

**Schriftliche Hausarbeit zur Abschlussprüfung der erweiternden Studien  
für Lehrer im Fach Informatik**

VII. Weiterbildungskurs in Zusammenarbeit mit der Fernuniversität Hagen

**Technische Informatik mit Locad2002**

Verfasser: Diplom Physiker Joachim Braun

# Technische Informatik mit Locad2002

## Inhaltsverzeichnis

1	Vorwort.....	4
2	Vorstellung von Locad2002 .....	5
3	Kurze Darstellung der Module .....	6
4	Modulfolge mit Zeitplan für verschieden Kurse.....	7
5	Ausführliche Beschreibung der Module .....	9
5.1	Grundlagen-Modul .....	9
5.2	Einfache Schaltungslogik-Modul .....	11
5.2.1	Aufgabe zur Einführung der Und-Verknüpfung sowie der Wahrheitstabelle .....	12
5.2.2	Aufgabe zur Und-Verknüpfungen mit Negation .....	14
5.2.3	Bemerkungen zu Schreibweisen.....	16
5.2.4	Arbeiten mit komplexeren Ausdrücken in Locad2002 .....	16
5.2.5	Definition über die Gleichheit von Schaltungen.....	17
5.2.6	Definition über die Gleichheit von booleschen Ausdrücken .....	18
5.2.7	Binäre Rechnungen mit Locad2002.....	18
5.2.8	Aufgabe zur Addition einer einstelligen Binärzahl zu einer zweistelligen Binärzahl.....	18
5.3	Erweiterte Schaltungslogik-Modul.....	20
5.3.1	Optimierung der booleschen Ausdrücke für eine Schaltung .....	20
5.3.2	Normalformen.....	22
5.3.3	Anwendung der De Morganschen Gesetze.....	25
5.3.4	Vorüberlegung zur Einführung von Addierern.....	26
5.3.5	Halbaddierer.....	27
5.3.6	Volladdierer .....	29
5.3.7	Paralleladdierer .....	30
5.3.8	Negative Dualzahlen.....	31
5.3.9	Sieben-Segment-Anzeige.....	33
5.3.10	ROM und PROM .....	36
5.4	NAND-Gatter-Modul .....	37
5.4.1	Transistor-Logik .....	38
5.4.2	Realisierung von Logik-Bausteinen mit NAND-Gliedern.....	42
5.5	Schnittstellen-Modul .....	44
5.5.1	Portanschluss-Baustein .....	44
5.5.2	Zweistellige 7-Segment Anzeige .....	45
6	Anhang.....	51
6.1	Bezugsquelle LOCAD 2002.....	51
6.1.1	Lizenzarten.....	51
6.1.2	Installation und Konfiguration.....	53
6.1.3	Einsatz von <i>LOCAD2002</i> unter Windows NT, 2000, XP.....	54

6.2	Das Interface.....	55
6.2.1	Hauptplatine.....	56
6.2.2	EIN-AUSGABE-Platine.....	57
6.2.3	ADDA-Platine.....	58
6.2.4	Bezugsquelle des Interface.....	59
6.3	Arbeitsblätter zu den einzelnen Kapiteln.....	60
6.3.1	Arbeitsblatt Grundlagen 1 von 2.....	61
6.3.2	Arbeitsblatt Grundlagen 2 von 2.....	62
6.3.3	Lösungen für Grundlagen 2 von 2.....	63
6.3.4	Arbeitsblatt Einfache Schaltungslogik 1 von 3.....	64
6.3.5	Lösungen Einfache Schaltungslogik 1 von 3.....	65
6.3.6	Arbeitsblatt Einfache Schaltungslogik 2 von 3.....	66
6.3.7	Lösungen zu Einfache Schaltungslogik 2 von 3.....	67
6.3.8	Einfache Schaltungslogik 3 von 3.....	68
6.3.9	Lösungen zu Einfache Schaltungslogik 3 von 3.....	69
6.3.10	Erweiterte Schaltungslogik 1 von 3.....	70
6.3.11	Lösungen zu erweiterte Schaltungslogik 1 von 3.....	71
6.3.12	Erweiterte Schaltungslogik 2 von 3.....	72
6.3.13	Lösungen zu Erweiterte Schaltungslogik 2 von 3.....	73
6.3.14	Erweiterte Schaltungslogik 3 von 3.....	74
6.3.15	Lösungen zu Erweiterte Schaltungslogik 3 von 3.....	75
6.4	Literaturverzeichnis.....	76

## 1 Vorwort

Die neuen Lehrpläne legen für Hessen einen bestimmten Durchgang für den Unterricht in Informatik fest. Neben verpflichtenden Teilen gibt es auch Wahlthema. Ein solches Wahlthema ist die Technische Informatik, welche im Jahrgang 13/2 vorgesehen ist. Diese Arbeit soll eine mögliche Behandlung von Technischer Informatik mit dem Programm Locad2002 beschreiben. Sie ist für Schüler und Schülerinnen gedacht, die Anfänger sind, und noch keine oder wenig Erfahrung mit Technischer Informatik haben.

Um eine schnelle Nutzung für den eigenen Unterricht zu ermöglichen, verwende ich zur Beschreibung eine Modultechnik. Der Stoff wird in einzelne Module zerlegt und dann beschrieben. Da einzelne Module untereinander vertauscht werden können, ergibt sich für den Leser oder die Leserin die Möglichkeit, eine geeignete Auswahl für den eigenen Unterricht zu treffen.

Am Beginn der Arbeit gebe ich Vorschläge für verschiedene Durchgänge für die Technische Informatik an. Der Leser kann dann entscheiden, ob er das für seine Lerngruppe verwenden kann. Er braucht nur die entsprechenden Module durchzulesen und vorzubereiten. Für Einheiten, welche ich selbst schon unterrichtet habe, gebe ich Möglichkeiten der Einführung vor. Bei Einheiten ohne persönliche Erfahrung, stelle ich den Sachverhalt vor und überlasse es dem Leser oder der Leserin einen Unterricht daraus zu entwickeln.

Darüber hinaus ermöglicht die Modultechnik eine einfache Erweiterung. Andere Vorschläge für Unterrichtsabläufe können problemlos integriert werden, so dass hoffentlich ein Pool von Modulen gesammelt werden kann, der interessierten Lehrern und Lehrerinnen einen schnellen Zugang zur Technischen Informatik ermöglicht.

Diese Arbeit ist von mir, bildlich gesprochen, als Stamm mit einzelnen Ästen gedacht. Es sollen weitere Äste, Blätter und sonstiges Ideen in Laufe der Entwicklung hinzukommen. Dazu hoffe ich auf die Mitwirkung von Kollegen und Kolleginnen, die Erfahrungen im Bereich der Technischen Informatik haben oder erwerben. Dazu wird diese Arbeit auf dem Bildungsserver publiziert.

## 2 Vorstellung von Locad2002

Locad ist ein Programm, mit dem man elektrische Schaltungen simulieren kann. Es beginnt bei einfachen Leitungen mit einer Lampe und endet bei selbstdefinierten IC's. Durch einfaches Anklicken einer Leitung wird diese „unter Strom“ gesetzt. Anschließend kann man die Auswirkungen dieser Aktion auf die ganze Schaltung beobachten. Die möglichen Bauteile von Locad sind:

Und	Oder
Exklusiv-Oder	Inverter
Negationspunkt	TriState-Gatter
LED	Analysepunkt
Verzögerungsglied	Schalter
Halbaddierer	Volladdierer
Flipflop (taktflankengesteuertes JK-MS-Flipflop)	
Schieberegister	Taktgeber
Multiplexer	Decoder
Parallelport-Anschluss	
Ziffernanzeigeeinheit	
Binärzähler (8 Bit, vorwärts/rückwärts)	
RAM (Random Access Memory)	
IC... (selbstdefinierte IC's)	

Es folgt ein kleines Schaltungsbeispiel mit den Schaltzeichen nach neuer DIN-Norm

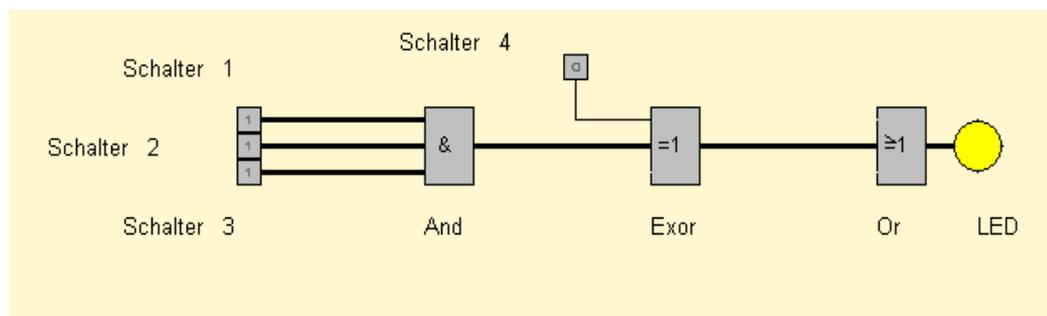


Abb. 2.1: Einfaches Schaltungsbeispiel mit And, Exor und Or

Man erkennt an den etwas dickeren Linien, dass die Schalter 1,2,3 eingeschaltet sind. Schalter 4 steht auf 0 ist also ausgeschaltet und von Schalter 4 fließt kein Strom zum Bauteil Exor. Das LED leuchtet, da es Strom erhält. Die Schüler und Schülerinnen

haben nach kurzer Zeit die Möglichkeit solche Schaltungen zu entwerfen, zu beschriften und zu analysieren z.B. mit Hilfe einer Wahrheitstabelle. Eine mögliche Frage könnte sein: "Was passiert nun, wenn Schalter 4 auch auf 1 steht, also ein Strom von Schalter 4 zum Bauteil Exor fließt?"

Man kann mit Locad2002 auch Geräte ansteuern sowie über eine Schnittstelle zwei Signale von aussen einlesen. Es handelt sich nicht um ein reines Simulationsprogramm, das nur am Rechner Verwendung finden kann, sondern den Anwendungen sind kaum Grenzen gesetzt. Ich werde darauf bei der Vorstellung der Module und den Unterrichtsvorschlägen eingehen.

### **3 Kurze Darstellung der Module**

Unter einem Modul verstehe ich eine Lerneinheit, die einen Bereich der Technischen Informatik abdeckt. Es folgen nun die einzelnen Module mit einer Kurzbeschreibung.

Grundlagen-Modul: Darunter fallen alle Fähigkeiten, die ein Schüler braucht um Locad 2002 bedienen zu können. Dies sind: Starten des Programms, Bedienung der Schaltflächen, Aufbau einer einfachen Schaltung, Speichern.

Ohne diese Grundlagen ist ein Arbeiten mit Locad 2002 nicht möglich. Deshalb muss dieses Modul bei jedem Unterricht verwendet werden, oder schon bekannt sein.

Einfache Schaltungslogik-Modul: Hierbei werden einfache Schaltungen entwickelt und mit Wahrheitstabellen ausgewertet und überprüft. Es bieten sich der Aufbau von Gattern an, die dann auf einen mathematischen Ausdruck zurückgeführt werden, oder genau umgekehrt. Dabei sollte auch das Rechnen im binären Zahlensystem mit Locad wiederholt und geübt werden. Je nach Lerngruppe kann man dabei auf verschiedenem mathematischen Niveau arbeiten.

Erweiterte Schaltungslogik-Modul: Dabei werden Halbaddierer und Volladdierer simuliert, die De Morganschen Gesetze sowohl theoretisch, als auch an Schaltungsbeispielen behandelt, Min- und Maxterme eingeführt und deren Nutzen für die Simulation besprochen.

NAND-Gatter-Modul: NAND-Gatter spielen in der Technik eine große Rolle, weil sie als Reihenschaltung zweier Transistoren einfach und billig herzustellen sind. Hier bietet es sich an, ein NAND-Gatter aufzubauen und seine Funktionsweise experimentell zu bestätigen. Mit Locad 2002 können dann Schaltungen mit NAND-Gliedern aufgebaut werden. Es kann erst eine „normale“ Schaltung mit UND-, ODER- Gliedern aufgebaut werden, die anschließend durch NAND-Gliedern zu ersetzen sind.

Schnittstellen-Modul: Es werden Signale über eine Schnittstelle am Druckerport ausgegeben. Dabei können gezielt einzelne Vorgänge ausserhalb des Computers angesteuert werden. Die Mischung zwischen realen Vorgängen und einer Simulation ist hier besonders interessant. Konkret kann man zum Beispiel eine 2-stellige 7-Segment-Anzeige über diese Schnittstelle ansteuern.

Hardware-Modul: Die Schnittstelle für die Aus- und Eingabe der Signale kann bei vorhanden Lötkenntnissen selbst gebaut werden. Dieses Modul eignet sich für eine Projektwoche oder eine Berufsschule sowie eine Computer-AG.

Jedes dieser Module wird in der Arbeit vorgestellt, Arbeitsblätter und Unterrichtsmaterialien befinden sich im Anhang.

#### **4 Modulfolge mit Zeitplan für verschieden Kurse**

Es folgen nun Vorschläge für mögliche Durchgänge bei verschiedenen Lerngruppen mit einer Abschätzung der erforderlichen Zeit. Ich gehe von 28 Stunden für den Grundkurs und 60 Stunden für den Leistungskurs aus, das entspricht 14 Unterrichtswochen im Halbjahr 13/2.

**„wenig motivierter Grundkurs“:** Die Schüler und Schülerinnen sehen in dem Fach Informatik einen von mehreren Grundkursen und zeigen kaum Interesse sich ernsthaft damit zu beschäftigen.

Grundlagen-Modul	8 Stunden
Einfache Schaltungslogik-Modul	16 Stunden
Erweiterte Schaltungslogik-Modul	4 Stunden

**„normaler Grundkurs“:**

Grundlagen-Modul	6 Stunden
Einfache Schaltungslogik-Modul	14 Stunden
Erweiterte Schaltungslogik-Modul	8 Stunden

**„starker Grundkurs“:** Die Schüler und Schülerinnen sind motiviert und interessiert. Sie wollen noch etwas aus den letzten Wochen ihrer Schullaufbahn mitnehmen.

