Basiswissen für junge Elektroniker

Logikschaltungen

Manche Geräte sollen nur dann starten, wenn alle Voraussetzungen erfüllt sind. Man nennt das eine logische Verknüpfung von Zuständen. Die Heizung fragt im Winter elektronische Temperaturfühler ab. Bei zu niedriger Außentemperatur und geringer Raumtemperatur wird sie sich einschalten. Dann zündet der Brenner und die Heizungspumpe treibt das warme Wasser durch die Rohre und Heizkörper. Meldet später der Innentemperaturfühler eine zu hohe Raumtemperatur, wird sich die Heizung von selbst wieder abschalten.

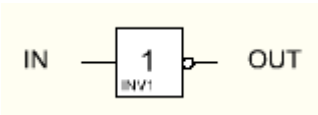
Die Temperaturfühler melden „kalt“ oder „warm“. Man kann die Zustände als logische Informationen auffassen. Kalt ist gleich eine logische Eins oder JA, warm ist eine logische Null oder NEIN. In der Elektronik wird ein Schaltelement z.B. JA als „Spannung vorhanden“ und NEIN als „Spannung nicht vorhanden“ melden oder als zugeordneten Spannungspegel ausgeben. Kann ich eine Spannung an einem Schaltungspunkt messen, dann entspricht der Zustand einer logischen Eins. Wenn nicht, dann stelle ich eine logische Null fest. Also haben wir alle denkbaren physikalischen Zustände auf 1 oder 0 reduziert. 1 bedeutet, ich kann eine Spannung messen, 0 es ist keine Spannung vorhanden. Zugegeben, das ist eine willkürliche Festlegung. Man könnte die Zustände auch umgekehrt definieren. Im ersten Fall spricht man von Positivlogig, im zweiten von Negativlogig.

So, es gibt nun viele Bauelemente, die man „logische Schaltelemente“ nennt. Wir werden uns mit den sog. CMOS-Logig-Gattern beschäftigen. Damit erhaltet ihr einen Einblick in die Welt der digitalen Logikschaltungen. Damit ihr die Bausteine nicht selbst verlöten müsst, um sie zu studieren, habe ich euch ein Board angefertigt, mit dem wir einiges testen können.

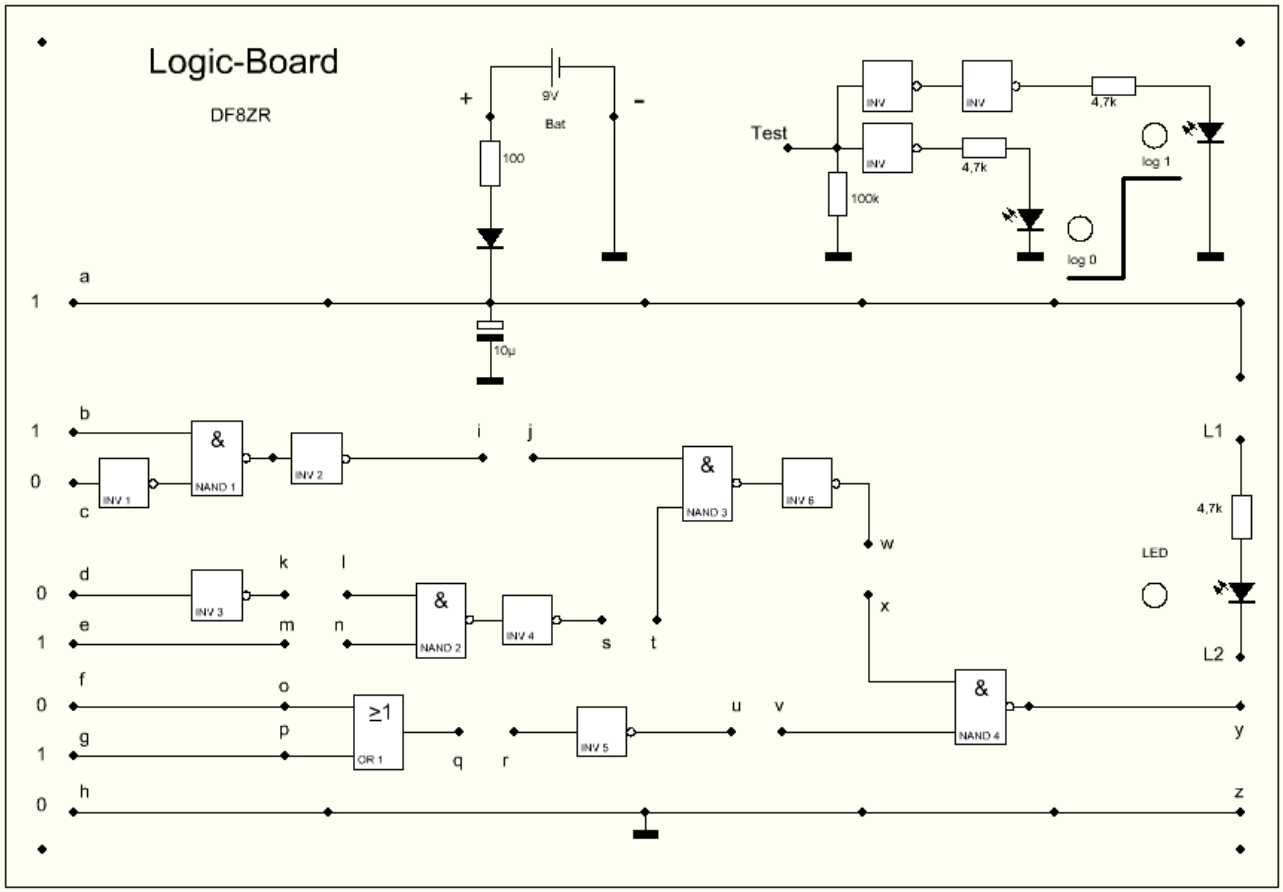
Zunächst erkennt man Symbole mit merkwürdiger Beschriftung. Die kleinen Kästchen enthalten Hinweise auf ihre logische Funktion. INV bedeutet z.B. Inverter. NAND ist eine logische UND-Verknüpfung mit folgendem Inverter. Und OR ist ein logisches ODER.

