

Versuchsvorbereitung P1-63: Digitale Elektronik, Schaltlogik

Kathrin Ender
Gruppe 10

28. Oktober 2007

Inhaltsverzeichnis

0	Vorbemerkung	3
1	Gatter aus diskreten Bauelementen	3
1.1	AND-Gatter	3
1.2	NOT- und NAND-Gatter	3
1.3	OR-Gatter	4
2	Logische Funktionen mit ICs	4
2.1	Inverter	4
2.2	EXOR	4
2.3	EXOR mit NAND-Gattern	5
3	Addierer	5
3.1	Halbaddierer	5
3.2	Volladdierer	5
3.3	Subtrahierer	6
4	Speicherelemente	6
4.1	RS-Flip-Flop (RS-FF)	6
4.2	Getakteter RS-Flip-Flop	6
4.3	JK-Master-Slave-Flip-Flop	7
5	Schieben, Multiplizieren, Rotieren	7
5.1	4-Bit-Schieberegister	7
5.2	Rotationsregister	8
6	Zähler	8
6.1	4-Bit-Asynchrnzähler	8
6.2	Asynchrone Deziinalzähler	8
6.3	4-Bit-Synchrnzähler	9
6.4	Synchrone Deziinalzähler	9
7	Digital-Analog-Wandlung	9

0 Vorbemerkung

Bei diesem Versuch geht es darum die Technik der digitalen Elektronik und deren Schaltlogik kennen zu lernen. Die wichtigsten Grundschaltungen werden erst einmal mittels Dioden, Widerständen, Transistoren, usw. realisiert. Später werden dann kompliziertere Schaltungen mit vorgefertigten Grundschaltungen aufgebaut.

In der digitalen Elektronik ist nicht der tatsächliche Wert eines Potentials von Bedeutung, sondern es wird lediglich zwischen einem hohen (high H) und tiefen (low L) Potential, also zwischen zwei Zuständen unterschieden. Ein high ist äquivalent zu logisch 1 und low ist äquivalent zu logisch=0.

Die in den folgenden Aufgaben zu zeichnenden Schaltskizzen sind im Anhang zu finden.

1 Gatter aus diskreten Bauelementen

1.1 AND-Gatter

Ein AND-Dioden-Gatter kann man wie im Anhang gezeigt aufbauen. Liegen einer der Eingänge oder beide am niedrigen Potential (also auf logisch 0), kann der Strom über die Dioden abfließen, so dass fast die gesamte Spannung am Widerstand R abfällt. C liegt damit also (da keine Spannung abfällt) auf logisch 0. Liegen die Eingänge A und B aber am hohen Potential (also auf logisch 1), so sperren die Dioden und damit wird ihr Widerstand groß, so dass fast die gesamte Spannung dort abfällt. C ist also logisch 1.

Wahrheitstabelle:

A	B	C=A∧B
1	1	1
1	0	0
0	1	0
0	0	0

1.2 NOT- und NAND-Gatter

Zuerst soll ein Transistor-NOT-Gatter gebaut werden. Aus diesem wird dann durch zusammenschalten mit dem AND-Gatter aus 1.1 ein NAND-Gatter realisiert.

NOT-Gatter: Schließt man den Eingang des NOT-Gatters an das hohe Potential an, so fließt ein Strom durch den Transistor. Fast die gesamte Spannung fällt dann an R_1 ab und damit ist B logisch 0. Liegt der Eingang am niedrigen Potential so fließt kein Strom, damit ist B logisch 1.

Wahrheitstabelle:

A	B=¬A
1	0
0	1

NAND-Gatter: Lässt sich leicht durch das Hintereinanderschalten eines AND- und eines NOT-Gatters erreichen.

Wahrheitstabelle:

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

1.3 OR-Gatter

Sobald einer oder beide der Eingänge am hohen Potential anliegen, fließt Strom durch die Dioden und alle Spannung fällt am Widerstand R ab, C ist also logisch 1. Liegt sowohl A als auch B an der Erdung (am niedrigen Potential) ist C logisch 0.

Wahrheitstabelle:

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

2 Weiter einfache logische Funktionen, realisiert mit ICs

2.1 Inverter

Ein digitaler Inverter ist im Prinzip ein NOT-Gatter. Er lässt sich aus einem NAND- oder einem NOR-Gatter realisieren, wenn man jeweils deren Eingänge miteinander verbindet.