Grundlagen-Modul	4 Stunden
Einfache Schaltungslogik-Modul	8 Stunden
Erweiterte Schaltungslogik-Modul	16 Stunden

**„normaler Leistungskurs“:**

Grundlagen-Modul	4 Stunden
Einfache Schaltungslogik-Modul	10 Stunden
Erweiterte Schaltungslogik-Modul	16 Stunden
Schnittstellen-Modul	14 Stunden
NAND-Gatter-Modul	16 Stunden

**„starker Leistungskurs“:** Hier macht es Sinn, das Hardware-Modul vor dem Schnittstellen-Modul durchzuführen. Damit können die Schüler und Schülerinnen ihre eigene Schnittstelle physisch löten und anschließend mit Locad2002 direkt ansteuern.

Grundlagen-Modul	4 Stunden
Einfache Schaltungslogik-Modul	10 Stunden
Erweiterte Schaltungslogik-Modul	16 Stunden
NAND-Gatter-Modul	8 Stunden
Hardware-Modul	10 Stunden
Schnittstellen-Modul	12 Stunden

Die vorgeschlagene Zeitplanung kann natürlich dem Unterrichtsgeschehen angepasst werden, die Abfolge der Module sollte aber beibehalten werden.

## 5 Ausführliche Beschreibung der Module

Im Abschnitt 5 werden die Module einzeln vorgestellt und eine oder mehrere mögliche Einführungen für den Unterricht angegeben.

### 5.1 Grundlagen-Modul

Die Installation erfolgt mit einer CD und läuft problemlos. Nach dem Abschluss steht Locad2002 mit einer Menü-Zeile wie in der Abbildung 5.1-1 zur Verfügung. Man erhält am oberen Bildschirmrand das Hauptmenü, darunter eine Zeile zur Einblendung von Schaltflächen, daran anschließend die Arbeitsfläche zum Aufbau von Schaltungen und am unteren Bildschirmrand eine Statuszeile zum Einblenden von Mitteilungen. Um eine Schaltung aufzubauen wählt man den Menüpunkt **Datei** und im dann erscheinenden Untermenü den Punkt **Neu** aus. Es wird eine freie Arbeitsfläche zum Aufbau einer Schaltung initialisiert und alle Hauptmenüpunkte werden aktiviert. Die Arbeitsfläche, sowie die Statuszeile am unteren Bildschirmrand habe ich ausgeblendet, um nicht eine halbe Seite mit einer leeren Fläche zu füllen.

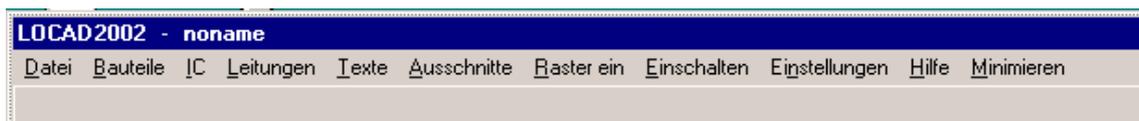
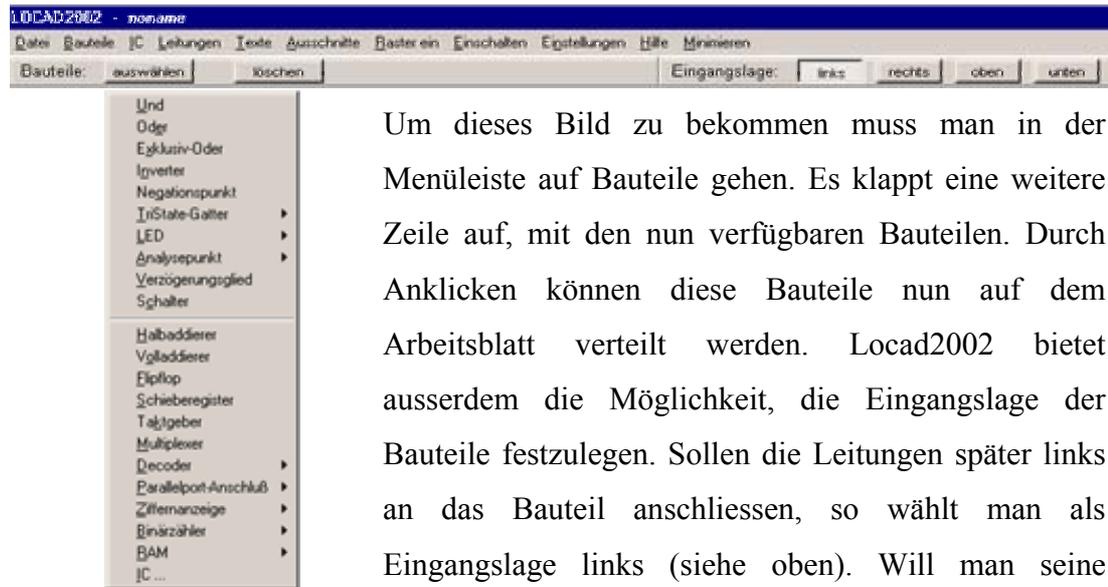


Abb. 5.1-1 Menü Zeile vom Arbeitsblatt

Welchem Zwecken die einzelnen Menüpunkte dienen, erfährt man, wenn man die F1-Taste drückt und in der daraufhin erscheinenden Inhaltsübersicht über die Hilfethemen die einzelnen Menüpunkte ("Befehle und Schaltflächen ...") mit der Maus anklickt. Um Informationen über das Hilfesystem zu bekommen, kann man auf das Fragezeichen (?) in der Menüleiste des Hilfefensters klicken und im daraufhin erscheinenden Menü den Punkt "Hilfe benutzen" anwählen. Die Bedienung erfolgt im Bezug auf das Speichern, Öffnen und Schliessen analog zu den bekannten Windows-Programmen und sollte den Schülern und Schülerinnen der heutigen geschulten und computerfreundlichen Generation keine Probleme machen.



Um dieses Bild zu bekommen muss man in der Menüleiste auf Bauteile gehen. Es klappt eine weitere Zeile auf, mit den nun verfügbaren Bauteilen. Durch Anklicken können diese Bauteile nun auf dem Arbeitsblatt verteilt werden. Locad2002 bietet ausserdem die Möglichkeit, die Eingangslage der Bauteile festzulegen. Sollen die Leitungen später links an das Bauteil anschliessen, so wählt man als Eingangslage links (siehe oben). Will man seine Leitungen von rechts in das Arbeitsblatt zeihen, so wählt man als Eingangslage rechts.

Abb. 5.1-2 Menü zum Festlegen der Bauteile

Neben dem Platzieren der Bauteile müssen die Leitungen gezogen werden. Der besseren Übersicht halber verzichtet Locad 2002 auf eine Stromversorgung für die Bauteile selbst. Man braucht also nur die Verbindungen zwischen den Bauteilen zu ziehen. Dazu wählt man den Menüpunkt Leitungen an. Anschließend den Unterpunkt Leitungen ziehen. Hilfreich ist es auch auf den Menüpunkt Rasterein zu klicken. Dadurch wird das Arbeitsblatt mit einer Punktrasterung versehen, was das Ziehen der Leitungen erleichtert. Ein solches Arbeitsblatt mit Bauteilen und schon gezogenen Leitungen wird in der folgenden Abbildung dargestellt.

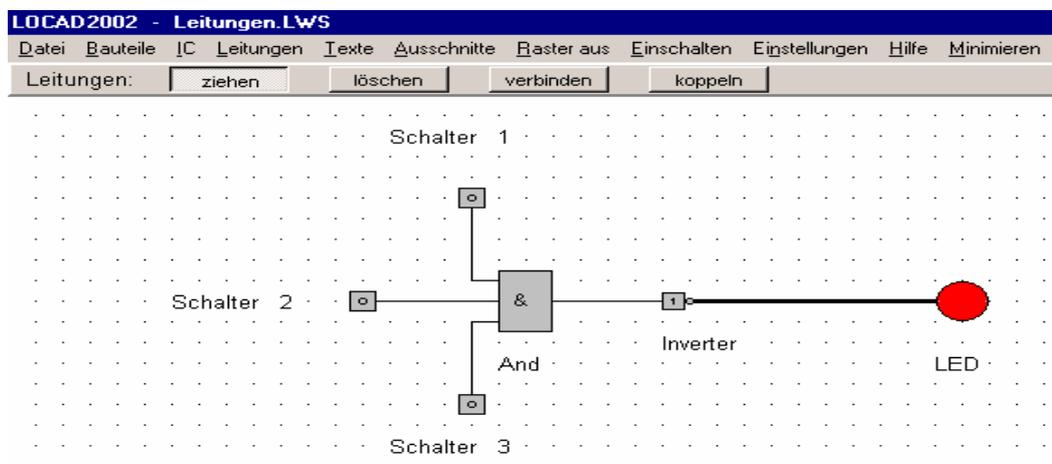


Abb. 5.1-3 Menü zum Ziehen der Leitungen

Ein Leuchten der LED wie in Abb. 5.1-3 dargestellt kann man allerdings nur erhalten, wenn man auf den Menüpunkt Einschalten geht und diesen auswählt. Hat man eine Schaltung fertiggestellt, so muss der Menüpunkt Einschalten ausgewählt werden. Jetzt kann man durch einfaches Anklicken einer Leitung für virtuellen Stromfluss sorgen. Das Gleiche passiert auch, wenn man an die offenen Enden der Leitungen jeweils einen Schalter platziert und diesen mit der Maus auswählt. Ein Platzieren der Bauteile und das Ziehen der Leitungen erfordert Übung. Es ist die Voraussetzung für ein sinnvolles arbeiten mit Locad2002 und muss deshalb so lange geübt werden, bis es problemlos klappt.

Im Anhang sind dazu zwei Arbeitsblätter vorgesehen. Diese sollten nach einer kurzen Einführung an die Schüler und Schülerinnen ausgeteilt werden. Es handelt sich um die beiden Arbeitsblätter „*Grundlagen 1 von 2*“ und „*Grundlagen 2 von 2*“ im Anhang.

Nachdem das Grundlagen-Modul behandelt wurde, sollten alle in der Lage sein Schaltungen sinnvoll zu öffnen und zu speichern, Bauteile mit der richtigen Ausrichtung zu setzen und die entsprechenden Leitungen zu ziehen. Es stehen damit alle Fähigkeiten zur Verfügung einfache Schaltungen und Gatter aufzubauen und zu untersuchen. Dies wird im folgenden beschrieben.

## **5.2 Einfache Schaltungslogik-Modul**

Das Wunder eines Computers besteht darin aus zwei simplen Zuständen nämlich Strom an und Strom aus, all die vielfältigen „Dinge“ zu erzeugen die uns umgeben. In der Regel wird kein Schüler mehr auf einer Schreibmaschine seine Bewerbung abtippen oder auf die Annehmlichkeiten des Internets verzichten. Man kann natürlich auf dem Standpunkt stehen, dass man, ähnlich wie beim Autofahren, es einfach akzeptieren und hinnehmen kann. Es funktioniert und man beschränkt sich auf das bloße Anwenden. Meiner Meinung nach sollte es aber gerade an einer allgemeinbildenden Schule dabei nicht bleiben. Ziel muss es sein, den Schülern und Schülerinnen einen Einblick zu geben, wie aus Strom an und Strom aus komplexere Logiken aufgebaut werden können, die dann auch komplexerer Problemstellungen bewältigen können. Ein erster Schritt in diese Richtung ist das folgende Modul.

Als Einführung kann man zum Beispiel folgende historisch begründete Aufgabe stellen:

### 5.2.1 Aufgabe zur Einführung der Und-Verknüpfung sowie der Wahrheitstabelle

Im Zeitalter der Industrialisierung kam es fast täglich zu Arbeitsunfällen, bei denen Arbeiter oder Arbeiterinnen mit Händen in bewegliche Teile der Maschine hineingriffen. Dies führte im schlimmsten Fall zu Amputationen und Erwerbslosigkeit. Um das zu verhindern soll eine Maschine erst dann laufen, wenn zwei Schalter gleichzeitig gedrückt sind. Damit ist sichergestellt, dass sich die Hände nicht in Gefahr befinden. Konstruiere eine Schaltung, die erst dann Strom an eine Maschine gibt, wenn zwei Schalter gedrückt sind. Die Maschine soll als LED dargestellt werden. Stelle eine Tabelle auf, die für jeden möglichen Zustand der beiden Schalter angibt, ob die Maschine läuft oder nicht.

Diese Aufgabe läuft natürlich auf eine Und-Verknüpfung hinaus und sollte zu folgendem Ergebnis führen:

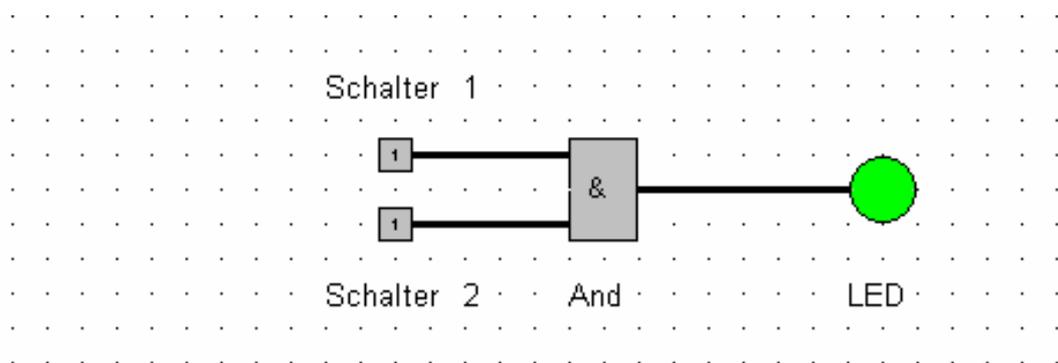


Abb. 5.2-1 Lösungsvorschlag zur Aufgabe 5.2.1

Anschließend wird eine Tabelle aufgestellt, die das Verhalten der Maschine in Abhängigkeit der Schalter darstellt. Dieses kann in Umgangssprache erfolgen.