Nun habe ich euch gleich drei unbekannte Begriffe genannt. Was macht man mit diesen Funktionen?

INVERTER



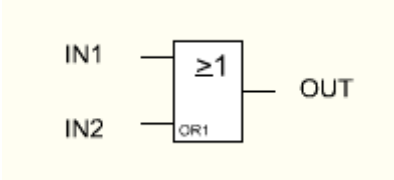
Wie die Bezeichnung andeutet, wird hier ein logisches Signal, ein logischer Zustand invertiert. Er wird umgekehrt. Eine 1 wird zur 0 und eine 0 wird zur 1. Nehmen wir an, 1 bedeutet, man kann +8V messen und 0, man misst nichts bzw. 0 Volt. So wollen wir zukünftig also die logischen Zustände in unserer Schaltung beschreiben und nachprüfen.



Zum Messen könnten wir das Multimeter einsetzen. Eleganter kann uns aber eine LED zeigen, welcher Zustand vorliegt. Dazu habe ich euch eine kleine Testschaltung installiert. Ihr seht sie oben rechts. Leuchtet die grüne LED, dann ist am Eingang eine 1, d.h. +8V. Leuchtet dagegen die rote LED, dann liegt eine logische 0 am Testeingang. Um zu prüfen, nehmen wir eine Verbindung und halten sie an den Punkt der Schaltung. Das kann ein Ausgang oder ein Eingang eines Logik-ICs sein. Die Schaltelemente, mit denen wir es hier zu tun haben, heißen logische Gatter. Wir werden gleich verstehen, warum der Begriff zutreffend ist.

OR

OR heißt deutsch ODER. Der Ausgang dieses Gatters ist 1, wenn an einem Eingang **oder** an beiden eine 1 anliegt. Er ist dann 0, wenn an keinem der Eingänge eine 1 anliegt. Man beschreibt diese Zustände in einer **Zustandstabelle**:

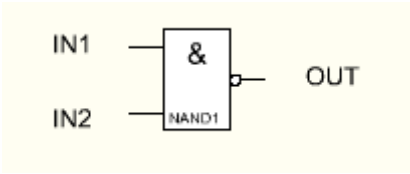


IN1	IN 2	OUT
0	0	0
0	1	1

1	0	1
1	1	1

Ein solches Gatter ist eigentlich ein offenes Gatter. Eine logische Eins kommt immer hindurch. Beim nächsten Gatter sieht die Sache aber anders aus.