Wahrheitstabelle:

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

Man könnte auch bei einem NAND-Gatter einen Eingang fest auf 1 legen, oder bei einem NOR-Gatter einen Eingang fest auf 0.

2.2 EXOR

Wahrheitstabelle:

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

Aus der Wahrheitstabelle kann man die disjunktive Normalform ablesen. Sie lautet:
 $C = (\neg A \wedge B) \vee (A \wedge \neg B)$ Eine EXOR-Schaltung kann also mit zwei NAND-, zwei AND- und einem OR-Gatter realisiert werden.

2.3 EXOR mit NAND-Gattern

Die disjunktive Normalform aus 2.2 lässt sich so umformen, dass eine EXOR-Schaltung nur mit NAND-Gattern gebaut werden kann.

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

Man benötigt also vier NAND-Gatter.

3 Addierer

3.1 Halbaddierer

Der Halbaddierer soll zwei einstellige Dualzahlen addieren. Er soll also die Summe S und den Übertrag C (Carry) ausgeben.

Wahrheitstabelle:

A	B	S	C
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

Legt man bei einem EXOR- bzw. einem AND-Gatter jeweils A und B an die Ausgänge, so wird die Summe S bzw. der Übertrag C ausgegeben.

3.2 Volladdierer

Wahrheitstabelle:

A	B	C_1	S	C_2
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

Der Volladdierer kann aus zwei Halbaddierern und einem OR-Gatter gebaut werden. Durch das Hintereinanderschalten mehrerer 1-Bit-Volladdierer kann man mehrstellige Dualzahlen addieren.

3.3 Subtrahierer

In der Schaltung für den 4-Bit-Subtrahierer (siehe Aufgabenblatt; Abb. 4) wird ein 4-Bit-Addierer verwendet, da $B-A=B+(-A)$. Nun muss man lediglich eine duale Darstellung für eine negative Zahl finden. Es gilt: $A+\bar{A}=1111=10000-0001$ und damit $-A=\bar{A}+0001-10000$.

Subtraktion positiver Differenzen ($B>A$): B und das invertierte A werden vom Addierer addiert, das Hinzuaddieren von 0001 wird erreicht indem man den Übertragseingang an den Übertragsausgang, der 1 ausgibt, anlegt. Die Ergebnisse der Einzeladditionen werden an die Eingänge von EXOR-Gattern gelegt, welche das Endergebnis der Subtraktion ausgeben sollen. Die anderen Eingänge dieser EXOR-Gatter sind über einen vorgeschalteten Invertierer mit dem Übertragsausgang des Addierers verbunden. Gibt dieser also eine 1 aus, was bei positiven Differenzen immer der Fall ist, kommt an den EXOR-Gattern jeweils eine Null an, so dass die Ausgangssignale der Additionen nicht verändert werden. Bei positiven Differenzen liegt SGN auf logisch 0, was gleich bedeutend ist mit einem positiven Vorzeichen.

Subtraktion negativer Differenzen ($B<A$): B und das invertierte A werden addiert und das Ergebnis wird an die EXOR-Gatter weitergeleitet. An den anderen Eingängen der EXOR-Gatter liegt, da der Übertrag der Addition bei negative Differenzen immer Null ist, logisch 1 an. Sie wirken also invertierend auf die Ausgabe des Addierers. Auch SGN liegt auf logisch 1, was das negative Vorzeichen anzeigt.

4 Speicherelemente

4.1 RS-Flip-Flop (RS-FF)

Ein RS-Flip-Flop besteht aus zwei NAND-Gattern, wobei die Ausgänge der beiden Gatter jeweils an einem der Eingänge des anderen Gatters anliegen. Aus der Schaltskizze lässt sich ablesen das gelten muss: $Q=\neg(S \wedge \neg(R \wedge Q))=\neg S \vee (R \wedge Q)$ bzw.

$$\neg Q=\neg(R \wedge \neg(S \wedge \neg Q))=\neg R \vee (S \wedge \neg Q),$$

wobei Q und $\neg Q$ die Werte der Ausgänge der NAND-Gatter sind. Daraus ergibt sich folgende Wahrheitstabelle:

S	R	Q	$\neg Q$	Zustand
1	1	Q	$\neg Q$	Speicher
1	0	0	1	Zurücksetzen
0	1	1	0	Setzen
0	0	0	0	verbotener Zustand

Stehen sowohl R als auch S auf eins, so bleiben die Werte für Q und $\neg Q$ erhalten, es handelt sich also um den Speicherzustand. Ist $R=1$ und $S=0$ so wird Q auf logisch 1 gesetzt. Für $R=0$ und $S=1$ wird Q hingegen gelöscht, bzw. auf logisch 0 gesetzt. $R=S=0$ ist der verbotene Zustand, da dann sowohl Q als auch $\neg Q$ logisch 1 wären, was ein Widerspruch ist. Beim RS-Flip-Flop handelt es sich also um einen Speicher.