Schalter 1	Schalter 2	Maschine (LED)
An	Aus	Aus
An	An	An
Aus	Aus	Aus
Aus	An	Aus

Abb. 5.2-2 Umgangssprachliche Tabelle zur Aufgabe 5.2.1

Der erste Kontakt mit einer Wahrheitstabelle kann ruhig auf diesem einfachen sprachlichen Niveau stattfinden. Wenn es gelingt die Wahrheitswerte in den

Vorstellungen der Schüler und Schülerinnen auf einfache Schalterstellungen zurückzuführen, ist schon viel gewonnen. Eventuelle Berührungsängste können verringert oder beseitigt werden. Es erscheint den Schülern und Schülerinnen bestimmt unzweckmäßig immer „An“ und „Aus“ zu schreiben. Also ersetzt man im Rahmen der positiven Logik alle „An“ durch 1 und alle „Aus“ durch 0. Schalter 1 und Schalter 2 werden durch A und B ersetzt und die Maschine durch A und B. Eventuell kann man hier schon die verkürzte Schreibweise für A und B einführen und AB schreiben. Die Tabelle sieht dann so aus:

A	B	A und B bzw. AB
1	0	0
1	1	1
0	0	0
0	1	0

Abb. 5.2-3 Tabelle zur Aufgabe 5.2.1

Analog geht man bei der Einführung der anderen Grundverknüpfungen vor. Ein Arbeitsauftrag könnte lauten: „Macht das für die Glieder Or, Exor, Inverter“. Natürlich wird beim Inverter nur der Wert eines Eingangskanals invertiert, also entsteht keine Tabelle wie bei den anderen Verknüpfungsgliedern. Die Schüler und Schülerinnen können hier spielerisch die Grundlagen der Technischen Informatik erfahren. Mit Hilfe eines Inverters lässt sich „nicht A“ geschrieben als „A quer“  $\bar{A}$  leicht in Locad2002 darstellen.

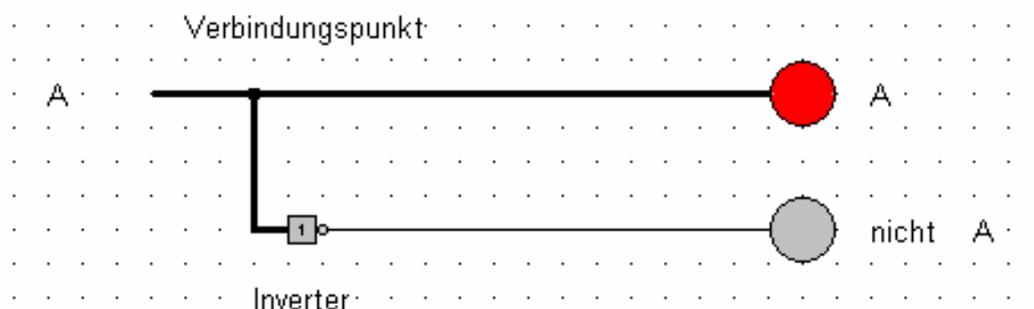


Abb. 5.2-4 Schalttechnische Umsetzung einer Negation

Wird A links durch Anklicken „unter Strom gesetzt“, so leuchtet das LED von A auf. Wird A ausgeschaltet, also die Leitung nicht mit Strom versorgt, so bewirkt der

Inverter einen Stromfluss, so dass ein LED bei nicht A aufleuchtet. Um Leitungen zu verbinden benötigt man einen Verbindungspunkt, der als kleiner quadratischer Punkt zu sehen ist. Diesen erhält man unter dem Menüpunkt Leitungen. Jetzt können beliebige Aufgaben zu diesem Themenkomplex gestellt werden.

### 5.2.2 Aufgabe zur Und-Verknüpfungen mit Negation

Onkel Dagobert will ein Geldbad in seinem Geldspeicher nehmen. Das geht natürlich nur, wenn die Panzerknacker in ihren Zellen sitzen. Ausserdem soll sein Neffe Donald nicht im Geldspeicher sein.

Entwickle eine Schaltung, die Onkel Dagobert ein Signal gibt, wann er ein Geldbad nehmen kann. Verwende dazu folgende Abkürzungen: Für die Panzerknacker „PK1“ „PK2“ und „PK3“, für Donald das Zeichen „DO“. Das Signal für Onkel Dagobert erfolgt als Leuchten einer LED. Wenn diese leuchtet ist Badezeit für Onkel Dagobert. Erstelle ausserdem eine Wahrheitstabelle. Entwickle eine Schaltung, bei der das LED immer leuchtet, wenn Dagobert nicht baden darf. Gib auch hierzu eine Wahrheitstabelle an.

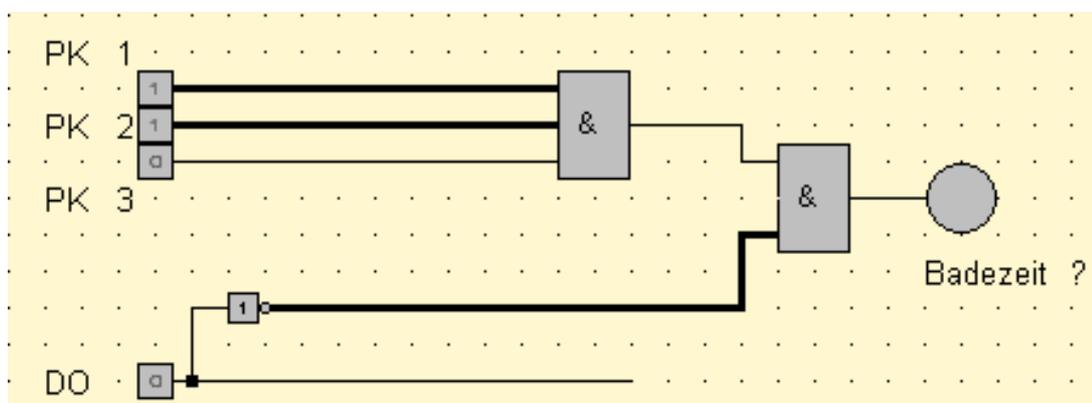


Abb. 5.2-5 Lösungsvorschlag zu Aufgabe 5.2.2

Die Schüler und Schülerinnen müssen sich über die Bedeutung von Strom an und Strom aus bei der Eingabe klar werden. Wenn bei PK 1 ein Strom fließt, könnte das zum Beispiel heissen, das Panzerknacker 1 hinter Schloss und Riegel ist. Analog kann man bei PK 2 und PK 3 vorgehen. Strom bei DO bedeutet, dass Donald noch im Geldspeicher ist, also keine Badezeit. Die Situation in Abb. 5.2-5 ist nun wie folgt zu interpretieren. Donald ist nicht da, also ist auf der Leitung DO kein Strom. Der

Inverter sorgt für einen Stromfluss an dem unteren und-Baustein. Panzerknacker 1 und 2 sind hinter Schloss und Riegel, aber Panzerknacker 3 ist frei. Onkel Dagobert darf natürlich jetzt noch nicht baden. Die geforderte Tabelle ist in der nächsten Abbildung dargestellt.

PK 1	PK 2	PK 3	DO	LED (baden?)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>
1	1	1	1	0

Abb. 5.2-6 Lösungsvorschlag zu Tabelle von Aufgabe 5.2.2

Man erkennt, dass nur in einem Fall das Baden erlaubt ist, alle Panzerknacker sind gefangen und Donald ist nicht da. Die Schwierigkeiten für Schüler und Schülerinnen sollten dabei nicht unterschätzt werden. Einige werden nicht alle Fälle in ihrer Tabelle berücksichtigen und die Frage wie viele Zeilen besetzt werden müssen führt auf die Produktregel in der Stochastik. Ein System mit vier voneinander unabhängigen Größen, die jeweils zwei innere Zustände einnehmen können, hat zwei hoch vier also sechzehn verschiedene Möglichkeiten. Der zweite Aufgabeteil lässt sich durch das Setzen eines Inverters vor dem LED lösen. Dadurch leuchtet das LED immer, wenn nicht gebadet werden kann. Ein nichtleuchtendes LED bedeutet dann Baderlaubnis.

### 5.2.3 Bemerkungen zu Schreibweisen

Je nach Lerngruppe muss man früher oder später eine mathematische Schreibweise für Verknüpfungen einführen. Spätestens beim Übergang in die erweiterte Schaltungslogik ist das zwingend erforderlich. Die Quantoren für „und“ und „oder“ sind meiner Meinung nach für die Lernenden nicht hilfreich. Das Problem sich zu erinnern, wie diese aussehen, verdeckt den Inhalt und verunsichert mehr, als das es hilft. Ein kleines Beispiel soll das verdeutlichen. Eine Lampe L soll dann leuchten, wenn der Schalter A und B oder der Schalter C gedrückt wird. Mit den üblichen Quantoren ausgedrückt:  $L = A \wedge B \vee C$ . Die Lernenden müssen jetzt wissen, dass der Quantor für und, also  $\wedge$  eine höhere Bindigkeit als der Quantor für oder hat. Der Aufwand diese formalen Sachverhalte zu klären, steht in keinem Verhältnis zum Nutzen. Eine vereinfachte Schreibweise erfüllt den Zweck genauso und liegt näher bei dem Vorwissen der Schüler und Schülerinnen. Der Quantor für und wird durch einen Punkt ersetzt, den man auch weglassen kann, der Quantor für oder durch ein Pluszeichen. Die Gleichung sieht dann so aus:  $L = AB + C$ .

Folgende Vorgehensweise macht meiner Meinung nach am meisten Sinn:

- a) Die Quantorenschreibweise vorstellen und kurz erläutern.
- b) Eine vereinfachte Schreibweise benutzen, und während des Benutzens immer wieder verdeutlichen, was dieser Ausdruck bedeutet.

### 5.2.4 Arbeiten mit komplexeren Ausdrücken in Locad2002

Nun können schwierigere Aufgabestellungen mit Locad2002 bearbeitet werden. Man gibt einen Ausdruck vor und die Lerngruppe soll eine entsprechende Schaltung entwickeln. Ein Beispiel dazu ist:  $L = \bar{A}B + AC + \bar{B}AC$

Es macht Sinn mit den Schülern und Schülerinnen festzulegen, dass Leitungen im Arbeitsblatt von Locad links beginnen und dann senkrecht verlaufen. Damit bleibt eine große Fläche für die eigentliche Schaltung übrig.

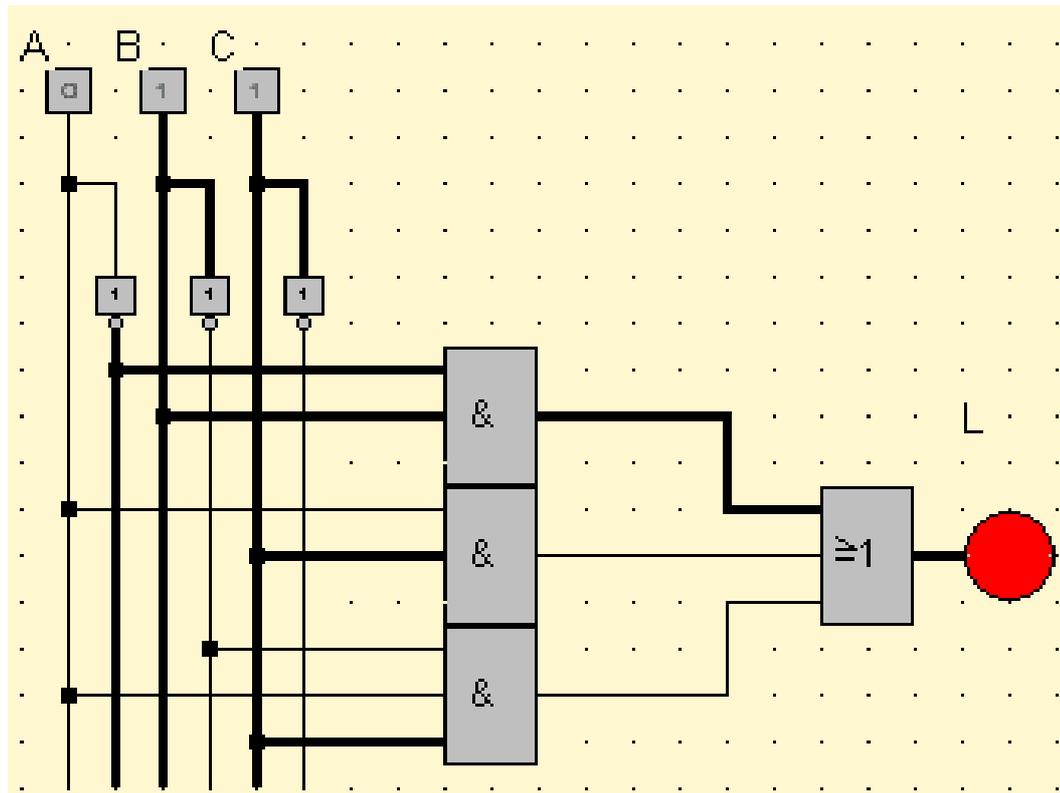


Abb. 5.2-7 Schaltung zu  $L = \bar{A} B + AC + \bar{B} AC$

Der Schalter A steht auf null, Schalter B auf eins. Damit wird  $\bar{A}$  auf eins gesetzt.  $\bar{A} B$  sind damit beide auf eins das oberste Und-Glied gibt den Strom weiter und das LED bei L leuchtet. Der letzte Term  $\bar{B} AC$  ist überflüssig, weil er schon in im zweiten Term  $AC$  enthalten ist. Das kann man auch mit einer Wertetabelle nachweisen. Die Problematik der Optimierung wir im Modul 5.3 behandelt. Man definiert dann die Gleichheit von zwei Schaltungen über die Ergebnisse in der Wertetabelle.

### 5.2.5 Definition über die Gleichheit von Schaltungen

Zwei Schaltungen sind dann gleich, wenn die beiden dazugehörigen Wahrheitstabellen für gleiche Eingangswerte immer auch gleiche Ausgangswerte liefern.

Dies Schaltung in Abb. 5.2-7 stellt einen booleschen Ausdruck dar. Es schliesst sich damit folgende Definition an.

