

Allgemeines zu unserem Experimentiermaterial

- a) Das Netzgerät liefert für **zwei** Teams die benötigte Gleichspannung von 5 Volt sowie ein Rechtecksignal auf der Basis der Netzfrequenz (bei uns 50 Hz). Eine Kontroll-LED leuchtet, wenn der Trafo am Netz liegt. Jedes Team kann maximal 0.5 Ampère Strom ziehen, was aber kaum je eine Einschränkung bedeutet.
- b) Die Steckplatte etwa in der Grösse eines A4-Blattes dient der mechanischen Fixierung der Bauteile sowie ihrer Stromversorgung. Stifte auf der Unterseite der Bauteile sorgen dafür, dass sie nicht falsch gepolt angeschlossen werden können. Legen Sie die Steckplatte im **Querformat** vor sich hin.
- c) Über eine Adaptereinheit mit Taktgeber und Taster wird die Stromversorgung auf das Steckbrett gebracht. Stecken Sie diesen Adapter oben links auf das Steckbrett (es geht nur im Hochformat). Stecken Sie das Anschlusskabel in die entsprechende Buchse des Netzgerätes.
Wenn Sie jetzt den roten Taster betätigen, sollte die LED aufleuchten. Dieser Taster ist **entprellt**.
Der Taktgeber (clock) mit dem 50 Hz respektive 1 Hz Ausgang ist nur aktiv, wenn sein Eingang R über ein (kurzes) Käbeli auf 0 heruntergezogen wird. Der 1 Hz Takt lässt sich über eine rote LED beobachten.
Beachten Sie, dass nicht alle kleinen Löcher in dem Gehäuse Buchsen sind.

Testen Sie den Taktgeber, indem Sie den 1 Hz Ausgang mit dem Takteingang des 4-bit-Binärzählers verbinden ! Sie können diesen Zähler natürlich auch von Hand takten über den roten Taster. Beachten Sie, dass der Zähler erst bei fallender Taktkante weiterzählt !

- d) Die 4-bit-Inputeinheit sollten Sie nun ebenfalls am linken Rand der Steckplatte unterhalb des Taktgebers anbringen. Sie können damit 4 Signale erzeugen, die als Input für die folgenden Schaltungen dienen. Wie immer leuchtet die rote LED, wenn eine 1 oder high ausgegeben wird.
- e) Die kleinen blauen Kabel sind (wie der Rest des Experimentiermaterials) sündhaft teuer, aber auch sehr praktisch. Jeder Stecker dient gleichzeitig als Buchse, die Stifte sind also 'stapelbar'. Ziehen Sie immer an den Steckern und **nie** nur am Kabel, wenn sie eine Verbindung trennen wollen !

Dieser Aufbau mit Taster/Taktgeber und Input-Einheit am linken Rand wird immer so bleiben. Lassen Sie das auch so gesteckt am Ende einer Sitzung.

Einige kleine Übungen:

1. Testen Sie die Input-Einheit. Stecken Sie das 4x AND rechts von der Input-Einheit auf die Steckplatte und verbinden Sie die beiden Eingänge des obersten AND-Gatters (dessen Output durch eine LED angezeigt wird) mit zwei Inputleitungen. Prüfen Sie alle 4 Kombinationen der Inputleitungen durch.
2. Dasselbe können Sie (wenn Sie wollen) auch mit den anderen Logikfunktionen machen.
3. Ein AND-Glied als **Torschaltung**: Verbinden Sie den 1 Hz Takt mit einem Eingang des obersten AND-Gliedes. Den anderen Eingang verbinden Sie mit einem Ausgang des Input-Adapters. Den Ausgang dieses AND-Gliedes führen Sie weiter auf den Takteingang des Zählers. Sie können nun steuern, ob das Taktsignal durchgehen soll oder nicht.
4. Erweiterung von 3. : Verbinden Sie den Reset-Eingang des 4-bit-Zählers mit einem weiteren Ausgang des Input-Adapters. Sie können nun durch ein Low-Signal den Zählerstand jederzeit auf 0 zurücksetzen.