4.2 Getakteter RS-Flip-Flop

Beim getakteten RS-Flip-Flop (siehe Aufgabenblatt, Abb. 5) ist dem einfachen FF ein Taktgeber vorgeschaltet. Der Wert des Takts wird zusammen mit den Werten von R und S jeweils an NAND-Gatter angelegt, deren Ausgänge mit den Eingängen des FF verbunden sind. Steht der Takt auf 0, so wird zwei mal der Wert 1 in den FF eingegeben, unabhängig von den Werten für R und S. Daher ist dies der Speicherzustand. Für $T=1$ ergibt sie die Wahrheitstabelle eines „invertierten“ FFs (das NAND-Gatter wirkt als Invertierer der Eingangswerte R und S).

R	S	Q	$\neg Q$	Zustand
0	0	Q	$\neg Q$	Speicher
1	0	0	1	Zurücksetzen
0	1	1	0	Setzen
1	1	1	1	verbotener Zustand

Nun wird auch der Grund für die Bezeichnung der Eingänge R und S klar. S steht für „set“, da Q für S=1 gesetzt wird. R steht für „reset“, da Q auf 0 gesetzt wird. Der Zustand R=S=0, also der Speicherzustand für T=1, ist an sich unnötig für den RST-FF, da T=0 schon als Speicherzustand fungiert. Daher kann man um den verbotenen Zustand (R=S=1) zu vermeiden, R= \neg S erzwingen, indem man den Eingang R mit A verbindet.

4.3 JK-Master-Slave-Flip-Flop

Der JK-MS-FF (siehe Aufgabenblatt, Abb.6) besteht aus zwei hintereinander geschalteten RST-FFs, dem so genannten Master und dem Slave. Die beiden sind an einen invertierten Takt angeschlossen. Dies hat zur Folge, dass sich immer einer der beiden in einem Speicherzustand befindet. Für T=0 ist der Master im Speicherzustand und für T=1 der Slave. Bei einem Taktwechsel 0 \rightarrow 1 übernimmt der Master die Eingangsinformationen von J und K, der Slave befindet sich jedoch im Speicherzustand. Er übernimmt dafür die Informationen vom Master bei einem 1 \rightarrow 0 Taktwechsel.

Die „vorrangigen“ Eingänge P und C sind zum Löschen bzw. Setzen der Informationen im JK-MS-FF. Für P=1 und C=0 wird der Q unabhängig von J und K auf logisch 0 gesetzt. Für P=0 und C=1 ist Q=1. P=C=0 gibt einen verbotenen Zustand und ist daher zu vermeiden. Gilt P=C=1 so sind diese Eingänge nicht von Bedeutung und es ergibt sich folgende Wahrheitstabelle:

Taktwechsel	J	K	q	\bar{q}	Q	\bar{Q}
0 \rightarrow 1	0	0	q	\bar{q}	Q	\bar{Q}
1 \rightarrow 0	0	0	q	\bar{q}	q	\bar{q}
0 \rightarrow 1	1	1	\bar{Q}	Q	Q	\bar{Q}
1 \rightarrow 0	1	1	q	\bar{q}	q	\bar{q}
0 \rightarrow 1	0	1	0	1	Q	\bar{Q}
1 \rightarrow 0	0	1	q	\bar{q}	q	\bar{q}
0 \rightarrow 1	1	0	1	0	Q	\bar{Q}
1 \rightarrow 0	1	0	q	\bar{q}	q	\bar{q}

Die Rückkopplung vom Slave zum Master sorgt dafür, dass wenn der JK-MS-FF einmal in einem gültigen Zustand war, er nicht mehr in den verbotenen kommen kann.