NAND



Das Tor schließt, wenn an beiden Eingängen eine 1 davor steht. Wollen also zwei Besucher zugleich hindurch, dann sperrt es. Es kann immer nur einer vom Eingang zum Ausgang gehen. Eigentlich ist hier eine UND-Verknüpfung am Werk. Das Gatter prüft, ob an beiden Eingängen die Zustände gleichzeitig 1 sind. Nur in diesem Falle gibt es eine 0 am OUT heraus. Man könnte damit ein reales Tor schließen.

An beiden Eingängen eine 1 heißt aber auch, die Bedingung ist durch „1 UND 1“ erfüllt. Man könnte jetzt eine logische 1 signalisieren. Solche AND-Bausteine gibt es auch. Hier ist aber intern noch ein Inverter dahinter geschaltet. Dadurch wird dann eine 0 ausgegeben. Das N vor NAND verweist im Englischen auf: NOT AND = NAND.

Arbeiten mit dem Board

Mit diesen drei logischen Bausteinen können wir sehr komplexe logische Schaltungen herstellen. Ich werde euch nacheinander Aufgaben stellen. Durch praktische Übungen fällt es leichter, die Welt der Logikschaltungen verstehen zu lernen. Am Anfang machen wir mal was ganz Einfaches.

Versuch 1

Aufgabe: Wir verwenden nur das NAND 4. Die LED soll leuchten.

Frage: Welche logischen Signale müssen wir an die Eingänge des NANDs 4 anschalten?

Lösung: Wenn die LED leuchten soll, dann muss durch sie ein Strom fließen. Wir verbinden daher zunächst L1 mit a und L2 mit y. Da jetzt die Betriebsspannung = log 1 = ca. +8V am Vorwiderstand anliegt, kann nur Strom fließen, wenn der Ausgang OUT von NAND 4 = y das logische Signal 0 hat.

Log 0 bedeutet ja eine elektrische Verbindung zum Minuspol der Batterie.

NAND 4 ist ein AND(UND) mit Invertierung des Ausgangssignals. Das NAND gibt nur dann eine 0 heraus, wenn an seinen Eingängen x und v jeweils eine 1 ist. Wir müssen deshalb diese Eingänge

mit der log 1 mit Leitungen verbinden. Übrigens ist eine NAND zugleich auch ein ODER für die log 0! Allerdings mit der Ausgabe einer 1, eben invertiert.

Versuch 2

Wir erweitern mit zwei zusätzlichen Gattern.

Aufgabe: OR 1 wird zugeschaltet. Zuvor bitte die festen Verbindungen aufheben.

Frage: Welche Signale müssen an f und g sein?

Lösung: Bitte notiert hier die möglichen Varianten:

f g

Versuch 3

Nun bringen wir NAND 3 und INV 6 ins Spiel.

Aufgabe: NAND 3 und INV 6 richtig zuschalten.

Lösung: Die Verbindung w,x herstellen. Dann die gewählten Signale an j und t legen.

Bitte hier die logischen Signale notieren:

j t

Versuch 4

Wir schalten INV 1, NAND 1 und INV 2 hinzu.

Aufgabe: Bestimme die Signale an b und c.

Lösung: Wieder zuerst die Verbindung i,j herstellen. Danach die Verbindung an t angeschaltet lassen.

Bitte hier die logischen Signale an b und c notieren:

b c

Versuch 5

Aufgabe: Jetzt fügen wir noch die Gatter INV 3, NAND 2 und INV 4 in die Gesamtschaltung ein. Nicht vergessen, eine Verbindung von q nach v zu machen!

Frage: Welche Signale müssen an den Eingängen der Gesamtschaltung sein, damit die LED leuchtet?

Lösung: Bitte hier das Schlüsselwort notieren:

b c d e f g

Erkenntnisse

Ihr habt sicherlich bemerkt, dass man die Punkte f und g beliebig beschalten kann. Es muss nur jeweils einmal eine log 1 an einem Eingang oder an beiden sein. OR 1 ist eben ein ODER! Daher sind auch **drei** Schlüsselwörter für das Leuchten der LED zulässig. Alle anderen Signale an den Eingängen der Schaltung sind aber festgelegt. Man könnte die Schaltung dazu verwenden, eine Tür zu sichern, wenn man an Stelle der LED ein elektromagnetisches Schloss vorsieht. Nur mit dem passenden Schlüsselwort würde sich die Tür öffnen lassen.