Die Gesetze der Boole'schen Algebra und einige unserer Übungsaufgaben

1. Prüfen Sie mindestens eines der beiden Gesetze von de Morgan:

$$(\text{NOT}(A)) \text{ AND } (\text{NOT}(B)) = \text{NOT}(A \text{ OR } B) = A \text{ NOR } B$$

$$(\text{NOT}(A)) \text{ OR } (\text{NOT}(B)) = \text{NOT}(A \text{ AND } B) = A \text{ NAND } B$$

Setzen Sie die Inverter-Einheit gleich neben die Input-Einheit, damit Ihnen zu A und B auch NOT(A) und NOT(B) zur Verfügung stehen.

Erzeugen Sie die Signale für die linke und die rechte Seite der Gleichung und überprüfen Sie, ob sie wirklich für alle Kombinationen der Inputsignale übereinstimmen. Verwenden Sie jeweils für das Endergebnis das oberste Gatter (dasjenige mit der LED-Anzeige).

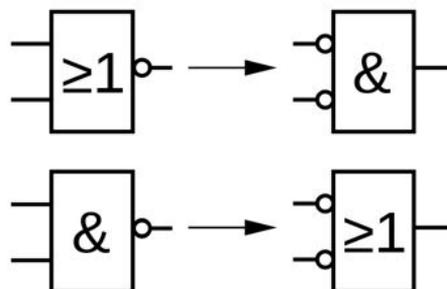
2. Prüfen Sie mindestens eines der beiden Distributivgesetze:

$$A \text{ AND } (B \text{ OR } C) = (A \text{ AND } B) \text{ OR } (A \text{ AND } C)$$

$$A \text{ OR } (B \text{ AND } C) = (A \text{ OR } B) \text{ AND } (A \text{ OR } C)$$

Erzeugen Sie die Signale für die linke und die rechte Seite der Gleichung und überprüfen Sie, ob sie wirklich für alle 8 Kombinationen der drei Inputsignale übereinstimmen. Verwenden Sie jeweils für das Endergebnis das oberste Gatter (dasjenige mit der LED-Anzeige).

3. NOR als Inverter: Übungsblatt Nr. 2a (siehe Anhang 1)
4. NAND als Inverter: Übungsblatt Nr. 2b (siehe Anhang 1)
5. Schaltung mit 4x NAND mit zwei Eingängen : Übungsblatt Nr. 3 (siehe Anhang 1)
6. Schaltung mit 4x NOR mit zwei Eingängen: Übungsblatt Nr. 4 (siehe Anhang 1)
7. Schaltung mit 4x NAND (oder 3x NAND und 1x Inverter) mit drei Eingängen: Übungsblatt Nr. 8 (siehe Anhang 1)
Diese Schaltung stellt eine 'digitale Weiche' dar (Multiplexer). Die Daten können dabei nur von links (A oder B) nach rechts (S) fließen.



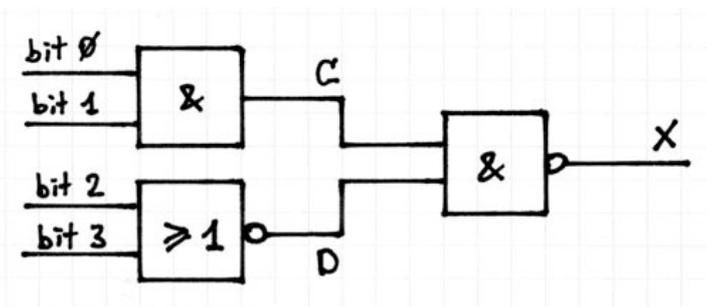
Die Gesetze von De Morgan in einer Darstellung durch Logik-Gatter
In der Darstellung rechts werden die Eingänge invertiert !

Versuche zur Decodierung von Binärzahlen

- Da alle Bausteine an denselben Datenleitungen lauschen braucht jeder eine eindeutige Hardwareadresse, mit der ihm signalisiert werden kann, dass er gemeint ist. Im Taschenrechner gibt es vielleicht 16 Adressleitungen, mit denen $2 \text{ hoch } 16 = 65'536$ Bausteine adressiert werden können. Moderne PC's haben bis zu 64 Adressleitungen ...

