flip-Flops zum entprellen

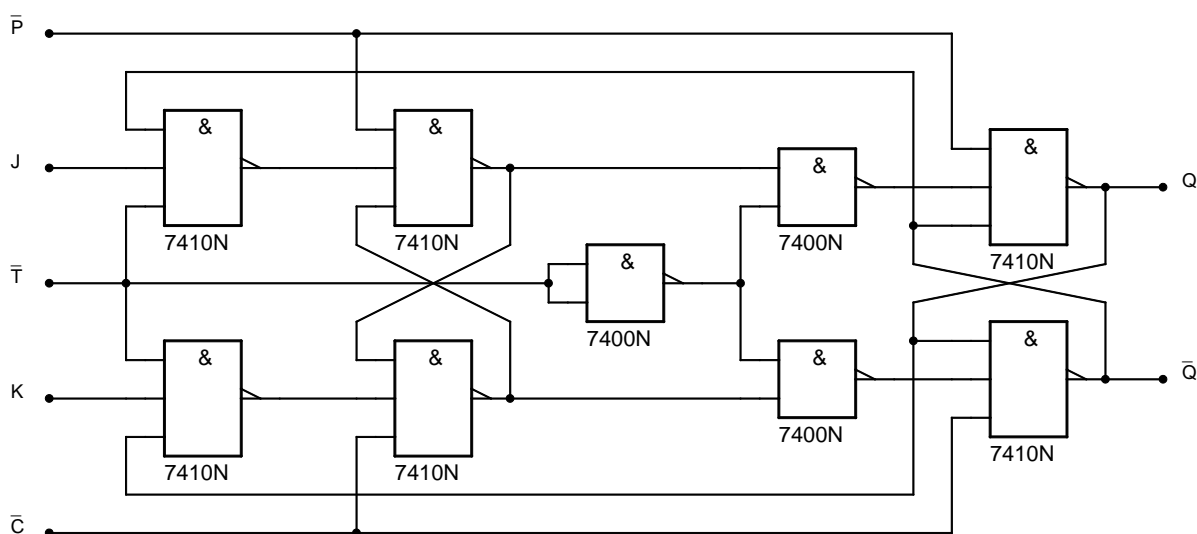


Abbildung 16: Schaltskizze eines Jump-Kill-Master-Slave-Flip-Flops

aktiviert und nie beide gleichzeitig. Daher schaltet dieses Flip-Flop nur bei Zustandsänderungen am Takteingang, bei sogenannten Taktflanken (siehe Tabelle 11).

Im Gegensatz zum RST-Flip-Flop kann hier, solange T auf 0 liegt, der Master-Flip-Flop mit J und K beliebig hin- und hergeschaltet werden. Erst wenn sich das Signal an T von 0 nach 1 ändert (+) wird der Slave-Flip-Flop entsprechend dem Master geschaltet. Da das Taktsignal zwischen Master und Slave invertiert wird, kann immer nur einer der beiden aktiv sein, sonst wäre dies wieder nur eine einfache Taktzustandssteuerung. Erst wenn T wieder auf 0 liegt, kann der Master wieder geändert werden. Dadurch ist der Zeitpunkt, an welchem die Information gespeichert werden soll, scharf begrenzt.

Mittels P und C kann der Wert des JK-MS-Flip-Flops direkt gesetzt werden, al-

J	K	Taktflanke	Q_{Master}^{n+1}	Q_{Slave}^{n+1}	
0	0	+	Q_n	Q_{Master}	Setze Slave auf Master
0	0	-	Q_n	Q_n	keine Änderung
1	0	+	Q_n	Q_{Master}	Setze Slave auf Master
1	0	-	1	Q_n	Setze Master auf 1
0	1	+	Q_n	Q_{Master}	Setze Slave auf Master
0	1	-	0	Q_n	Setze Master auf 0
1	1	+	Q_n	Q_{Master}	Setze Slave auf Master
1	1	-	\bar{Q}_n	Q_n	Master togglen

Tabelle 11: Funktionstabelle eines Jump-Kill-Master-Slave-Flip-Flops

lerdings muss hierbei auf den verbotenen Zustand $P = C = 1$ geachtet werden!

Durch den dritten Eingang der ersten NAND-Gatter wird auch der verbotene Zustand entschärft, da das Gatter nur durchschaltet, wenn der gegenüberliegende Ausgang invertiert ist. Ein gleichzeitiges *HIGH* an J und K bewirkt somit nur ein Umschalten: In den Master wird der invertierte Wert des Slaves geschrieben.