### 5.2.6 Definition über die Gleichheit von booleschen Ausdrücken

Zwei boolesche Ausdrücke sind dann gleich, wenn die beiden dazugehörigen Wahrheitstabellen für gleiche Eingangswerte immer auch gleiche Ausgangswerte liefern.

### 5.2.7 Binäre Rechnungen mit Locad2002

Das binäre Zahlensystem ist das zu einem Computer gehörenden System. Den Schülern und Schülerinnen dürfte das zu diesem Zeitpunkt klar sein. Ein eingeschalteter Strom bedeutet eine Eins, ein ausgeschalteter Strom eine Null. Jetzt müssen Berechnungen, wie zum Beispiel eine Addition mit diesen binären Zahlen durchgeführt werden. Hierbei bietet es sich an, das binäre Zahlensystem zu wiederholen und die Addition „auf dem Papier“ auszuführen. Anschliessend überlegt man sich eine Möglichkeit mit Locad2002 das in einer Schaltung zu automatisieren. Zum Ende des Moduls über einfache Schaltungslogik, bietet es sich an, folgendes Problem zu bearbeiten. Vorher sollte das Umsetzen von Wahrheitstabellen in boolesche Ausdrücke und deren Darstellung in Locad an Beispielen geübt worden sein. Dazu sind auch zwei Arbeitsblätter im Anhang vorgesehen.

### 5.2.8 Aufgabe zur Addition einer einstelligen Binärzahl zu einer zweistelligen Binärzahl

Zu einer zweistelligen Binärzahl wird eine einstellige Binärzahl  $Z$  hinzuaddiert. Stelle eine Wahrheitstabelle auf und setze diese Tabelle in eine Schaltung in Locad um. Verwende folgende Schreibweisen Für die Binärzahl  $B_2B_1$  für die zu addierende Zahl  $Z$  und für das Ergebnis  $E_3E_2E_1$ . Die Tabelle sollte folgendermaßen aussehen:

Zeile	$B_2$	$B_1$		$Z$		$E_3$	$E_2$	$E_1$
A	0	0		0		0	0	0
B	0	0		1		0	0	1
C	0	1		0		0	0	1
D	0	1		1		0	1	0
E	1	0		0		0	1	0
F	1	0		1		0	1	1
G	1	1		0		0	1	1
H	1	1		1		1	0	0

Abb. 5.2-8 Tabelle zu Aufgabe 5.2.8

Zur Verdeutlichung soll Zeile F dienen. Zu der Binärzahl 10 wird die Zahl Z, also 1 addiert. Es ergibt sich 11. Jetzt werden für  $E_3E_2E_1$  boolesche Terme aufgestellt und diese anschließend in eine Schaltung transferiert.

Aus Zeile B folgt:  $E_1 = \bar{B}_2\bar{B}_1Z$  aus Zeile C:  $E_1 = \bar{B}_2B_1\bar{Z}$  usw. Damit ergeben sich folgende Ausdrücke für das Ergebnis:

$$E_1 = \bar{B}_2\bar{B}_1Z + \bar{B}_2B_1\bar{Z} + B_2\bar{B}_1Z + B_2B_1\bar{Z}$$

$$E_2 = \bar{B}_2B_1Z + B_2\bar{B}_1\bar{Z} + B_2\bar{B}_1Z + B_2B_1\bar{Z}$$

$$E_3 = B_2B_1Z$$

Jetzt müssen diese Ausdrücke in eine Schaltung umgesetzt werden. Man erkennt, dass zum Beispiel für  $E_2$  vier Und-Glieder und zwei Oder-Glieder verwendet werden müssen. Die Schüler und Schülerinnen werden bei der Bearbeitung dieser Aufgabe mit zwei Problemen konfrontiert. Erstens werden sie eine Vereinfachung der Terme fordern und zweitens eine Komprimierung oder Verkleinerung der Schaltungsdarstellung. Beide Forderungen können als Ansatz zum Weiterarbeiten dienen.

Es folgt eine Darstellung der Schaltung zu Aufgabe 5.2.8 ohne Optimierung der Terme und ohne „Verkleinerung“ der Schaltung als IC.

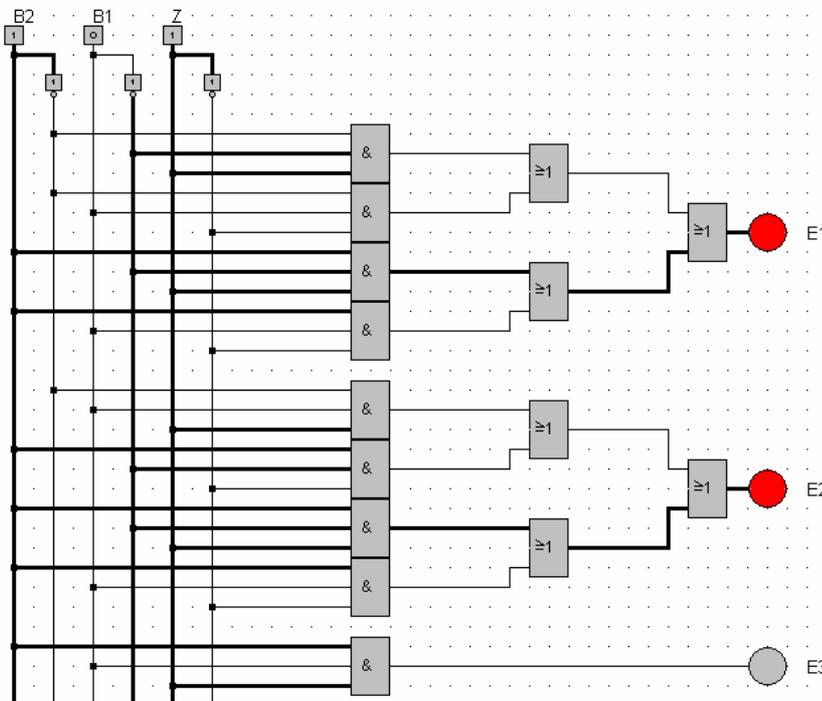
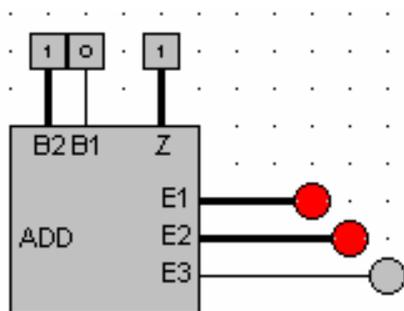


Abb. 5.2-9 Schaltung für die Addition einer einstelligen Binärzahl Z zu einer zweistelligen Binärzahl  $B_2B_1$  mit dem Ergebnis  $E_3E_2E_1$ .

Abgebildet ist die Addition von  $B_2B_1 = 10$  mit  $Z = 1$ . Das Ergebnis ist  $E_3E_2E_1 = 011$ . Diese Schaltung ist groß genug, um eine integrierte Schaltung, also ein IC einzuführen. Locad2002 bietet diese Möglichkeit. Klickt man den Menüpunkt IC an, so kann ein IC definiert werden. Dazu muss zuerst die Größe des IC's festgelegt werden. Anschließend werden die benötigten Anschlüsse an das IC gesetzt. Dabei müssen auch gleich Namen vergeben werden. Als letztes werden die einzelnen Anschlüsse den entsprechenden Leitungen in der Schaltung zugeordnet. Das zu Abbildung 5.2-9 gehörende IC kann zum Beispiel so aussehen:



Die Anschlüsse sind an den selben Stellen angebracht wie in der Schaltung 5.2-9 und das Ergebnis der Addition ist natürlich gleich. Das IC ist im Verhältnis zur Schaltung deutlich kleiner als es hier dargestellt ist. Es benötigt ein Raster von 10x10 Punkten.

Abb. 5.2-10 Das IC zur Schaltung aus Abb. . 5.2-9

Schülern und Schülerinnen wird durch diese Einführung einer integrierten Schaltung wirklich bewusst gemacht, welche Vorteile ein solches IC bietet. Es ergibt sich zwangsläufig aus dem Versuch komplexere Schaltungen aufzubauen. Damit wird der Nutzen einsichtig und die Standardfrage der Schüler: „Wozu brauchen wir denn das?“ erübrigt sich. Die zweite Möglichkeit, eine Schaltung weniger komplex zu machen liegt in deren Optimierung. Verschiedene Terme in Schaltung 5.2.-9 sind überflüssig. Jetzt folgt der Übergang zum Erweiterten Schaltungslogik-Modul indem diese Optimierungen behandelt werden.

### 5.3 Erweiterte Schaltungslogik-Modul

#### 5.3.1 Optimierung der booleschen Ausdrücke für eine Schaltung

Als Einstieg bietet sich eine Optimierung der booleschen Ausdrücken aus Aufgabe 5.2.8 an.

$$E_1 = \bar{B}_2\bar{B}_1Z + \bar{B}_2B_1\bar{Z} + B_2\bar{B}_1Z + B_2B_1\bar{Z}$$

$$E_2 = \bar{B}_2B_1Z + B_2\bar{B}_1\bar{Z} + B_2\bar{B}_1Z + B_2B_1\bar{Z}$$

$$E_3 = B_2B_1Z$$

Nehmen wir die Terme eins und drei bei  $E_1$ . Durch Ausklammern erhält man  $(\overline{B_2} + B_2)\overline{B_1}Z$ . Dieser Schritt sollte Schülern und Schülerinnen nicht schwer fallen, da sie das aus den „normalen“ Gesetzen der Algebra kennen. Was bedeutet aber nun  $(\overline{B_2} + B_2)$ ? In Worten ausgedrückt: Nicht  $B_2$  oder  $B_2$ . Eine kleine Wahrheitstabelle kann den Sachverhalt klären.

A	$\overline{A}$	$A + \overline{A}$
0	1	1
1	0	1

Abb. 5.3-1 Wahrheitstabelle zu  $A + \overline{A} = 1$

Damit wird klar, dass  $(\overline{B_2} + B_2) = 1$  ist. Nachdem die Terme zwei und vier von Gleichung  $E_1$  genauso zusammengefasst werden, ergibt sich:  $E_1 = \overline{B_1}Z + B_1\overline{Z}$ . Bei der zweiten Gleichung ergibt sich für die Terme zwei und drei der Ausdruck:  $B_2\overline{B_1}$ . Ein weiteres Zusammenfassen ist nicht möglich. Daraus folgt:

$E_2 = \overline{B_2}B_1Z + B_2\overline{B_1} + B_2B_1\overline{Z}$ .  $E_3$  bleibt gleich. Die Gleichungen für Aufgabe 5.2.8 sehen jetzt so aus:

$$E_1 = \overline{B_1}Z + B_1\overline{Z}$$

$$E_2 = \overline{B_2}B_1Z + B_2\overline{B_1} + B_2B_1\overline{Z}$$

$$E_3 = B_2B_1Z$$

Damit hat sich der Aufwand für den Aufbau einer Schaltung ungefähr halbiert. Die Schüler und Schülerinnen können jetzt die Wahrheitstabelle 5.2-8 mit den neuen Formeln überprüfen und stellen eine Übereinstimmung fest. Also sind nach Definition 5.2.6 die booleschen Ausdrücke gleich. Auf eine Darstellung der konkreten Schaltung in Locad verzichte ich aus Platzgründen. Zur Systematisierung folgen nun die Normalformen, welche an einem konkreten Beispiel erläutert werden. Es geht dabei um die Darstellung von booleschen Ausdrücken nach einem standardisierten Verfahren. Für Schüler und Schülerinnen bietet sich hierbei die Möglichkeit, wissenschaftliches Arbeiten zu erlernen und eine Vorgehensweise zur Erzeugung boolescher Ausdrücke zu erhalten. Darüber hinaus sind Normalformen ein Gegenstand des Studiums.

### 5.3.2 Normalformen

Eine Treppenhausbeleuchtung kann an- oder ausgeschaltet sein. Es gibt zwei Schalter, mit denen die Beleuchtung geschaltet wird. Die Schaltvariablen sind:

$B_0$  = Zustand der Beleuchtung vor dem Schaltprozess

$B_1$  = Zustand der Beleuchtung nach dem Schaltprozess

$S_1$  = Zustand Schalter 1

$S_2$  = Zustand Schalter 2

$B_1$  ergibt sich als Funktion von  $B_0$ ,  $S_1$  und  $S_2$ . Ist zum Beispiel  $B_0 = 0$ ,  $S_1 = 0$  und  $S_2 = 1$ , so folgt für  $B_1 = 1$ . In Worten ausgedrückt bedeutet das folgendes: die Beleuchtung ist aus, der Schalter  $S_1$  wird nicht gedrückt, der Schalter  $S_2$  wird gedrückt und damit die Beleuchtung eingeschaltet. Also ist  $B_1 = 1$ . Es ist immer wieder sinnvoll, die Bedeutung von Schaltungsvariablen verbalisieren zu lassen. Das trägt zu einem besseren Verständnis bei und koppelt eine abstrakte Schreibweise an die Realität. Auch lassen sich damit Aufgaben stellen, die ein analytisches Textverständnis trainieren. Gerade im Hinblick auf die Pisa-Ergebnisse sollte das Ausdrücken von Sachverhalten immer wieder in den Unterricht eingebaut werden.

Die komplette Tabelle für die Aufgabe lautet:

Zeile	$B_0$	$S_1$	$S_2$	$B_1$
A	0	0	0	0
B	1	0	0	1
C	0	0	1	1
D	1	0	1	0
E	0	1	0	1
F	1	1	0	0
G	0	1	1	0
H	1	1	1	1

Abb. 5.3-2 Wahrheitstabelle zur Treppenhaus-Aufgabe

Das Aufstellen der disjunktiven Normalform läuft nach folgendem Verfahren ab:

**Für alle Kombinationen, für die der Funktionswert  $B_1$  den Zustand eins hat, werden die Konjunktionsterme aufgeschrieben und diese durch ODER-Verknüpfung zusammengefasst.**

Ein **Konjunktionsterm** ist eine **UND-Verknüpfung**, die alle Schaltvariablen in negierter oder nicht-negierter Form genau einmal enthält.