Wir können mit unserer Input-Einheit eine 4-bit-Adresse (0 bis 15 dezimal oder 0 bis F hexadezimal) erzeugen. Realisieren Sie nun für eine bestimmte Adresse a eine Schaltung, welche genau dann low liefert, wenn die Input-Einheit das Bitmuster für a liefert. In allen anderen Fällen soll der Output high sein.

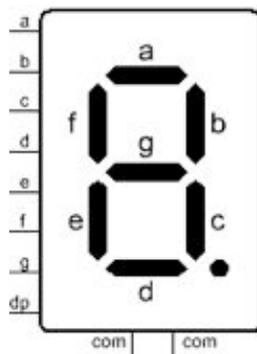
Ein einfaches Beispiel: Die Adresse a sei 3 (dezimal), also 0011 (binär). Bit 0 und Bit 1 müssen also eins sein, was wir mit AND testen. Bit 2 und Bit 3 müssen null sein, was wir mit NOR testen. Und nur wenn beide Bedingungen erfüllt sind liegt die Adresse 3 an. Dies führt zur folgenden Schaltung:



C und D sind genau dann high, wenn die angelegte Adresse 3 ist. Somit ist X genau dann low, wenn die angelegte Adresse 3 ist.

Decodieren Sie nun beispielsweise die Adressen 2, 7 oder 11 !

- Mit 4 Bits können wir alle Ziffern von 0 bis 9 binär darstellen. Entwickeln Sie eine Schaltung, die genau dann high liefert, wenn das entsprechende Segment einer 7-Segment-Anzeige aufleuchten soll:



Jedes Segment ist nichts anderes als eine LED. Das linke Bild zeigt die 'Common Cathode' - Variante: Die Ausgänge aller LED-Segmente sind zusammengeführt auf die beiden Pins mit der Bezeichnung 'com'. Damit das Segment a leuchtet muss also am Eingang a ein high-Signal anliegen, welches zudem etwa 20 mA Strom liefern sollte. Der Eingang dp bedient den Dezimalpunkt.

a muss bei den folgenden Ziffern leuchten: 0, 2, 3, 5, 6, 7, 8, 9. Das heisst andersherum, dass a low sein muss bei den Ziffern 1 und 4. Welche Schaltung liefert genau für die Binärwerte 0001 und 0100 ein low-Signal ?

Damit haben wir 7 verschiedene Aufgaben, für jedes Segment eine !

Versuche zu Speichergliedern

1. Bauen Sie ein Grund-FlipFlop nach Figur 1 aus zwei NAND-Gliedern und testen Sie die Funktionsweise
2. Bauen Sie ein RS-FlipFlop aus vier NAND-Gliedern nach Figur 2 und analysieren Sie die Funktionsweise
3. Bauen Sie einen Fangspeicher aus 4 NAND-Gliedern nach Figur 3 und überprüfen Sie die Funktionsweise
4. Das JK-FlipFlop aus dem Internet nach Figur 4 funktioniert nur beinahe !!
Bedienen Sie den Takteingang (hier C für 'clock') über den Taster. Die Werte für die Eingänge J und K holen Sie von der Input-Einheit. Verwenden Sie das oberste NAND mit der LED, um den Output Q des Speichergliedes sichtbar zu machen. Sie können auch noch Q-quer sichtbar machen, wenn Sie zusätzlich zwei Inverter einsetzen.
Testen Sie die Gesamtfunktion des Speichergliedes für alle Kombinationen der Werte von J und K.
5. Prüfen Sie die Funktionsweise eines JK-MS-FlipFlops.
 - Bedienen Sie die Eingänge J und K mit den oberen beiden Bits der Input-Einheit.
 - Bedienen Sie den Set- und Reset-Eingang mit den unteren beiden Bits der Input-Einheit. Setzen Sie zu Beginn beide Werte auf 1, es sollten ja nie Set und Reset gleichzeitig betätigt werden !
 - Bedienen Sie den Takteingang über den Taster. Bei welcher Taktflanke wechselt der Output ?!
6. Fortsetzung zu 5 :
Welche Wirkung haben offene, unbelegte Eingänge ?!
Was geschieht also, wenn Sie einfach den Ausgang Q des ersten JK-MS-Flipflop auf den Takteingang des zweiten geben und die übrigen Eingänge gar nicht beschalten ?
7. Fortsetzung zu 6 :
Schalten Sie den dritten und den vierten JK-MS wie bei 6 dazu. Was haben Sie damit gebaut ?
8. Wir arbeiten wieder mit einem einzigen JK-MS-FlipFlop. Legen Sie einen Dateneingang D von der Input-Einheit auf J und den invertierten Wert von D auf K. Welche Funktion hat jetzt unser JK-MS-FlipFlop ?