5 Schieben, Multiplizieren, Rotieren

5.1 4-Bit-Schieberegister

Es wird ein 4-Bit-Schieberegister wie in der Vorbereitungsmappe beschrieben aufgebaut (siehe Abbildung 17). Ein solches Schieberegister besteht aus hintereinander

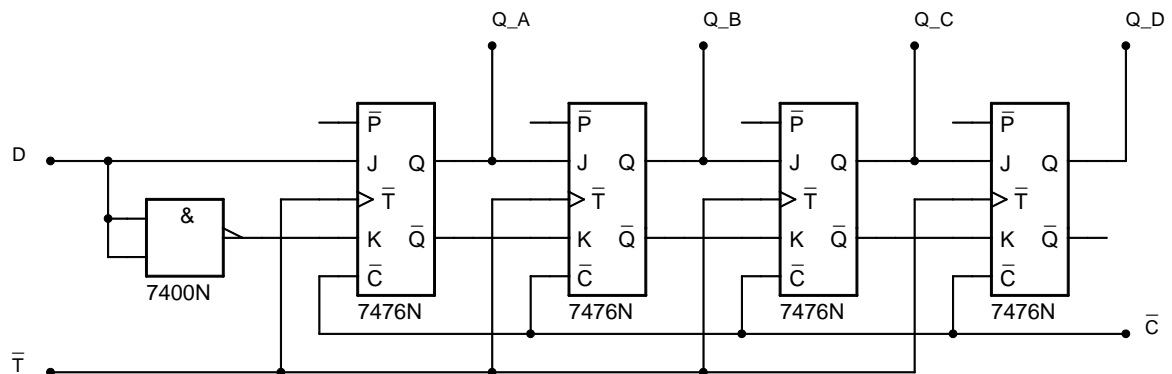


Abbildung 17: Schaltskizze eines 4-Bit-Schieberegisters

geschalteten JK-MS-Flip-Flops, wobei immer Q mit J und \bar{Q} mit K verbunden wird. Das Eingangssignal wird am den J-Eingang und negiert am K-Eingang des ersten Flip-Flops eingegeben. Bei positiver Taktflanke wird eine Information immer in das nächste Flip-Flop gegeben, die Informationen werden nach rechts „geschoben“. Somit werden die seriell eingegebenen Bits an den verschiedenen Ausgängen gleichzeitig ausgegeben. Das Schieberegister stellt also einen Seriell-Parallel-Wandler dar.

Da beim Prellen des Taktsignals die Bits weitergeschoben werden, ist eine Entprellschaltung wie in Abbildung 15 zu verwenden.

Über die Preset und Clear-Eingänge kann der Zustand der einzelnen Flip-Flops direkt eingegeben werden.

5.2 4-Bit-Rotationsregister

Wird bei einem Schieberegister wie im letzten Kapitel der Ausgang des letzten Flip-Flops mit dem Eingang des Ersten verbunden (siehe Abbildung 18), so wandern die Informationen immer wieder im Kreis, sie rotieren durch das Register. Dabei wird über P_A bis P_D der Ausgangszustand des Registers festgelegt und bei positiven Taktflanken des Taktsignals rotiert. Hiermit lässt sich zum Beispiel ein Lauflicht realisieren.

Statt des Inverters am ersten K-Eingang kann auch der \bar{Q} -Ausgang des letzten Flip-Flops mit dem ersten K-Eingang verbunden werden.