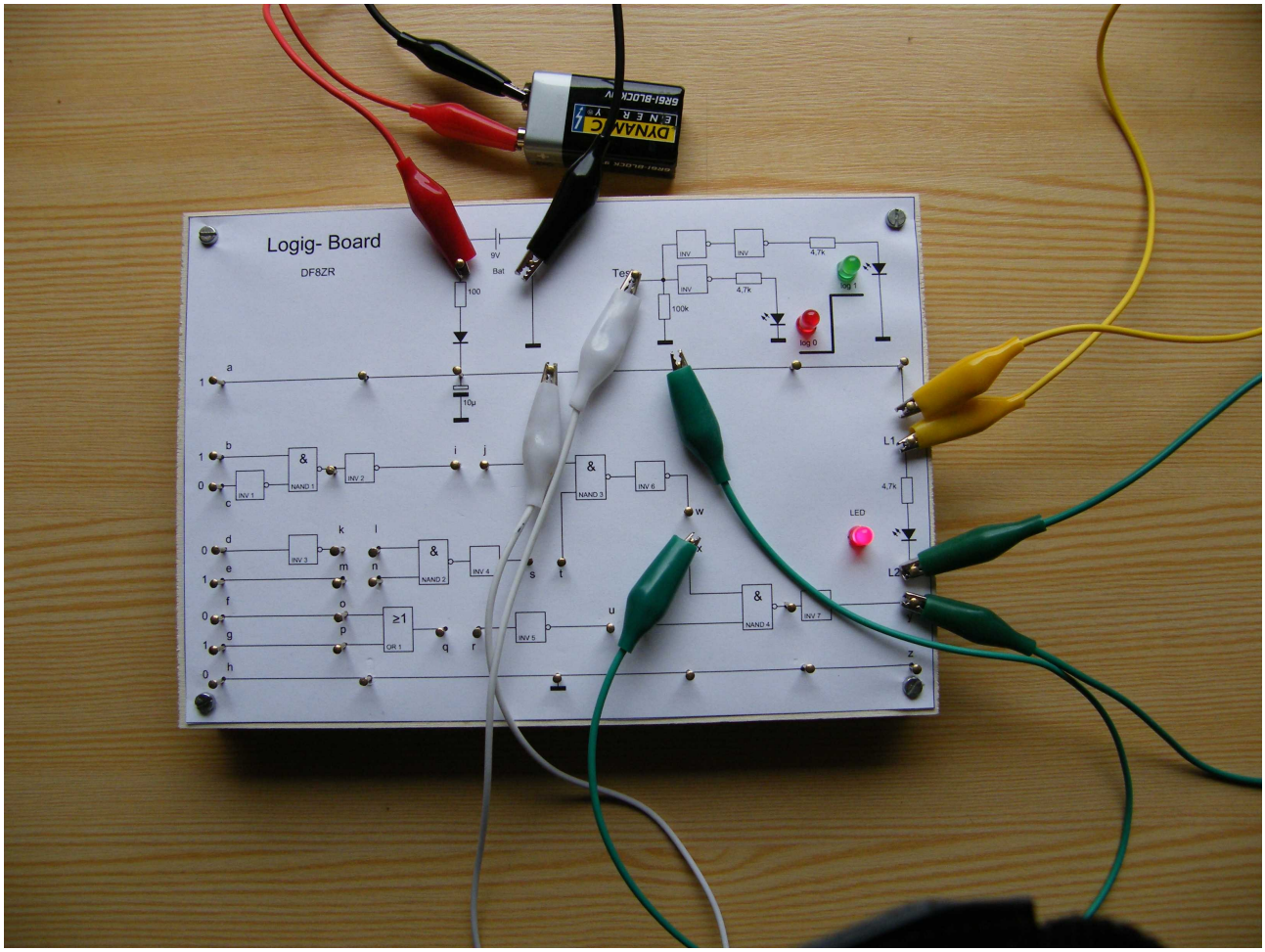
In dieser Anwendung wäre natürlich ein OR 1 nicht sinnvoll. Denn mit einem NAND an dieser Stelle wäre ein 7-stelliges Schlüsselwort sicherer als zwei mögliche mit nur 6 Stellen. eine Stelle nennt man übrigens Bit. Wir öffnen die Tür mit 7 Bits bzw. einem 7-Bit-Wort.

Mit 7 Bits lassen sich 2^7 Kombinationen einstellen. Eigentlich nur 2^7 -1, denn wenn alle Eingänge 0 sind, ist das kein Schlüsselwort. Ebenso macht es keinen Sinn, wenn alle auf 1 geschaltet werden. Also 128 -2 = 126 brauchbare Verschlüsselungen wären mit diesem Zahlenschloss anwendbar. Jemand, dem das richtige Schlüsselwort nicht bekannt ist, müsste exakt diese Einstellungen ohne Wiederholungen vornehmen, um die Tür zu öffnen. Das braucht Zeit, dennoch ist die Sicherheit zu gering. Praktisch werden heute elektronische Zugangssicherungen mit vielen tausend Kombinationen verwendet. Nur eine ist gültig!