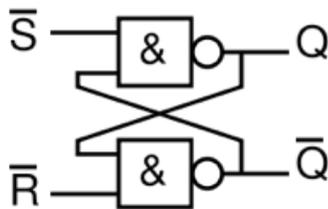


Fig. 1

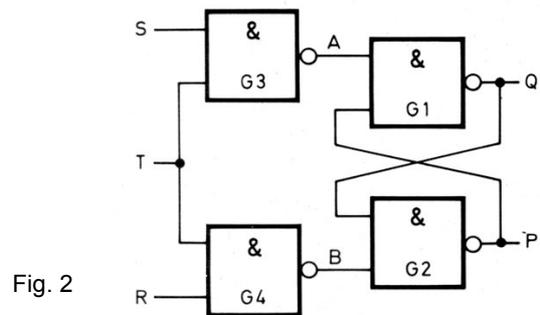


Fig. 2

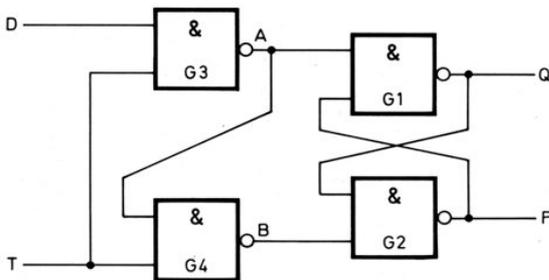


Fig. 3

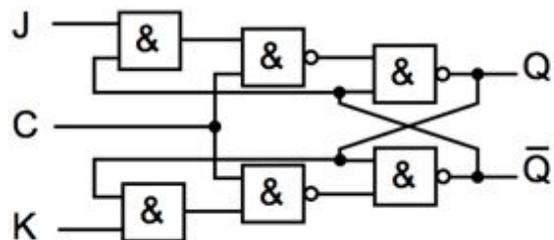
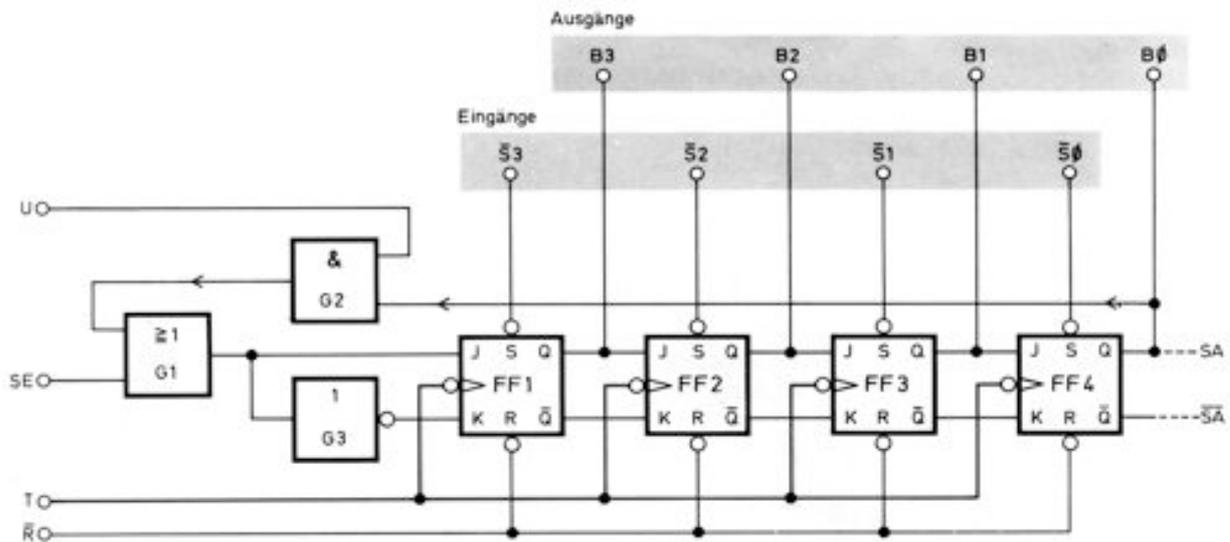