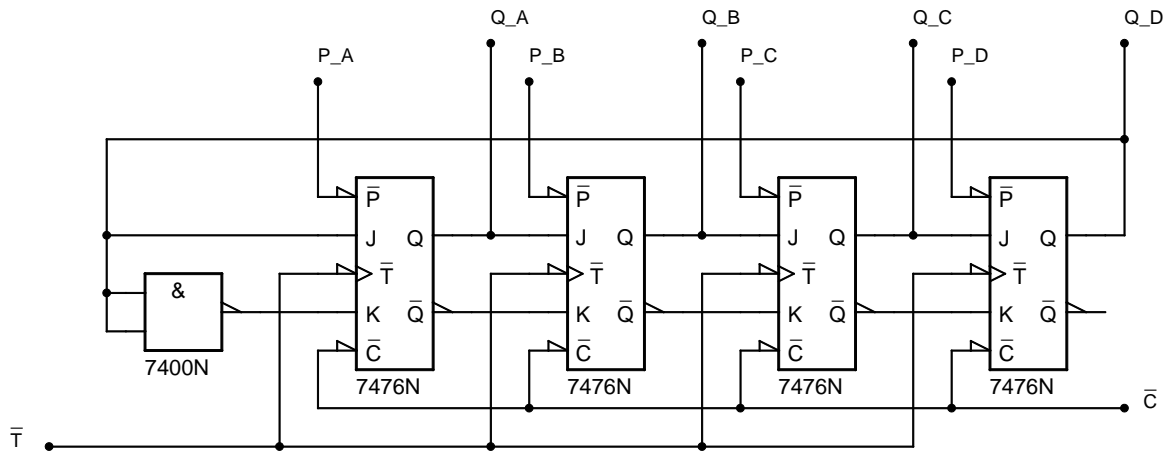


Abbildung 18: Schaltskizze eines 4-Bit-Rotationsregisters

6 Zähler

6.1 4-Bit-Asynchrone Zähler

Bei einem Asynchrone Zähler werden wieder mehrere JK-MS-Flip-Flops hintereinander geschaltet, allerdings wird der Q-Ausgang immer mit dem Takteingang des nächsten Flip-Flops verbunden (Siehe Abbildung 19). Die J- und K-Anschlüsse werden offen gelassen, damit liegen sie auf 1 und die Flip-Flops fungieren jeweils als Umschalter (Toggle).

Somit verdoppelt sich nach jedem Flip-Flop die Anzahl der zum Schalten benötigten Taktflanken. Daher zählt der Zähler binär nach oben, wobei das erste Bit dem Zustand des ersten Flip-Flops entspricht. Allerdings halbiert sich dadurch auch der

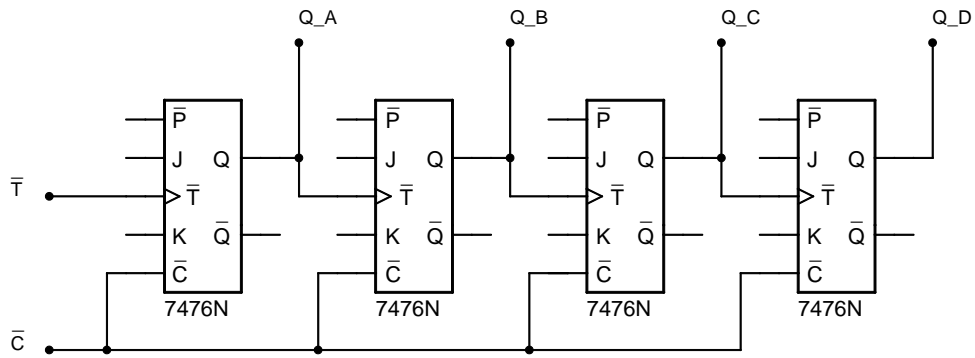


Abbildung 19: Schaltskizze eines 4-Bit-Asynchronezählers

Takt nach jedem Flip-Flop, weshalb der Zähler asynchron arbeitet. Darüber hinaus addieren sich die Laufzeiten durch die einzelnen Flip-Flops, weshalb dieser Zähler für genaue schnelle Zählungen ungeeignet ist.

Um rückwärts zu zählen, muss die Komplementärzahl gebildet werden. Dies erreicht man, indem man statt an Q an \bar{Q} das Ausgangssignal abgreift, aber Q mit dem Taktsignal des nächsten Zählers verbunden lässt. Alternativ kann das Ausgangssignal an Q abgegriffen werden und \bar{Q} wird mit dem Takteingang des nächsten Flip-Flops verbunden.