So ist zum Beispiel der Term  $B_0 \bar{S}_1 S_2$  ein Konjunktionsterm der Schaltvariablen  $B_0$ ,  $S_1$  und  $S_2$ . Der Term  $B_0 S_2$  ist kein Konjunktionsterm, weil  $S_1$  nicht enthalten ist. Da jeder Konjunktionsterm nur bei einer bestimmten Belegung der Schaltvariablen den Wert eins hat, wird ein Konjunktionsterm auch **Minterm** genannt. Den Minterm, der für eine bestimmte Kombination den Wert eins hat, erhält man, indem man die UND-Verknüpfung aller vorkommenden Schaltvariablen hinschreibt und die Variablen negiert, die bei dieser Kombination den Wert Null haben. Also folgt:

$$\text{Minterm zu Zeile B: } B_0 \bar{S}_1 \bar{S}_2$$

$$\text{Minterm zu Zeile C: } \bar{B}_0 \bar{S}_1 S_2$$

$$\text{Minterm zu Zeile E: } \bar{B}_0 S_1 \bar{S}_2$$

$$\text{Minterm zu Zeile H: } B_0 S_1 S_2$$

Die Minterme werden jetzt mit ODER zusammengefasst. Es ergibt sich damit:

$$B_1 = B_0 \bar{S}_1 \bar{S}_2 + \bar{B}_0 \bar{S}_1 S_2 + \bar{B}_0 S_1 \bar{S}_2 + B_0 S_1 S_2$$

Dieser Ausdruck heisst disjunktive Normalform. Die Minterme sind disjunktiv verknüpft. Als zweite Möglichkeit zur Darstellung gibt es die konjunktive Normalform.

**Für alle Kombinationen, für die der Funktionswert  $B_1$  den Zustand 0 hat, werden die Disjunktionsterme aufgeschrieben und diese durch UND-Verknüpfungen zusammengefasst.**

Ein **Disjunktionsterm** ist eine **ODER-Verknüpfung**, die alle Schaltvariablen in negierter oder nicht negierter Form genau einmal enthält. Zum Beispiel sind  $B_0 + \bar{S}_1 + \bar{S}_2$  und  $\bar{B}_0 + \bar{S}_1 + S_2$  Disjunktionsterme der Schaltvariablen  $B_0$ ,  $S_1$ ,  $S_2$ .

Weil jeder Disjunktionsterm nur bei einer Kombination der Variablen den Wert Null, sonst immer den Wert eins hat, heisst ein Disjunktionsterm auch **Maxterm**. Den Maxterm, der für eine bestimmte Kombination den Wert Null hat, erhält man indem man die ODER-Verknüpfung aller vorkommenden Schaltvariablen hinschreibt und die Variablen negiert, die bei dieser Kombination im Zustand eins sind.

Also lauten die Terme:

$$\text{Maxterm zu Zeile A: } B_0 + S_1 + S_2$$

$$\text{Maxterm zu Zeile D: } \bar{B}_0 + S_1 + \bar{S}_2$$

$$\text{Maxterm zu Zeile F: } \bar{B}_0 + \bar{S}_1 + S_2$$

$$\text{Maxterm zu Zeile G: } B_0 + \bar{S}_1 + \bar{S}_2$$

Diese Maxterme werden nun mit UND zusammengefasst. Damit ergibt sich die konjunktive Normalform zu:

$$B_1 = (B_0 + S_1 + S_2) (\bar{B}_0 + S_1 + \bar{S}_2) (\bar{B}_0 + \bar{S}_1 + S_2) (B_0 + \bar{S}_1 + \bar{S}_2)$$

Zwischen den Klammern steht ein Punkt für die UND-Verknüpfung. Dieser wird dann in der Kurzschreibweise weggelassen.

Mit Hilfe der Normalformen wird offensichtlich, dass jede Schaltfunktion sich auf mindestens zwei Arten darstellen lässt:

1. als ODER-Verknüpfung der Minterme                      **(disjunktive Normalform)**
2. als UND-Verknüpfung der Maxterme                      **(konjunktive Normalform).**

Beide Formen können mit den Gesetzen der booleschen Algebra ineinander umgeformt werden. Dabei ist es einfacher die konjunktive Normalform in die disjunktive Normalform umzuwandeln. Man muss noch beachten, dass  $B_0 \bar{B}_0 = 0$  ist. In Worten ausgedrückt bedeutet das:  $B_0$  und nicht  $B_0$  können nicht gleichzeitig erfüllt sein. Im Arbeitsblatt (siehe Anhang) wird das Aufstellen der disjunktiven- und konjunktiven Normalform geübt. Anschließend soll deren Gleichheit gezeigt werden. Die De Morganschen Gesetze spielen in der booleschen Algebra eine wichtige Rolle. Mit Locad2002 kann man damit konkret an Schaltungen arbeiten und so bei Schülern und Schülerinnen eine Verknüpfung zwischen Theorie und Praxis erreichen. Sie stellen nach den Gesetzen der Mathematik eine Gleichung für L und nicht L auf und überprüfen mit Locad2002, ob ihre Rechnung richtig war. Dies geschieht im folgenden Unterkapitel.

### 5.3.3 Anwendung der De Morganschen Gesetze

Mit Hilfe dieser Gesetze können boolesche Ausdrücke negiert werden. Es ist hilfreich zuerst einmal diese Gesetze an Beispielen zu üben. Eine Wahrheitstabelle für den Ausdruck  $L = AB + C$  wird aufgestellt. Dabei wird auch  $\bar{L}$  mit in die Tabelle aufgenommen, Es ergibt sich folgendes:

A	B	C	L	$\bar{L}$
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	1	0	1	0
1	1	1	1	0

Abb. 5.3-3 Wahrheitstabelle zu  $L = AB + C$

Den Ausdruck für  $\bar{L}$  erhält man durch die De Morganschen Gesetze als:

$\bar{L} = \overline{AC} + \overline{BC}$ . An der Tabelle 5.3-3 kann die Richtigkeit dieser Formel überprüft werden. Anschließend wird eine Schaltung mit Locad 2002 aufgebaut, die beide Terme enthält. Die Schüler und Schülerinnen sollen dann alle Werte der Tabelle ausprobieren.

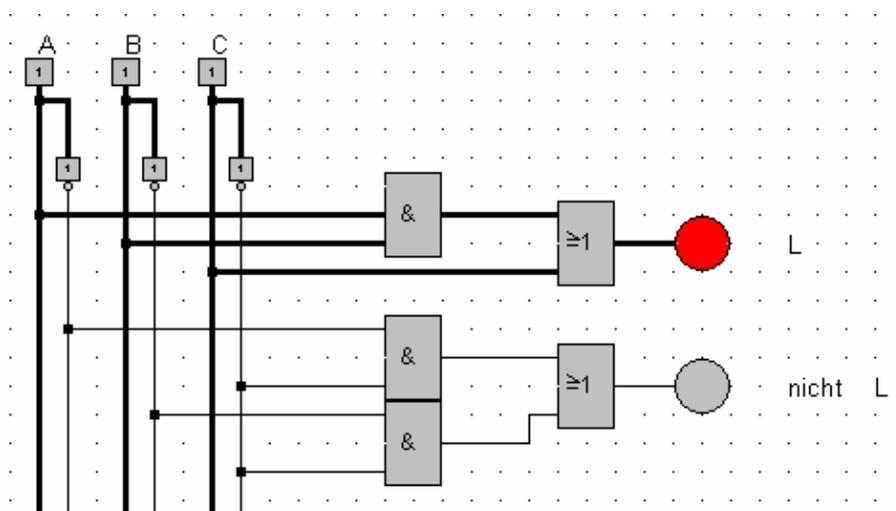


Abb. 5.3-4 Umsetzung der Ausdrücke für L und  $\bar{L}$  in eine Schaltung

Es können nun alle möglichen Kombinationen für A, B und C durchgespielt werden. Entweder leuchtet das LED bei L oder bei nicht L, aber niemals beide.

Natürlich kann man auch durch das einfache Setzen eines Inverters oder eines Negationspunktes einen Zustand negieren, also aus dem Wert für L den Wert für  $\bar{L}$  machen, ich halte es aber für besser so vorzugehen wie oben beschrieben, da die Schüler und Schülerinnen einen Schaltungsaufbau durchführen müssen und dabei ihre „abstrakten“ Terme konkret in einer Schaltung umsetzen.

#### 5.3.4 Vorüberlegung zur Einführung von Addieren

Wir greifen das Thema „Addition“ von Binärzahlen noch einmal auf und erarbeiten uns die Schaltpläne für einen Halb- und einen Volladdierer. Um die Wichtigkeit dieser Addierer bei den Schülern und Schülerinnen verständlich zu machen, helfen folgende Überlegungen.

Wie lassen sich Multiplikation, Division und Subtraktion auf eine Addition zurückführen?

Eine Multiplikation kann leicht durch eine Addition ersetzt werden. Zum Beispiel ist  $5 * 3 = 5 + 5 + 5$ . Für eine Division gilt: Die Vorkommazahl bei  $12 / 4$  ist gleichbedeutend mit der Frage nach der maximalen Anzahl der Vieren bei der Gleichung  $12 - 4 - 4 - \dots \geq 0$ . Eine Subtraktion lässt sich als Addition mit einer negativen Zahl darstellen. Letztendlich laufen alle Grundrechenarten für den Prozessor auf eine Addition hinaus. Rechenoperationen höherer Ordnung werden durch Rechenprogramme mit Hilfe der vier Grundrechenarten durch Reihenbildung und/oder Iterationsverfahren erzeugt. Eventuell können die Schüler und Schülerinnen in Gruppenarbeit Überlegungen zur Entwicklung höherer Rechenoperationen machen und dann gemeinsam diskutieren. Nach dieser Einführung sollte den Lernenden klar sein, dass die Rechner der heutigen Generation ohne Addition nicht funktionsfähig ist. Daraus ergibt sich zwingend die Forderung nach einem Addierwerk. Die Grundlagen dazu werden nun im nächsten Unterkapitel dargestellt.

### 5.3.5 Halbaddierer

Auf die Rechenvorschriften für Dualzahlen wird hier nicht eingegangen, diese sind im Anhang aufgeführt. Zum besseren Verständnis werden die Additionsregeln im Dualsystem angegeben:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

Der Halbaddierer soll zwei einstellige Dualzahlen addieren können. Es ergibt sich folgende Tabelle:

a	b	ü	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Die Schaltung muss für jede Dualziffer einen Eingang und für die Summe und den Übertrag jeweils einen Ausgang haben. Mit den Kenntnissen aus den vorangehenden Kapiteln, können wir diese Tabelle sofort in die folgende formale Schreibweise umsetzen:

$$\ddot{u} = ab$$

$$s = \bar{a}b + a\bar{b}$$

Der Übertrag wird somit durch eine einfache UND-Schaltung realisiert. Die Summe ist etwas komplizierter zu bilden. Unter Beachtung der Prioritäten der Operatoren ergibt sich das Schaltnetz:

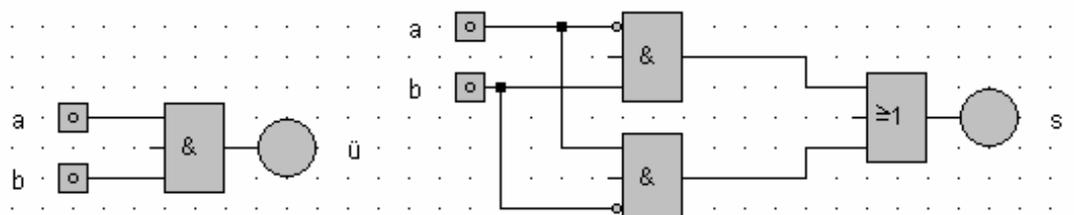


Abb. 5.3-5 Schaltungen für Übertrag und Summe

Die ausgefüllten Punkte bei den UND-Gliedern der Summe sind Negationspunkte und invertieren ein Signal.

Die beiden Teilschaltungen können zu einem Gesamtschaltnetz zusammengefasst werden. Man erhält damit eine Schaltung, die die Addition zweier Dualziffern realisiert. Da diese Schaltung nicht in der Lage ist, einen Übertrag aus einer vorhergehenden Stelle zu verarbeiten, kann sie bei der Addition von Dualzahlen auch nur an der Einerstelle eingesetzt werden. Sie hat daher den Namen *Halbaddierer*.

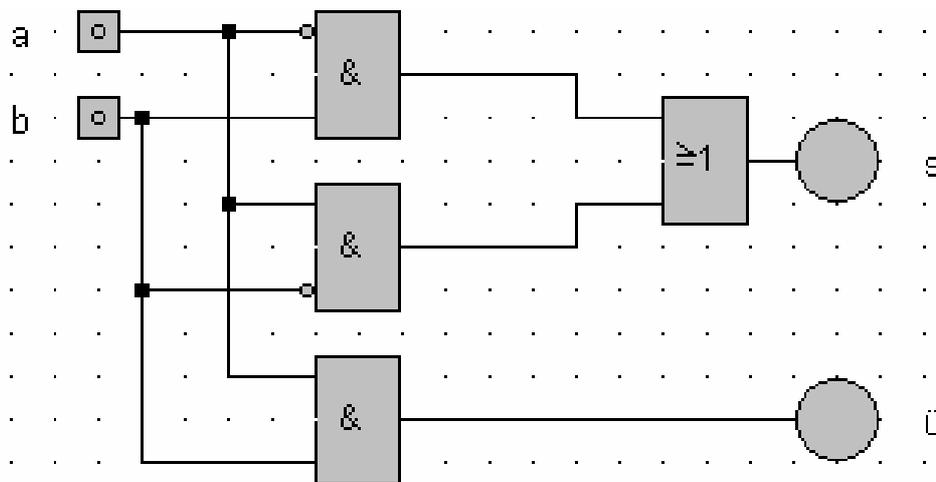


Abb. 5.3-6 Halbaddierer

Die Schüler und Schülerinnen sollen diese Schaltung aufbauen und deren Funktionsweise überprüfen.