Fig. 4

Versuche zu Zählern

1. Schliessen Sie den Takteingang unseres 4-bit Binärzählers am 1 Hz Taktgeberausgang an und lassen Sie den Zähler zählen.
Benutzen Sie einen Inverter, um mit dem Taster jederzeit einen Reset des Counters auslösen zu können.
Benutzen Sie ein zusätzliches NAND-Glied, um den Zähler dazu zu bringen, dass er modulo 10 zählt: Nach 9 soll gleich wieder 0 kommen.
2. Fortsetzung zu 1 :
Bauen Sie unseren Counter um zu einem modulo 12 - Zähler (Anzeige von Stunden !) und einem Zähler, der modulo 7 zählt (ziemlich sinnlos, aber ein bisschen schwieriger ...).
3. Fortsetzung zu 1 :
Benutzen Sie zwei Input-Leitungen (eine für die low-Signale und eine für die high-Signale), um ein beliebiges Bitmuster auf die Eingänge unseres Binärzählers zu legen. Der Zählerstand wird sofort auf dieses Bitmuster gebracht, sobald der Eingang E des Counters auf 0 gezogen wird (sogenannter 'Preset' des Counters).
Bedienen Sie diesen Eingang E über eine weitere Input-Leitung.
4. Bauen Sie einen 4-bit-Binärzähler auf wie beim Versuch 7 zu den Speichergliedern. Zusätzlich sollte über den Taster (und einen Inverter ...) ein Reset dieses 4-bit-Counters ausgelöst werden können.
5. Fortsetzung zu 4 :
Bauen Sie einen 4-bit-Synchronzähler ! Beschalten Sie ihn dann so, dass er modulo 10 zählt.
6. Fortsetzung zu 4 :
Bauen Sie Ihren 4-bit-Synchronzähler um zu einem synchronen Rückwärtszähler (jetzt wieder modulo 16)
7. Fortsetzung zu 6 :
Beschalten Sie den synchronen Rückwärtszähler zusätzlich so, dass er sich nach 0 auf 9 stellt statt auf 15 (also dass er modulo 10 rückwärts zählt).
8. Fortsetzung zu 7:
Statten Sie den Rückwärtszähler noch so aus, dass er per Taster auf einen vorgewählten Wert eingestellt werden kann (Preset).
9. **** Aufwendig ****
Bauen Sie einen Oktal- oder Johnson-Zähler mit Reset (siehe Anhang 3). Analysieren Sie seine Funktionsweise.
10. **** Demo zum Baustein CMOS 4026, einem Dezimalzähler mit integriertem Decoder/Treiber für 7-Segment-Anzeigen: Samuel Hasler's "Brettli 1" mit einem vierstelligen Dezimalzähler, der mehr als eine Million Takte pro Sekunde verarbeiten könnte. Effektiv war der Timer so beschaltet, dass eine Taktfrequenz von etwa 187 kHz resultierte. Gesamtkosten dieser Elektronikbauteile: Etwa 10 Franken ...**

Versuche zu Registern



1. Bauen Sie ein 4-bit-Schieberegister gemäss obigem Bild. Ignorieren Sie vorerst die parallelen Eingänge. Benützen Sie den Taster für die Taktleitung und die oberste Inputleitung für den Reset (Grundstellung also 1). Die unterste Inputleitung soll 0 oder 1 liefern für den seriellen Eingang SE des Registers, die zweitunterste soll den Umschalter U bedienen, mit dem festgelegt wird, ob das rechts herausfallende Bit links wieder eingespeist werden soll oder nicht. Die Ausgänge können wir direkt an den LED's der JK-MS-FlipFlops beobachten.