6.2 Asynchroner Dezimalzähler

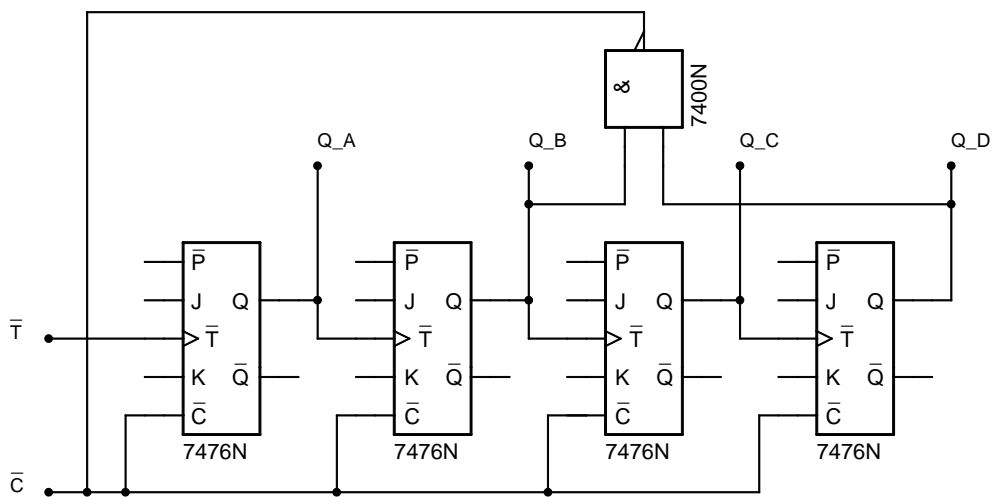


Abbildung 20: Schaltskizze eines dezimalen 4-Bit-Asynchronezählers

Wenn der Asynchronezähler aus dem letzten Kapitel nur bis zur dezimalen Zehn zählen soll, muss bei binär 1010 ($\hat{=}$ 10 dezimal) der Zähler über die C-Eingänge

resettet werden. Da nur vorwärts gezählt wird, reicht es, die Ausgänge Q_B und Q_D mittels NAND-Gatter an die C-Eingänge der Flip-Flops anzuschließen (siehe Abbildung 20). Dieses Gatter schaltet dann beim Erreichen der dezimalen 10 auf 0 und resettet dadurch den Zähler.

6.3 4-Bit-Synchronzähler

Beim Synchronzähler wird das Anfangs-Taktsignal an den Takteingang eines jeden Flip-Flops gegeben. Damit aber ein Flip-Flop nur dann schaltet, wenn alle vorhergehenden Flip-Flops auf 1 liegen, müssen alle vorhergehenden Q-Ausgänge über AND an den J- und K-Eingang angeschlossen werden, um nur dann das Togglen zu aktivieren. Dies lässt sich einfacher realisieren, indem die AND-Gatter staffelt, was heißt, dass an einem Eingang des AND-Zählers der Ausgang des vorhergehenden AND-Gatters anliegt (siehe Abbildung 21).

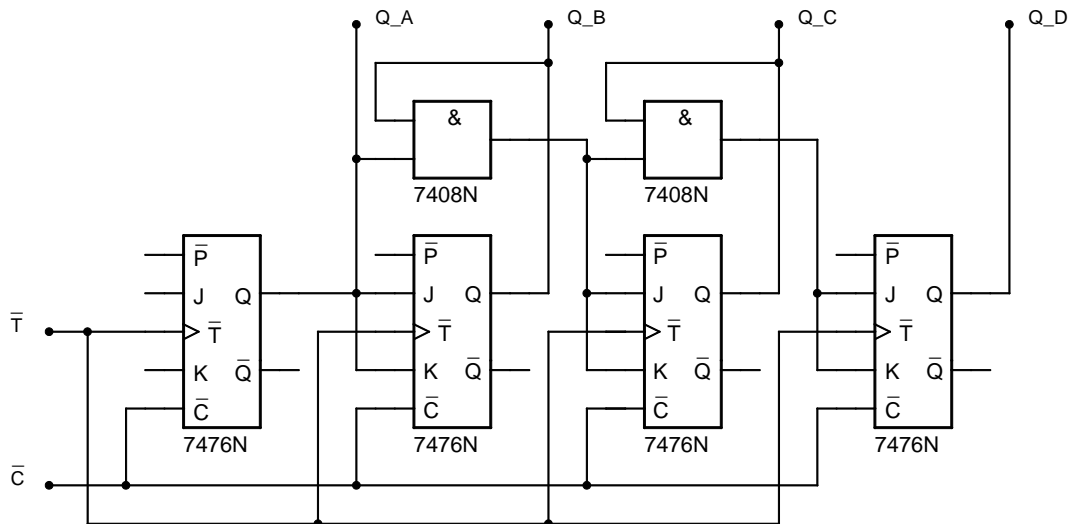


Abbildung 21: Schaltskizze eines 4-Bit-Synchronzählers mit gestaffelten AND-Gattern

6.4 Synchroner Dezimalzähler

Damit der Synchronzähler aus dem letzten Kapitel zum Dezimalzähler wird, müssen hier wie in Kapitel 6.2 die Ausgänge Q_B und Q_D mittels NAND-Gatter an die C-Eingänge der Flip-Flops angeschlossen werden. Alternativ können auch die betreffenden Bits mittels geschickter Schaltung von AND-Gattern direkt getogget werden.