Es gibt den Halbaddierer schon als fertiges Blockschaltymbol in der Bauteilliste von Locad2002. Dieser sieht dann so aus:

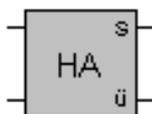


Abb. 5.3-7 Blockschaltymbol eines Halbaddierers

### 5.3.6 Volladdierer

Da bei der Addition mehrstelliger Dualzahlen ein eventueller Übertrag aus vorhergehenden Stellen berücksichtigt werden muss, brauchen wir auch eine Schaltung, die drei einstellige Dualzahlen addieren kann.

Die folgende Tabelle zeigt die benötigte Zuordnung:

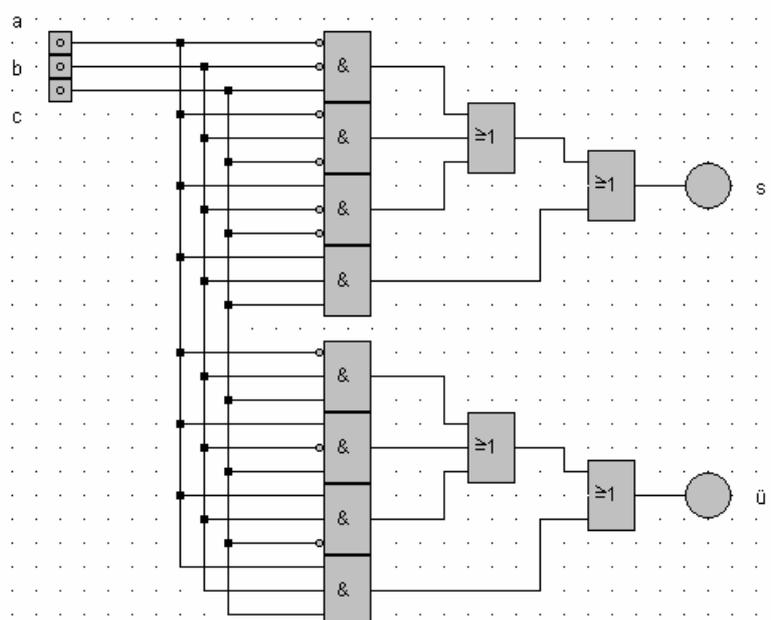
a	b	c	ü	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Hieraus lassen sich die Ausdrücke für  $\ddot{u}$  und  $s$  in disjunktiver Normalform entwickeln:

$$\ddot{u} = \bar{a} b c + a \bar{b} c + a b \bar{c} + a b c$$

$$s = \bar{a} \bar{b} c + \bar{a} b \bar{c} + a \bar{b} \bar{c} + a b c$$

Eine entsprechende Umsetzung in eine Schaltung liefert das gewünschte Schaltnetz mit dem Namen *Volladdierer*.



Auch hier könnte eine mögliche Aufgabe heißen:

Baue diese Schaltung nach und überprüfe die Funktionsweise. Führe alle möglichen Vereinfachungen durch und teste diese an dem veränderten Schaltbild.

Abb. 5.3-8 Volladdierer

Eine weitere Aufgabe lautet: Baue die folgende aus zwei Halbaddierern und einem ODER-Glied bestehende Schaltung auf und zeige, dass damit ebenfalls eine Volladdition realisiert wird.

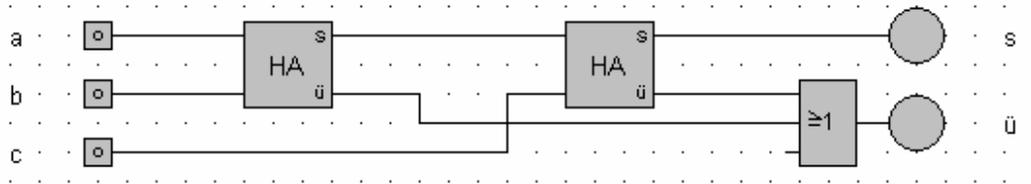


Abb. 5.3-9 Volladdierer mit zwei Halbaddierern und einem Oder-Glied

Wir werden im folgenden fertige Volladdierer-Bausteine verwenden. Diese stehen im Menü *Bauteile* zur Verfügung.

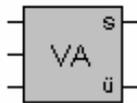


Abb. 5.3-10 Blockschaltymbol eines Volladdierers

### 5.3.7 Paralleladdierer

Mit den bisher entwickelten Rechenschaltungen können wir nun einen Addierer für zwei mehrstellige Dualzahlen aufbauen. Wir wollen uns zur Verringerung des Schaltungsaufwandes auf 4 Stellen beschränken.

Für die Einerstelle wird nur ein Halbaddierer benötigt, da hier nur zwei Dualziffern zu addieren sind, für jede weitere Stelle brauchen wir einen Volladdierer. Jeder Volladdierer muss einen eventuellen Übertrag aus der vorhergehenden Stelle verarbeiten. Diese Überlegung führt zu folgendem Schaltbild.

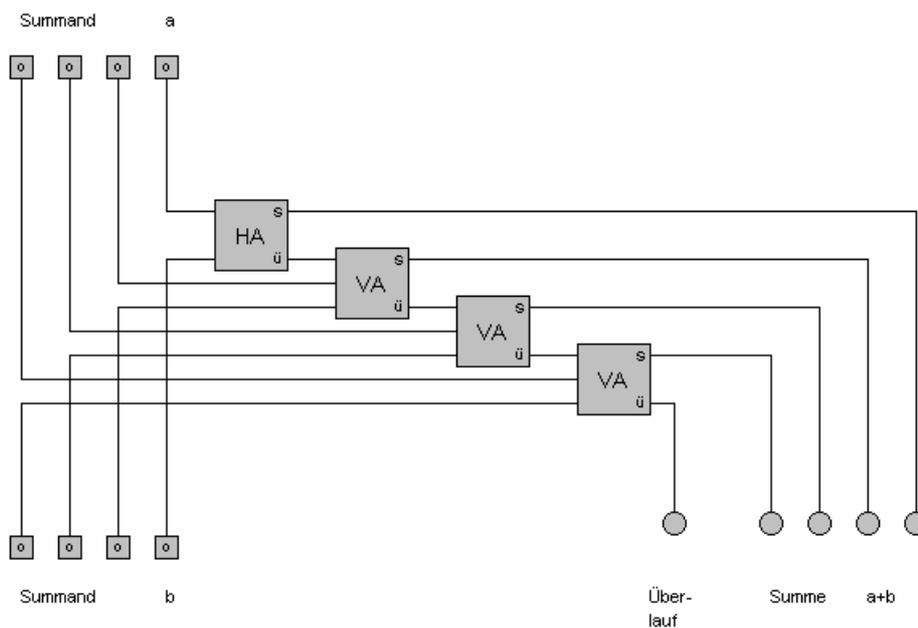


Abb. 5.3-11 Paralleladdierer

Der Name *Paralleladdierer* kommt daher, dass die Berechnung im Gegensatz zum Serienaddierer in einem Rechenschritt durchgeführt wird. Eine interessante Eigenschaft von Dualzahlen ist die Möglichkeit auch negative Zahlen darzustellen. Eventuell kennen die Schüler und Schülerinnen diesen Sachverhalt von Programmiersprachen wie zum Beispiel Delphi. Im anschließenden Kapitel wird erläutert, wie eine solchen negative Zahl dargestellt wird. Dies trägt hoffentlich dazu bei ein tieferes Verständnis zu bewirken.

### 5.3.8 Negative Dualzahlen

Um zwei Dualzahlen voneinander abzuziehen, genügt es eine normale Addition durchzuführen und für die zweite Zahl die Gegenzahl zu nehmen. Man verwendet dabei folgenden Zusammenhang:  $5 - 3 = 5 + (-3)$ . Jetzt wird „nur“ noch die Darstellung von negativen Dualzahlen benötigt. Durch eine Definition wird dies festgelegt. Steht bei der höchstwertigen Stelle einer Dualzahl eine eins, so handelt es sich um eine negative Zahl. Ein Beispiel für eine vierstellige Dualzahl soll das verdeutlichen. Der Wert von **1011** berechnet sich zu

$$-1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = -5.$$



Zahlenbereich von  $-32768$  bis  $+32767$ . Die Zweierkomplement-Bildung kann man durch den Aufbau einer Schaltung systematisieren. Je nach Leistungsstärke des Kurses, können dies die Schüler und Schülerinnen selbständig. Falls das nicht geht, bietet sich hier ein Kurzvortrag von Schülerseite oder eine gemeinsame Entwicklung der folgenden Schaltung an.

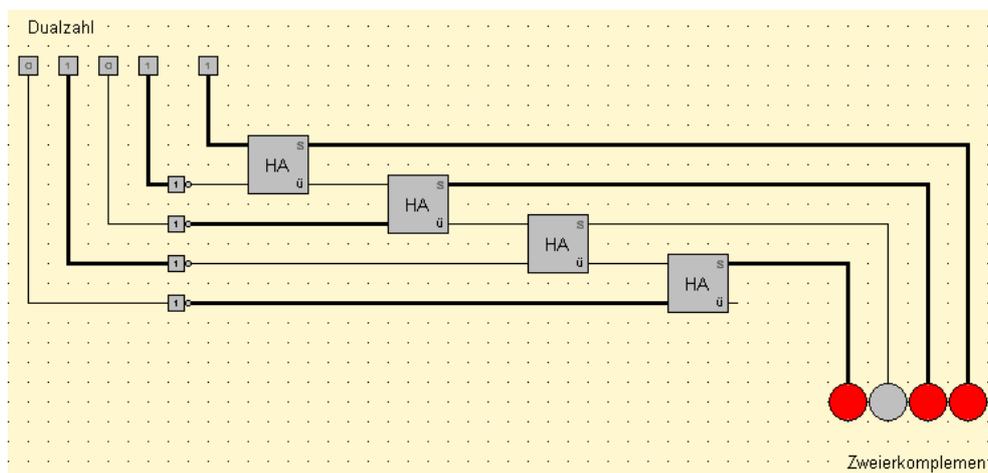
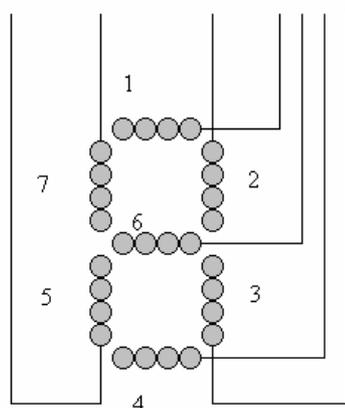


Abb. 5.3-12 Schaltung zur Zweierkomplement-Bildung

Die vierstellige Dualzahl wird links oben eingegeben. Die Inverter bewirken eine Bildung des Einerkomplements. Der fest auf eins gesetzte Schalter ist für die Addition von eins zu dem Einerkomplement zuständig. Bei den vier Halbaddierern wird der Ausgangswert bestimmt und die richtigen LED's werden angesteuert. Ein Überlauf wird hier nicht berücksichtigt. Konkret ergibt sich als Zweierkomplement zu 0101 ( 5 ) der Wert 1011 ( -5 ). Gibt man als Dualzahl eine 1111 ( 15 ) ein, so wird wegen dem fehlenden Überlauf ein falsches Ergebnis ausgegeben. Zur Darstellung von einzelnen Ziffern kann eine Sieben-Segment-Anzeige eingesetzt werden, die im folgenden beschrieben wird.



### 5.3.9 Sieben-Segment-Anzeige

Jeder kennt diesen Anzeigentyp aus eigener Erfahrung. Bei den digitalen Armbanduhren und bei vielen Messgeräten sind sie überall gegenwärtig. Die Schüler und Schülerinnen sollen eine solche Anzeige mit LED's realisieren. Dazu müssen die einzelnen Segmente je nach eingegebener Zahl angesteuert werden.

Abb. 5.3-13 Sieben-Segment-Anzeige mit LED's

Dezimal-Zahl	Segmente
0	1,2,3,4,5,7
1	2,3
2	1,2,4,5,6
3	1,2,3,4,6
4	2,3,6,7
5	1,3,4,6,7
6	1,3,4,5,6,7
7	1,2,3
8	1,2,3,4,5,6,7
9	1,2,3,4,6,7

Um eine Schaltung zur Ansteuerung der Sieben-Segment-Anzeige zu entwickeln braucht man ein weiteres Bauteil, die sogenannte Koppeldiode. Dieses Bauteil lässt den Strom nur in einer Richtung durch. Die Richtung der Spitze gibt an, in welche Richtung der Strom fließt. Man kann natürlich auch die normalen Verbindungspunkte zum „verlöten“ von Leitungen benutzen, bekommt aber dann ein Problem, das eventuell mehr Leitungen einen Stromfluss bekommen als beabsichtigt war.

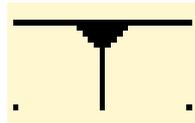


Abb. 5.3-14 Koppeldiode

Der Unterschied zwischen Koppeldioden und Verbindungspunkten wird in der folgenden Schaltung klar ersichtlich.

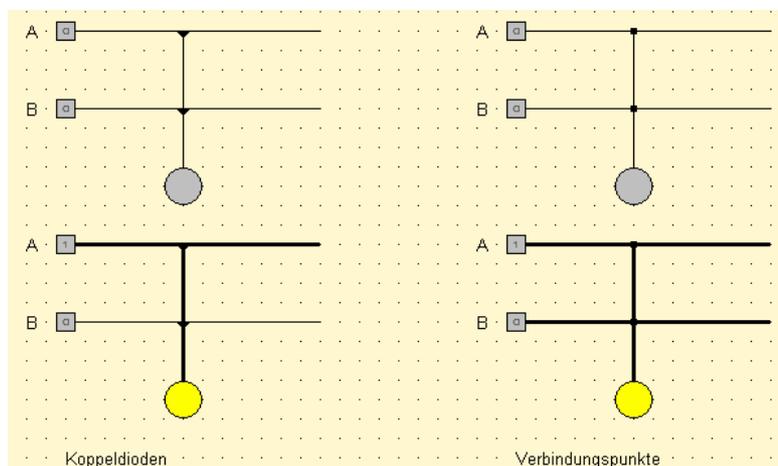


Abb. 5.3-15 Vergleich Schaltverhalten von Koppeldioden und Verbindungspunkten

Während bei den Koppeldioden nach dem Betätigen von Schalter A die Leitung von Schalter B keinen Stromfluss „spürt“, tritt bei den Verbindungspunkten ein Stromfluss auf. Damit ist nicht mehr unterscheidbar ob Schalter B gedrückt wurde, oder ob der Strom auf der Leitung von einem Betätigen des Schalters bei A herrührt.