Das Register kann nun seriell geladen werden, was 4 Takte benötigt. Sie können das Register löschen, und Sie können den Inhalt des Registers nach rechts schieben. Sie können dem Register mit U eine lineare oder eine ringförmige Struktur geben.

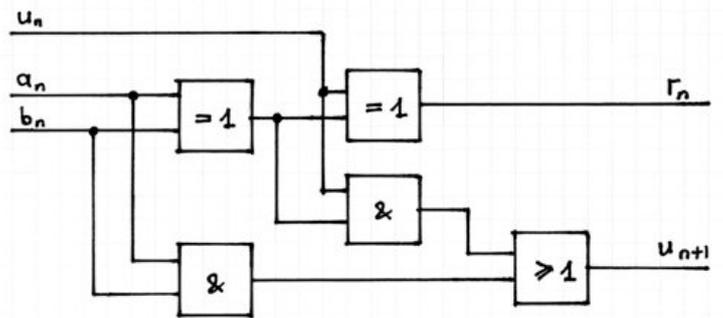
2. Fortsetzung zu 1:
Fügen Sie noch eine parallele Eingabemöglichkeit hinzu: Es steht uns ja noch eine Input-Leitung zur Verfügung. Es ist einfacher, wenn wir das Setzen der 4 Bits (nach einem Reset) durch einen 1-Impuls auslösen, da wir dann vier NAND's als **Eingangstore** verwenden können !

Müssen bei SE und bei U bestimmte Werte anliegen, damit die parallele Dateneingabe nicht verfälscht wird ?

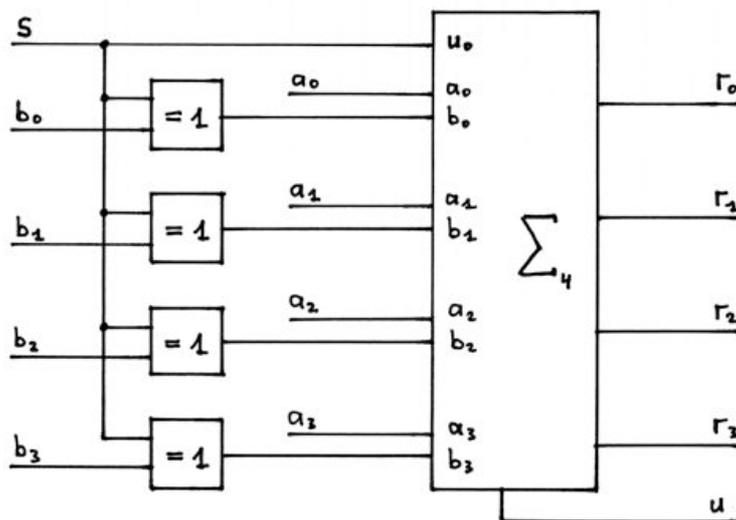
3. Was bedeutet das Rechts-Schieben des Bitmusters arithmetisch, wenn Sie dabei links eine 0 einspeisen ? Beachten Sie, dass wir bei SA ablesen können, ob rechts eine 0 rausfällt oder eine 1. Das wird für die Realisierung der Ganzzahl-Multiplikation noch von Bedeutung sein. Alle Prozessoren beherrschen diese Aufgabe, und der entsprechende Befehl ist in allen höheren Programmiersprachen und in der Maschinensprache jedes Prozessors implementiert:
 $b := a \text{ shr } 1$ bedeutet in Delphi Pascal dasselbe wie $b := a \text{ div } 2$ ('shr' für 'shift right')
 $b := a \text{ shl } 3$ bedeutet in Delphi Pascal dasselbe wie $b := a * 8$ ('shl' für 'shift left')