7 Digital-Analog-Wandlung

Da wir im Dezimalzähler neun Stufen haben (zuzüglich der Null), muss durch jede Stufe der Strom um ein Zehntel des Maximalstroms erhöht werden:

$$I = \frac{I_{\max}}{n} = \frac{90\% \cdot 100 \mu\text{A}}{9} = 10 \mu\text{A}$$

Somit muss bei der niedrigsten Stufe ein Strom von $10 \mu\text{A}$ fließen. Daraus folgt für den Widerstand an der ersten Stufe:

$$R_1 = \frac{U}{I} = \frac{4 \text{ V}}{10 \mu\text{A}} = 400 \text{ k}\Omega$$

Da in der zweiten, vierten und achten Stufe jeweils der doppelte Strom der vorhergehenden Stufe fließen soll, folgt für die Widerstände der weiteren Stufen:

$$\begin{aligned} R_2 &= \frac{R_1}{2} = 200 \text{ k}\Omega \\ R_3 &= \frac{R_2}{2} = \frac{R_1}{4} = 100 \text{ k}\Omega \\ R_4 &= \frac{R_3}{2} = \frac{R_1}{8} = 50 \text{ k}\Omega \end{aligned}$$

Diese Widerstände werden der Reihe nach an die Ausgänge eines Dezimalzählers angeschlossen, wobei R_1 an Q_A angeschlossen wird (siehe Abbildung 22). Da die Widerstände parallel geschaltet sind, addieren sich ihre Ströme zum Gesamtstrom, welcher nun proportional zur Zählstufe des Dezimalzählers ist. Dieser Gesamtstrom wird mit dem Multimeter zwischen den parallel zusammengeschalteten Anschlüssen der Widerstände und der Masse gemessen.

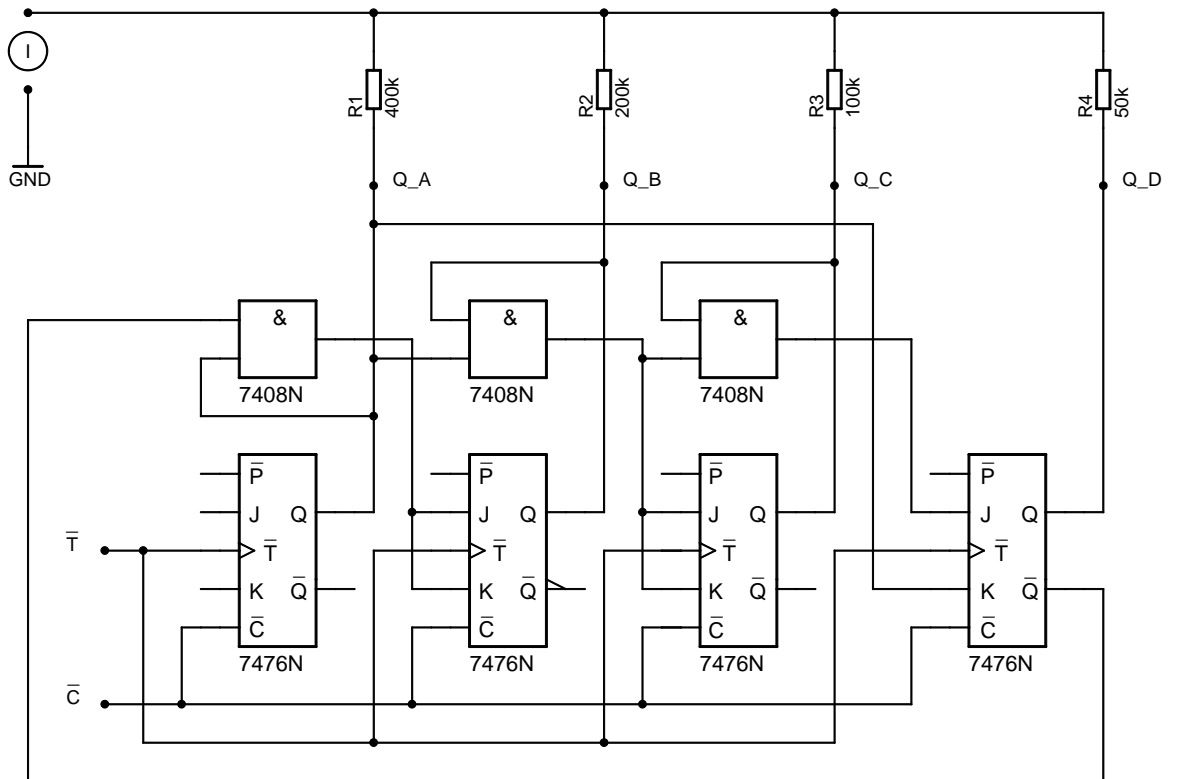


Abbildung 22: Schaltskizze eines 4-Bit-Synchron-Dezimalzählers als Digital-Analog-Wandler