5 Schieben, Multiplizieren, Rotieren

5.1 4-Bit-Schieberegister

Ein 4-Bit-Schieberegister (siehe Aufgabenblatt, Abb.7) besteht aus 4 hintereinandergeschalteten JK-MS-FFs, die alle an den selben Takt angeschlossen sind. Wechselt man von T=0 auf T=1 so übernimmt der erste JK-MS-FF die Eingaben (J und K) in seinen Master. Beim Umschalten von T=1 auf T=0 übernimmt der Slave die Werte vom Master. Das heißt nach dem Taktzyklus 0 \rightarrow 1 \rightarrow 0 erscheint die Eingabe am Ausgang des ersten JK-MS-FFs. Die dahinter geschalteten JK-MS-FFs übernehmen beim ersten Teil des Taktzykluses die Slave-Ausgabe des Vorgeschalteten in ihren Master und beim zweiten Teil wandern diese Werte

aus dem Master in den Slave. Während eines Taktzykluses, wird ein Ausgabewert also um eine Ausgabestelle weitergeschoben, daher Schieberegister.

Die Eingaben (J und K) werden durch die Stellung eines Schalters erzeugt, durch ein NAND-Gatter wird außerdem erzwungen, dass die eingegebenen Werte komplementär sind. Bei geschlossenem Schalter ist $J=0$ und $K=1$, da der Schalter eine Erdung erzeugt, und es wird bei einem Taktzyklus eine 0 an der ersten Ausgabestelle erzeugt. Bei offenem Schalter ist $J=1$ und $K=0$. Es wird also bei einem Taktzyklus eine 1 im ersten Register erzeugt. Indem man C (Clear) auf Null legt, können alle Register auf Null gesetzt werden. Um ein Prellen beim Umschalten des Taktes zu vermeiden, wie es bei mechanischen Schaltern vorkommt, benutzt man einen FF als Taktgeber.

5.2 Rotationsregister

Ein Rotationsregister kann man realisieren, indem man bei einem Schieberegister den Ausgang Q_D mit den Eingängen des ersten JK-MS-FF verbindet. Der Wert der das D-Register erreicht hat, fällt nun nicht einfach „runter“, sondern wird wieder ins A-Register geschoben. Man kann also vorher über die Presets eingestellte Werte rotieren lassen, daher Rotationsregister.

6 Zähler

6.1 4-Bit-Asynchrone Zähler

Ein 4-Bit-Asynchrone Zähler lässt sich aus vier hintereinandergeschalteten JK-MS-FFs bauen (siehe Aufgabenblatt, Abb.8), wobei die Ausgänge des Vorgeschalteten mit dem Takt-Eingang des Nachfolgenden verbunden sind. Durch C (Clear) können alle Ausgaben (Q_A, Q_B, Q_C, Q_D) auf Null gesetzt werden. An E wird nun ein Taktzyklus angelegt. Beim Wechsel $0 \rightarrow 1$ wird q im Master des FFA auf 1 gesetzt, da der Master die invertierten Werte des Slaves übernimmt (da J und K nicht angeschlossen sind und damit auf 1 liegen), welcher auf 0 steht. Q im Slave bleibt auf Null. Beim Wechsel $1 \rightarrow 0$ wird der Wert vom Master an den Slave weitergegeben. Nach einem Taktzyklus steht der Slave also auf 1.

Bei einem weiteren Taktzyklus wird der Master erst auf 0 gesetzt, was schließlich an den Slave weitergegeben wird. Der FFA wechselt also bei jedem Taktzyklus seinen Ausgabewert. Da sein Ausgang an den Takteingang des nächsten FFs angeschlossen ist erhält dieser einen Taktimpuls, wenn der FFA gerade von 1 auf 0 umspringt. Dies setzt sich durch die Kette fort, so dass der Zähler hochzählt, wobei 15 die höchste darstellbare Zahl ist (danach springt er wieder auf Null). Die binär dargestellte Zahl lässt sich über $Z = Q_A + 2 \cdot Q_B + 4 \cdot Q_C + 8 \cdot Q_D$ ins Dezimalsystem umrechnen lassen. Der Zähler heißt asynchron, da die FF zeitlich nacheinander kippen.

6.2 Asynchroner Dezimalzähler

Um aus einem dualen Asynchrone Zähler einen dezimalen zu machen muss man ihn so modifizieren, dass er beim Erreichen der Zahl 10 ($Q_A = 0; Q_B = 1; Q_C = 0; Q_D = 1$) automatisch wieder auf 0000 umstellt. Dies lässt sich durch ein NAND-Gatter, dessen Eingänge mit Q_B und Q_D verbunden sind und dessen Ausgang an Clear anliegt erreichen. Wenn beide Eingaben eins sind (dies tritt zum ersten Mal beim Erreichen der 10 auf) so gibt das NAND-Gatter eine Null aus, so dass alle FFS gelöscht werden.