Versuche zur Addition und zur Subtraktion von Binärzahlen

1. Erzeugen Sie zu zwei Input-Bits das Summenbit und den Übertrag. Welche beiden Logik-Gatter verwenden Sie dazu ?
2. Bauen Sie ein Addierwerk, welches zwei 2-bit-Zahlen korrekt addiert und auch noch den Übertrag für die nächste Stelle liefert. Verwenden Sie für jedes Bit die folgende Schaltung:



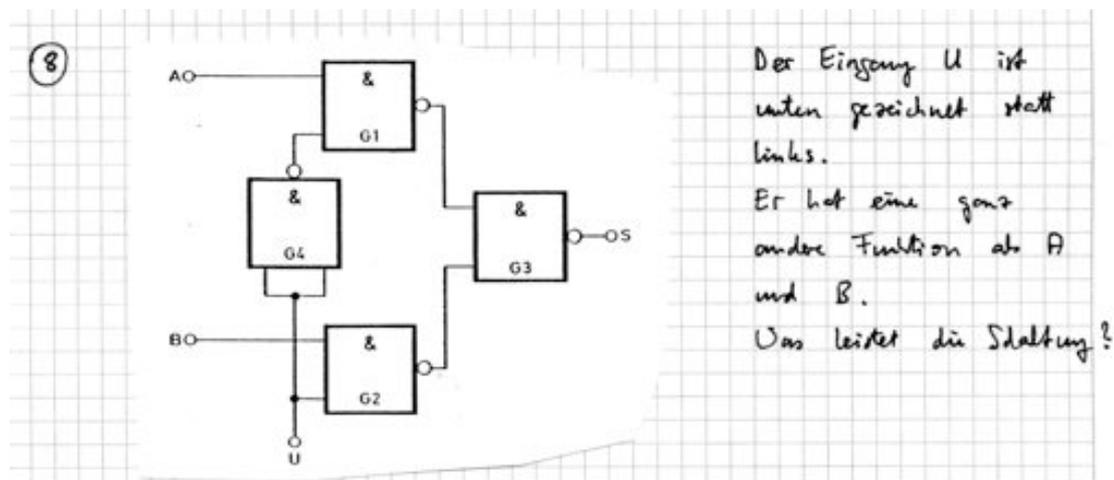
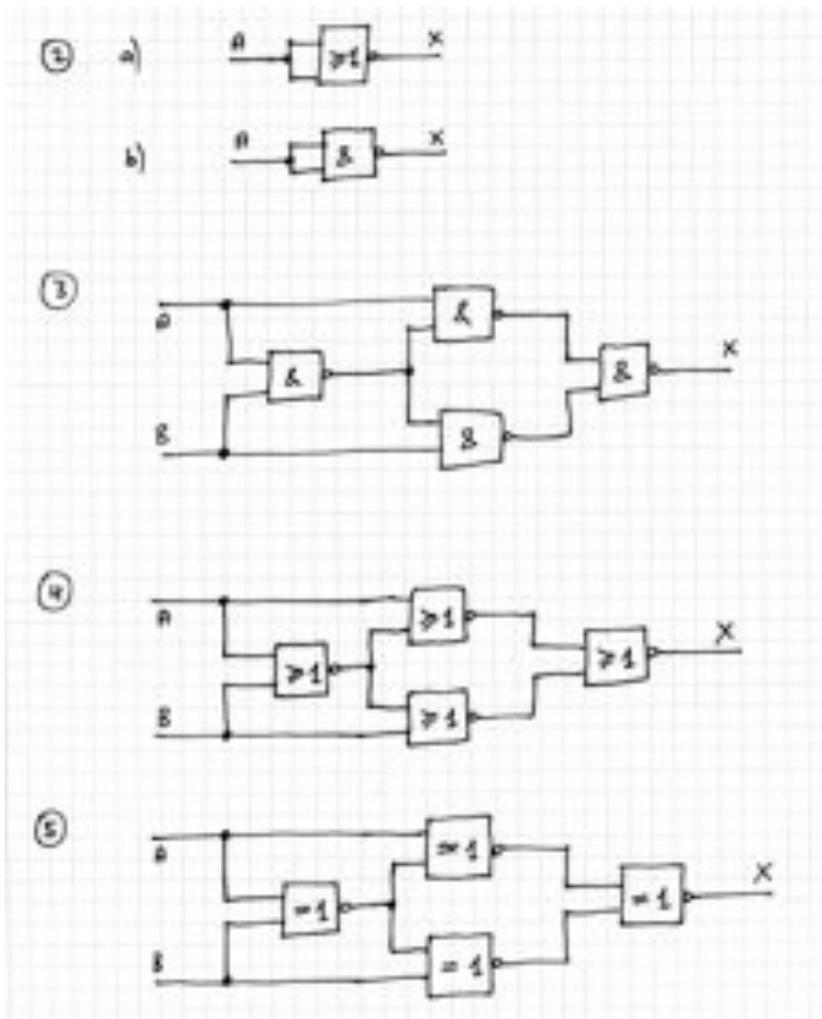
3. Für 2 Teams mit reichlich XOR- und AND-Gattern und zwei Input-Einheiten: Bauen Sie einen Volladdierer für zwei 4-bit-Zahlen. Setzen Sie die zweite Input-Einheit gleich rechts neben die erste und bauen Sie das Addierwerk in den Spalten 3, 4 und 5 des Steckbrettes auf. Benützen Sie die Preset-Funktion des 4-bit-Zählers, um das Resultat anzuzeigen !
4. Fortsetzung zu 3:
Erweitern Sie Ihren 4-bit-Volladdierer zu einem umschaltbaren Addierer/Subtrahierer (siehe Schaltbild). Der Klotz mit dem Namen Sigma4 steht für den 4-bit-Addierer, den Sie bereits aufgebaut haben. Setzen Sie das Klötzchen mit den zusätzlich benötigten vier XOR-Gattern oben in die zweite Spalte des Steckbrettes.