Übungen

Wir wollen mit dem Board noch lernen, wie man Logikschaltungen analysiert. Dazu setzen wir jetzt die Test-Anzeige oben rechts ein. Verfolgen wir einmal die Signale von den Eingängen bis zur LED am Ausgang der Schaltung. Wir klemmen eine Verbindung an „Test“ an und halten das andere Ende an die verschiedenen Punkte. Auf dem Board ist die Schaltung ohne Signalzustände dargestellt. Es folgt nochmal die Kopie. Bitte tragt selbst alle logischen Pegel mit 1 oder 0 an den Punkten auf dem **Arbeitsblatt 1** ein.

Wenn ihr alles richtig verfolgt habt, erkennt ihr den Sinn solcher Prüfmittel. Man könnte im Falle von Störungen bzw. Fehlern den Zustand der Schaltung aufnehmen und in aller Ruhe nach der Ursache suchen. Solche Logik-Prüfstifte gibt es im Fachhandel zu kaufen. Sie sind ein gutes Hilfsmittel, um Reparaturen vorzunehmen. Allerdings sind sie meistens auf eine Logik-Familie abgestimmt. In unserem Brett arbeiten wir mit der CMOS-Logik. Die log 1 richtet sich nach der Betriebsspannung. Ein weiter Bereich von +3 bis +15 V ist erlaubt, während andere Familien eine feste Spannung vorsehen, z.B. ist sie bei TTL möglichst genau +5V.



Wahrheitstabellen

Zum Abschluss möchte ich noch auf den Begriff Wahrheitstabelle eingehen. Schon zu Beginn habe

ich sie als Zustandstabelle vorgestellt.

Wir können ja mal die Tabelle für das NAND 4 aufnehmen. Zunächst werden alle Verbinden vor dem Gatter getrennt. Dann legen wir die möglichen Signalkombinationen(Signal-Paare) an die Eingänge x und v und messen mit „Test“ am OUT von NAND 4. Tragt die Ergebnisse bitte hier ein:

x	v	OUT

Diese Darstellung nennt man **Wahrheitstabelle**. Für sehr komplexe Schaltungen kann die Wahrheitstabelle sehr umfangreich werden. Eine Schaltung kann auch mehrere Ausgänge haben. Übrigens kann man heute logische Schaltungen auf dem Chip eines integrierten Schaltkreises programmieren. Man braucht eine spezielle Programmiersprache, um die Funktionen einzuspeichern. Solche Bausteine sind unter der Bezeichnung FPGA = **F**ield **P**rogrammable **G**ate **A**rray im Handel. In vielen elektronischen Geräten sind sie zu finden. Die hohe Integrationsdichte erlaubt es, Schaltungen kostengünstig zu realisieren. Allerdings kommen wir bei solchen ICs nicht mehr mit unserer Testleitung an die einzelnen Punkte. Sie werden am Computer entworfen. Manche können nachträglich korrigiert werden, die einfachen allerdings nicht. Aber vor dem Start einer Massenproduktion sollte im IC selbstverständlich alles richtig angelegt sein.

Und weil es so schön geklappt hat, jetzt noch die Wahrheitstabelle eines OR. Es gibt in unserer Schaltung nur eins. Daher bitte testen und notieren:

o	p	q

Schlusswort

Hat euch der Umgang mit Logikschaltungen Spaß gemacht? Ich denke schon, denn man kann soviel damit machen. Wir können uns noch eine Weile damit beschäftigen, wenn ihr wollt. Und dabei auch den INV 5 verwenden, der bisher nicht zum Einsatz kam. Denkt euch mal eine neue Aufgabe aus und diskutiert sie mit mir. Jedenfalls werden euch in Zukunft manche Dinge aus diesem Bereich der Elektronik verständlicher sein. Und wenn ihr einmal Software-Programmierer werden wollt,

dann gehören logische Probleme zum Alltag. Je mehr ihr über Elektronik wisst, umso überzeugender könnt ihr euch bei späteren Bewerbungen um eine Lehrstelle oder Job vorstellen.

DF8ZR; 02.11.2008