6.3 4-Bit-Synchronzähler

Beim Synchronzähler sollen alle FF gleichzeitig kippen. Dafür müssen alle am selben Takt liegen. Der FFA ändert mit jedem vollen Taktzyklus seinen Wert, da $J_A = K_A = 1$. Der FFB muss seinen Wert ändern, wenn der FFA im vorherigen Taktzyklus auf 1 gestellt wurde. Dies erreicht man, indem man die JK-Eingänge des FFB an den Ausgang des FFA legt. Der FFC wiederum muss umspringen, wenn der FFB und der FFA vor dem Taktimpuls auf 1 standen. Daher legt man seine JK-Eingänge an den Ausgang eines AND-Gatters, dessen Eingänge mit den Ausgängen des FFA und des FFB verbunden sind. Die JK-Eingänge des FFD verbindet man mit dem Ausgang eines weiteren AND-Gatters, dessen Eingänge mit dem Ausgang des ersten AND-Gatters und dem Ausgang des FFC verbunden sind. Der FFD schaltet also nur um, wenn vor dem Taktimpuls alle drei vorgeschalteten FFs auf eins standen. Sind diese Umschaltbedingungen für einen der drei letzten FFs nicht erfüllt, so stehen seine JK-Eingänge beide auf 0 und der angezeigte Wert wird beibehalten.

6.4 Synchroner Dezimalzähler

Einen Synchrondezimalzähler erhält man, indem man einen dualen Synchronzähler so verändert, dass er von der Zahl 9 ($Q_A = 1$; $Q_B = 0$; $Q_C = 0$; $Q_D = 1$) direkt wieder zur 0 schaltet. Es muss verhindert werden, dass Q_B auf eins umspringt, wenn $Q_A = 1$ und $Q_D = 1$. Dies ist möglich, indem man $\overline{Q_D}$ und Q_A an ein AND-Gatter anschließt, dessen Ausgang mit den beiden Eingängen des FFB verbunden wird.

Jetzt muss noch Q_D von 1 auf 0 gesetzt werden beim Taktimpuls nach der 9. Hierzu legt man den Ausgang des zweiten AND-Gatters nur noch an den J-Eingang des FFDs. An den K-Eingang wird Q_A gelegt. Für die Eingänge des FFDs gilt dann:

- $J=1$, wenn alle FFs davor 1, sonst $J=0$; $K=Q_A$
- $J=0$, $K=0$; tritt auf bei $0 \rightarrow 1$; $2 \rightarrow 3$; $4 \rightarrow 5$; $6 \rightarrow 7$: nichts ändert sich $Q_D = 0$
- $J=1$, $K=1$; tritt auf bei $7 \rightarrow 8$: Q_D wechselt von 0 zu 1
- $J=0$, $K=1$; tritt auf bei $1 \rightarrow 2$; $3 \rightarrow 4$; $5 \rightarrow 6$; $9 \rightarrow 10$: Q_D wird auf Null gesetzt

7 Digital-Analog-Wandlung

An einen Dezimalzähler, soll über ein geeignetes Widerstandsnetzwerk ein Drehimpulsmessgerät angeschlossen werden. Da beim Zählerstand 9 90% des möglichen Vollausschlages erreicht sein soll und $I_{max} = 100\mu A$, muss bei 1, wenn also nur Q_A auf einem hohen Potential liegt ein Ausschlag von 10% bzw. eine Stromstärke von $10\mu A$ erreicht sein. Bei 2, wenn nur Q_B auf einem hohen Potential liegt, soll der Ausschlag 20% bzw. die Stromstärke $20\mu A$ sein. Bei 4, wenn nur Q_C auf einem hohen Potential liegt, ist ein Ausschlag von 40% bzw. eine Stromstärke von $40\mu A$ zu erwarten. Bei 8, wenn nur Q_D auf hohem Potential liegt, sollte der Ausschlag 80% und die Stromstärke $80\mu A$ betragen. Mit diesen Voraussetzungen lässt sich über $R_i = \frac{4V}{I_i}$ der jeweilige Widerstand berechnen.

- $R_A = 400k\Omega$
- $R_B = 200k\Omega$
- $R_C = 100k\Omega$
- $R_D = 50k\Omega$