Bedienen Sie den Umschalter S über die 0 - Buchse des Taktgebers:

Eingesteckt setzen Sie S auf 0 und die Schaltung funktioniert als Addierer. Die XOR-Glieder verändern dann das Bitmuster von b nicht, und der nullte Übertrag ist auch 0. Lassen Sie hingegen S offen, so ist das dasselbe wie wenn Sie S auf 1 setzen. In diesem Fall liefert die Schaltung als Resultat $a - b$, da das Bitmuster von b invertiert wird und das nullte Übertragsbit gesetzt ist.

Anhang 1: Einige unserer Übungsaufgaben



Anhang 2: Materialübersicht

Jedem Zweier-Team steht das folgende Material zur Verfügung (März 2011):

- | | |
|-------------------------------------|--------------------------------------|
| 1 Steckbrett | 2 vier AND, 1x mit LED |
| 1 Taktgeber/Stromversorgung/Taster | 1 vier NAND, 1x mit LED |
| 1 Input-Einheit mit 4 LED | 1 vier OR, 1x mit LED |
| 1 4-bit-Binärzähler mit LED-Anzeige | 1 vier NOR, 1x mit LED |
| 10 Kabeli 571.21 à 4.5 cm | 2 vier XOR, 1x mit LED |
| 10 Kabeli 571.22 à 10 cm | 1 vier NOT, alle mit LED |
| 10 Kabeli 571.23 à 17 cm | 2 zwei JK-MS-FlipFlop, beide mit LED |
| 5 Kabeli 571.24 à 32 cm | |

Zwei Teams teilen sich zudem ein Netzgerät

Zusätzlich gibt es noch in einer siebten Kiste das folgende Material (März 2011):

- 5 Kabeli 571.26 à 32 cm Ministecker 2mm - Normalstecker 4mm
- 1 zwei Vierfach-AND/NAND, beide mit LED
- 1 zwei Vierfach-OR/NOR, beide mit LED
- 2 zwei JK-FlipFlop, beide mit LED
- 1 4-bit Output-Einheit mit LED
- 1 vier 2:1 Multiplexer
- 1 Buch "SIMULOG LS-TTL Teil 1: Schaltnetze und Schaltwerke"
- 1 dieses pdf-Dokument