Mit Hilfe der Koppeldioden und der Tabelle zum Ansteuern der einzelnen Segmente wird die Sieben-Segment-Anzeige konstruiert. Die LED's werden nebeneinander gesetzt und versorgen sich damit auch ohne eine Leitung mit Strom, da sie sich berühren. Das funktioniert allerdings nur mit großen LED's, da diese keine Lücke lassen.

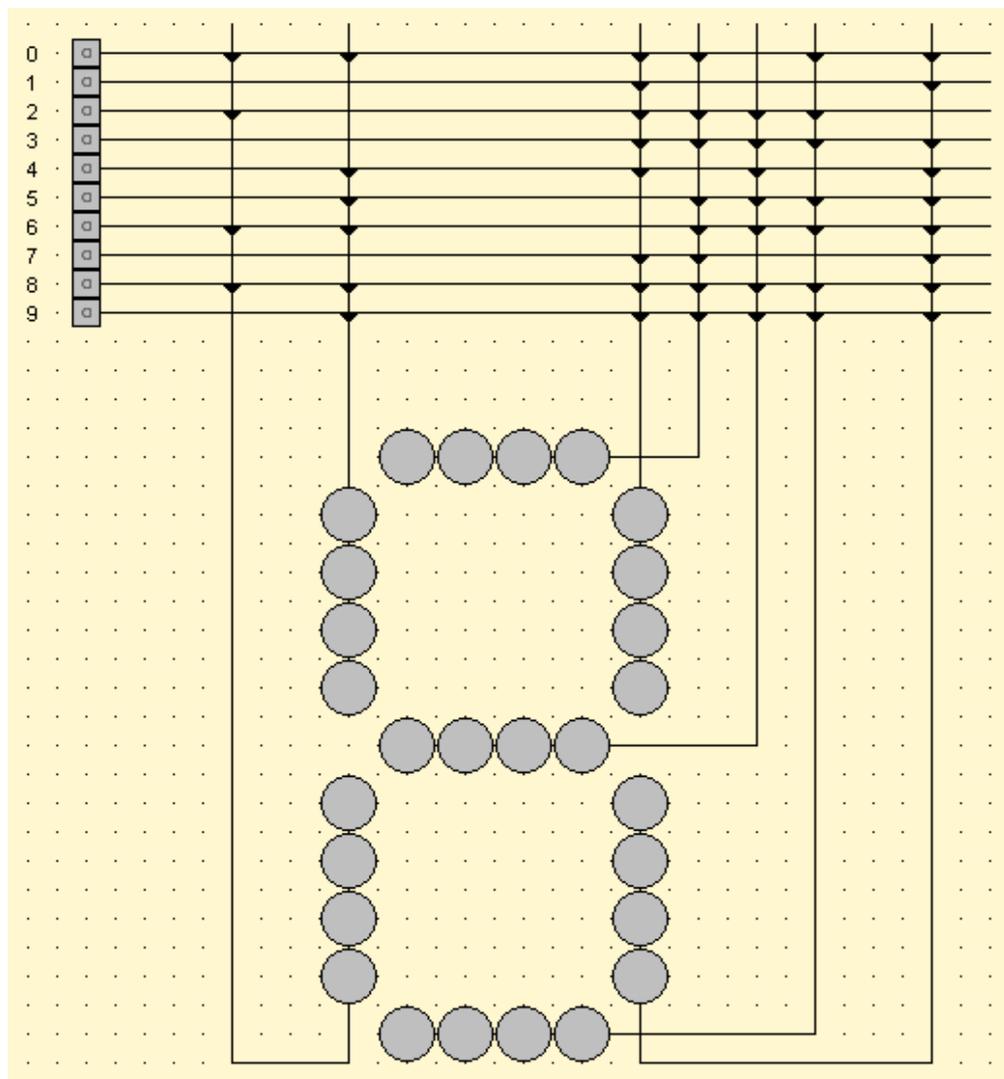


Abb. 5.3-16 Sieben-Segment-Anzeige mit Ansteuerung durch Koppeldioden

Eine mögliche Aufgabe für Schüler und Schülerinnen ist das Schrumpfen der Steuermatrix auf ein IC und die Konstruktion einer zweistelligen Sieben-Segment-Anzeige.

### 5.3.10 ROM und PROM

ROM ist die Abkürzung für **read-only-memory** (Nur-Lese-Speicher).

Es handelt sich hierbei um einen Speicher, dessen Inhalt bei der Herstellung festgelegt wird und danach nicht mehr geändert werden kann. Der Inhalt eines ROMs kann nur gelesen werden und bleibt auch bei Stromausfall erhalten. ROMs werden als Speicher für feste Programme und Daten benutzt.

PROM ist die Abkürzung für **programmable read-only-memory** (programmierbarer Nur-Lese-Speicher). Es handelt sich hierbei um einen Speicher, dessen Inhalt vom Anwender mit einem speziellen Gerät einmal festgelegt werden kann und sich danach nicht mehr ändern lässt. Nach der Programmierung verhält sich ein PROM wie ein ROM.

Die folgende Abbildung zeigt eine Möglichkeit, mit Hilfe von Koppeldioden ein PROM als Diodenmatrix zu realisieren:

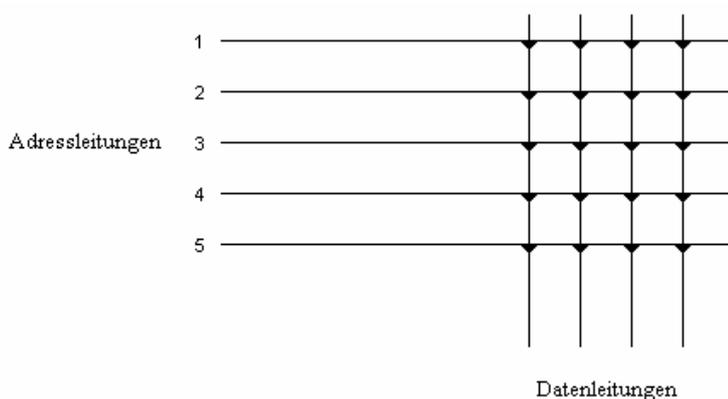


Abb. 5.3-17 PROM als Diodenmatrix

Die waagerechten Leitungen sind die Adressleitungen, die senkrechten die Datenleitungen. Allen Leitungen sind an den Kreuzungspunkten über Koppeldioden verbunden. Mit Hilfe des Programmiergerätes können nun ganz bestimmte Koppeldioden durch wohldefinierte Stromstöße „durchgebrannt“ werden. Damit ist die Verbindung der Adressleitung zu der entsprechenden Datenleitung zerstört und

ein möglicher 1-Zustand auf der Adressleitung kann sich nicht auf die Datenleitung fortsetzen. Eine zerstörte Koppeldiode repräsentiert somit eine 0, eine unbeschädigte eine 1. Wir wollen jetzt ein PROM so programmieren, dass es die ersten 4 Primzahlen als Festwerte enthält, die dann durch Anlegen eines 1-Signals an die entsprechende Adressleitung in Dualdarstellung auf den Datenleitungen erscheinen sollen. Dazu gehen wir von einer 4 x 3-Diodenmatrix aus, in der auf allen Kreuzungspunkten Koppeldioden angebracht sind. Auf Adressleitung 1 soll die erste Primzahl (2) in Dualdarstellung programmiert werden. Dazu müssen die Koppeldioden zerstört werden, die zu den Datenleitungen  $d_0$  und  $d_2$  gehören. Mit *LOCAD* wird die Zerstörung durch Löschen realisiert.

Eine entsprechende Vorgehensweise für die anderen Adressleitungen liefert schließlich folgende Matrix:

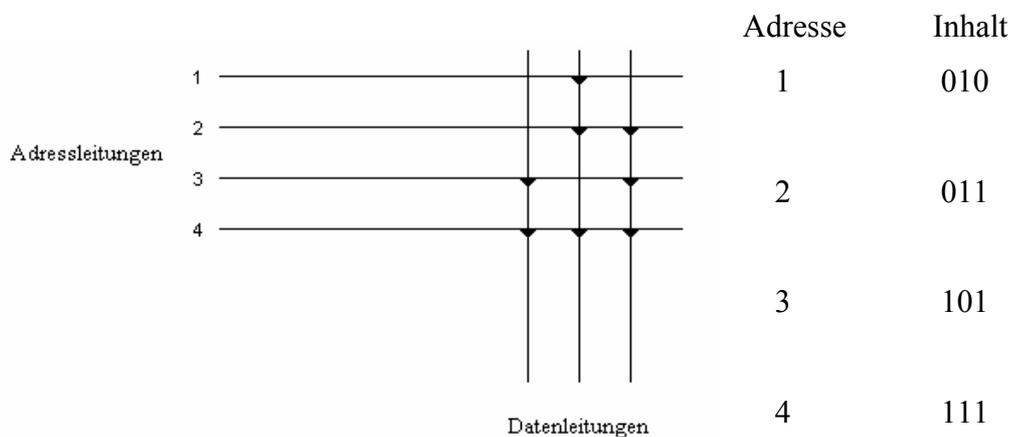


Abb. 5.3-18 4 x 3-Diodenmatrix mit den ersten 4 Primzahlen als Festwerte

Es darf immer nur eine Adressleitung auf Signal 1 gesetzt sein, weil es andernfalls zu einer Datenüberlagerung kommt. Dazu gibt es im Anhang auch ein Aufgaben-Blatt.

#### 5.4 *NAND-Gatter-Modul*

Dieses Modul eignet sich für eine fächerübergreifende Behandlungsweise. Ein möglicher Einstieg ist eine Wiederholung des Transistors als Schalter. Eventuell können auch elektronisch versierte Schüler und Schülerinnen mit einem Vortrag einen Einstieg in das neue Gebiet eröffnen. In vielen Physiksammlungen befinden sich die nötigen Bauteile zum Aufbau von Transistorschaltungen. Wenn diese Materialien vorhanden sind, könnten Schüler und Schülerinnen die einzelnen Schaltungen aufbauen und mit Messgeräten durchmessen. Der Bezug zwischen

physischen Transistoren und den entwickelten Schaltgliedern bewirkt eine Vernetzung im Denken der Lernenden und führt damit zu einem besseren Verständnis. Transistoren schalten schnell und geräuschlos, sind wartungsfrei und haben eine hohe Lebensdauer. Außerdem sind Transistoren billig und eignen sich damit zur physischen Umsetzung von Logik-Bausteinen ganz besonders.

#### 5.4.1 Transistor-Logik

Logische Schaltungen, in denen man für jeden Eingang einen Transistor verwendet, werden als Transistor-Transistor-Logik (TTL) oder einfacher als Transistor-Logik bezeichnet. Der Transistor ist ein Halbleiterbauelement, das bei entsprechender Ansteuerung als kontaktloser Schalter verwendet werden kann. Zum Verständnis der Prinzipschaltungen zur Realisierung der Grundfunktionen muss man nur diese Schalteneigenschaft des Transistors kennen.

Ein Transistor besitzt drei Anschlüsse, die als **Basis**, **Emitter** und **Kollektor** bezeichnet werden. Die folgende Abbildung zeigt das Schaltsymbol eines npn-Transistors.

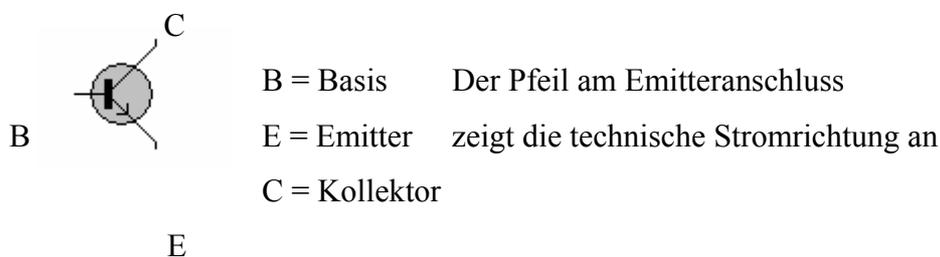


Abb. 5.4-1 npn-Transistor

Die folgenden Schaltungen sollen die Schalteneigenschaft verdeutlichen.

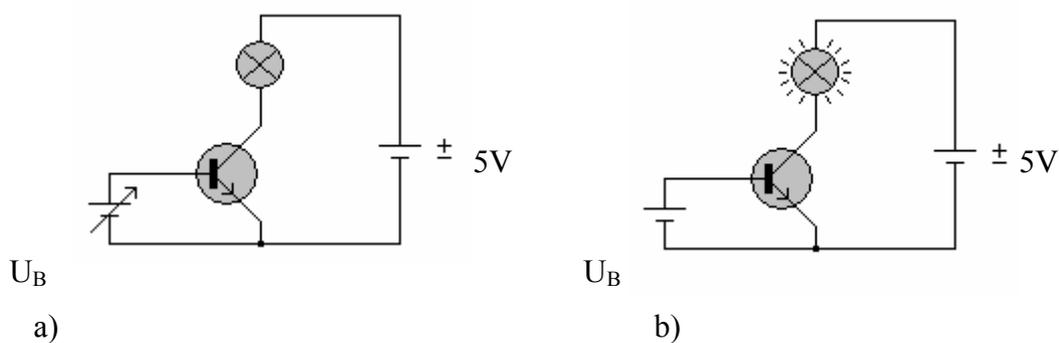


Abb. 5.4-2 Transistor als Schalter a)  $U_B < 0,6 \text{ V}$       b)  $U_B \approx 0,8 \text{ V}$

Wir beginnen mit der Abbildung a). Die regelbare Spannungsquelle  $U_B$  sei zu Beginn auf 0 V eingestellt. Die Lampe im Kollektor-Emitter-Kreis leuchtet nicht, d.h. die Kollektor-Emitter-Strecke hat einen sehr großen Widerstand. Sie verhält sich wie ein geöffneter Schalter. Man sagt, der Transistor „sperrt“.

Bei einer langsamen Erhöhung der Basis-Emitter-Spannung bleibt dieser Zustand so lange bestehen, bis die sogenannte Schwellenspannung (ca. 0,6 Volt bei Silizium-Transistoren) überschritten wird. Bei geringfügiger Überschreitung der Schwellenspannung leuchtet die Lampe mit voller Leistung. In diesem Bereich hat die Kollektor-Emitter-Strecke einen sehr geringen Widerstand. Sie verhält sich wie ein geschlossener Schalter. Man sagt, der Transistor ist „durchgeschaltet“.

Eine weitere Erhöhung der Basis-Emitter-Spannung ändert nichts an diesem Zustand. Allerdings darf diese Spannung einen vom Transistortyp abhängigen Grenzwert nicht überschreiten, da ansonsten der Transistor zerstört würde.

Mit diesen Kenntnissen lässt sich die nachstehende Schaltung analysieren. Den Binärzuständen 0 und 1 werden jetzt die auf Seite 5 erwähnten Spannungspegel der TTL-Technik zugeordnet.

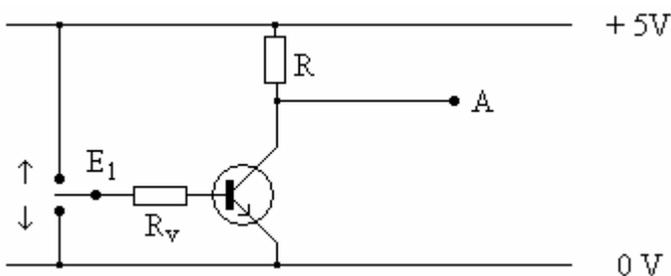


Abb. 5.4-3 Transistorschaltung

Der Eingang  $E_1$  kann über einen Schalter mit dem Minuspol ( $0V \hat{=} 0\text{-Zustand}$ ) oder mit dem Pluspol ( $5V \hat{=} 1\text{-Zustand}$ ) der Spannungsquelle verbunden werden.

Der Widerstand  $R_v$  in der Basiszuleitung ist notwendig, damit bei Verbindung mit dem Pluspol der Grenzwert für die Basis-Emitter-Spannung nicht überschritten wird. Er muss allerdings so bemessen sein, dass sich in diesem Fall eine Basis-Emitter-Spannung einstellt, die leicht über der Schwellenspannung liegt.

Wir legen den Schalter jetzt nach unten um und bringen den Eingang damit in den 0-Zustand. Zwischen Basis und Emitter liegt dann eine Spannung von 0V an. Der Transistor sperrt somit und es kann kein Strom in der Kollektor-Emitter-Leitung fließen. Damit fällt nach dem Ohmschen Gesetz  $U = R \cdot I$  keine Spannung am Widerstand R ab und der Ausgang A hat das Potential 5V, befindet sich also im 1-Zustand. Wir bringen jetzt den Eingang in den 1-Zustand, indem wir den Schalter nach oben umlegen. Die Spannung zwischen Basis und Emitter liegt dann über der Schwellenspannung und der Transistor ist durchgeschaltet. Der Widerstand der Kollektor-Emitter-Strecke ist jetzt so gering, dass das Potential des Ausgangs A nur knapp über 0V liegt und sich der Ausgang damit im 0-Zustand befindet.

Zusammenfassend haben wir folgende Zusammenhänge zwischen Ein- und Ausgang vorliegen:

$E_1$	A	Das ist die Funktionstabelle der NICHT-Verknüpfung. Die obige Schaltung stellt somit eine Realisierung der NICHT-Funktion dar.
0	1	
1	0	

Die Realisierung der UND-Funktion durch eine Reihenschaltung zweier Schalter legt die Vermutung nahe, dass eine Schaltung mit zwei in Reihe liegenden Transistoren ein ähnliches Verhalten zeigt. Wir untersuchen deshalb die nachfolgende Schaltung.

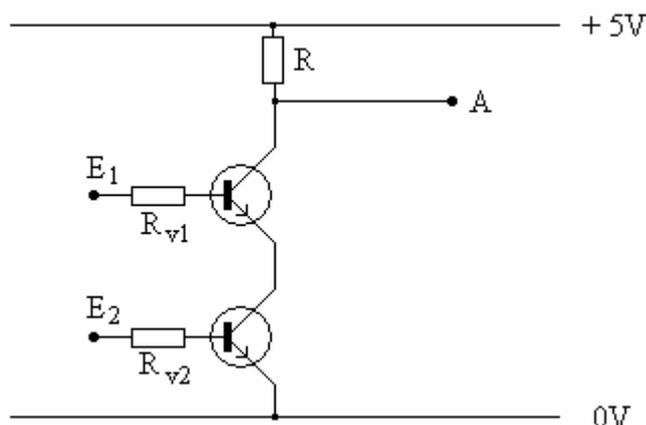


Abb. 5.4-4 Reihenschaltung zweier Transistoren

Solange kein Strom über den Widerstand R fließt, hat der Ausgang A das Potential 5V und befindet sich damit im 1-Zustand. Nur in dem Fall, dass beide Transistoren durchgeschaltet sind, liegt das Potential des Ausganges knapp über 0V ( $\hat{=}$  0-Zustand). Das führt zu folgender Funktionstabelle:

E <sub>1</sub>	E <sub>2</sub>	A		E <sub>1</sub>	E <sub>2</sub>	E <sub>1</sub> E <sub>2</sub>
0	0	1	im Vergleich dazu die Tabelle der UND-Funktion	0	0	0
0	1	1		0	1	0
1	0	1		1	0	0
1	1	0		1	1	1

Wenn wir diese Funktion der Transistorschaltung mit der UND-Funktion vergleichen, stellen wir fest, dass die obige Schaltung gerade die Negation der UND-Funktion liefert. (NICHT UND).

Wegen der einfachen technischen Realisierung wird diese Schaltung auch als eigenständiger Digital-Baustein hergestellt. Er wird als NAND-Gatter bezeichnet (gebildet aus: **NOT AND**).

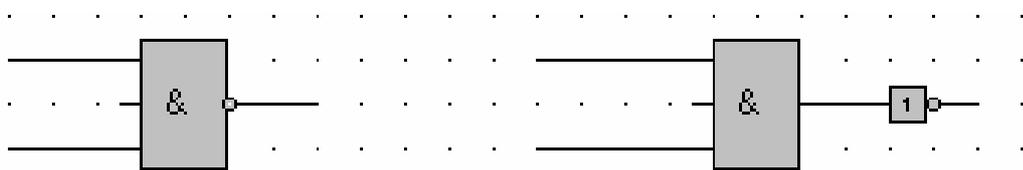


Abb. 5.4-5 Schaltsymbol eines NAND-Gatters

Beide Schaltsymbole können als NAND-Gatter verwendet werden. Der kleine Punkt am Ausgang des linken UND-Gliedes stellt einen Negationspunkt dar. Damit wird das an dieser Stelle eingehende Signal negiert. Ein Inverter, wie im Bild rechts zu sehen, macht das Gleiche.

Um aus der NAND-Schaltung eine UND-Schaltung zu erhalten, brauchen wir nur noch den Ausgang mit einer NICHT-Schaltung zu negieren. Das führt zu folgendem Schaltplan:



Eine mögliche Aufgabe für Schüler und Schülerinnen lautet: Mache dir die einzelnen Realisierungen mit Hilfe von Funktionstabellen klar.

Mit diesen Grundbausteinen lassen sich die einzelnen Schaltungen ausschließlich mit NAND-Gliedern aufbauen. Man gibt also zum Beispiel eine Schaltung vor und lässt diese dann durch NAND-Glieder ausdrücken. Aufgabe 5.2.2 kann damit folgendermaßen dargestellt werden.

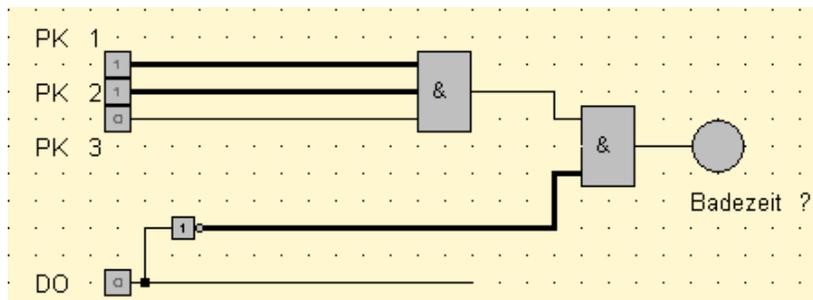


Abb. 5.4-10 Darstellung von Aufgabe 5.2.2 in bekannter Notation

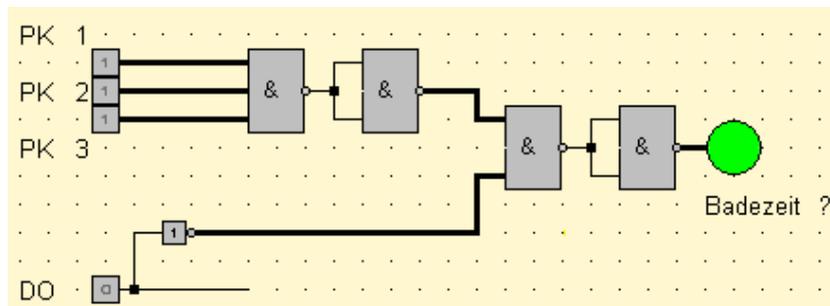


Abb. 5.4-11 Darstellung von Aufgabe 5.2.2 mit NAND-Gliedern

Im Unterschied zu Abb. 5.4-10 stehen alle drei Schalter bei PK1, PK2 und PK3 auf eins. Die Panzerknacker sind gefangen, DO steht auf Null, also ist Donald nicht im Geldspeicher. Eine günstige Zeit für ein Geldbad.

## 5.5 Schnittstellen-Modul

### 5.5.1 Portanschluss-Baustein

Der in der folgenden Abbildung gezeigte Portanschluss-Baustein ist in LOCAD2002 verfügbar und bietet die Möglichkeit, über die mit AUS 0..7 beschrifteten Anschlüsse 1-0-Signale direkt auf die parallele Druckerschnittstelle zu leiten und über die mit EIN 0..2 beschrifteten Anschlüsse Signale von dort zu empfangen.



Abb. 5.5-1 Portanschluss-Baustein

In der Regel ist die parallele Druckerschnittstelle als 25-polige SUB-D-Buchsenleiste vorhanden, deren Pin-Nummerierung in der nachstehenden Abbildung dargestellt ist.

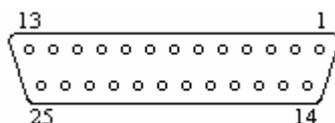


Abb. 5.5-2 SUB-D-Buchsenleiste

Die folgende Tabelle gibt die Zuordnung der Anschlüsse am Anschluss-Baustein zu den einzelnen Pins der SUB-D-Buchse wieder.

Anschluss-Baustein	SUB-D-Buchse Pin-Nummer
AUS 0	2
AUS 1	3
AUS 2	4
AUS 3	5
AUS 4	6
AUS 5	7
AUS 6	8
AUS 7	9

Die Pins 18 bis 25 sind mit Masse verbunden.

Ein 1-Signal an einem der Anschlüsse AUS 0..7 erscheint als 5V-Signal - bezogen auf die Massepins - an dem zugeordneten Pin der SUB-D-Buchse.

Ein 5V-Signal an Pin 11, 12 oder 15 erscheint an der zugeordneten Eingangsleitung des Anschluss-Bausteins als 1-Signal. Über diese drei Pins können einer Schaltung also Signale von außen zugeführt werden. Die Reaktion der Schaltung hierauf lässt sich über die Anschlüsse AUS 0..7 wiederum nach außen übertragen.

Da den Ausgängen maximal ein Strom von etwa 100 mA entnommen werden kann, ist es zwar möglich, Leuchtdioden über einen entsprechenden Vorwiderstand direkt dort anzuschließen und ein- bzw. auszuschalten, aber größere elektrische Energieverbraucher wie beispielsweise Elektromotore von LEGO oder Fischertechnik können nicht direkt betrieben werden.

Um die Ausgänge nicht zu überlasten, müssen solche Verbraucher über ein Interface an den Port angeschlossen werden.

Das Interface wurde von Eberhard Eube entwickelt. Nähere Hinweise zu diesem Interface finden Sie im Anhang.

### 5.5.2 Zweistellige 7-Segment Anzeige

Im Unterricht kann mit dem Interface eine zweistellige 7-Segmentanzeige außerhalb des Computers angesteuert werden. Die Druckerschnittstelle wird dazu benutzt, Signale aus dem Programm auszugeben. Da Windows einen Zugriff auf den Druckerport verhindert, müssen Zusatzprogramme installiert werden. Eine genaue Anleitung dazu ist im Lieferumfang von Locad2002 enthalten.

Nach dem Installieren kann die zweistellige 7-Segmentanzeige vorgegebene Werte anzeigen.

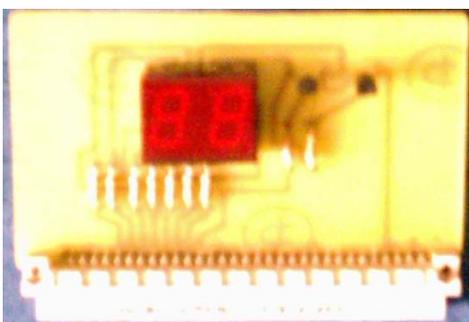


Abb. 5.5-3 Zweistellige 7-Segmentanzeige mit Anschluss an die Hauptplatine

Hier macht es Sinn zuerst eine zweistellige 7-Segmentanzeige rein auf dem Bildschirm zu entwickeln. Wie kann auf der ersten oder zweiten Stelle dieser

Anzeige eine vorgegebene Zahl erscheinen. Es wird schnell klar werden, dass ein Signal zum Umschalten der Stelle dienen muss. Man benötigt also acht Signale, um die Anforderungen zu erfüllen. Anschliessend wird die Zweistellige 7-Segmentanzeige an die Hauptplatine gesteckt. Aufgabe ist nun die Reaktion der Anzeige auf verschiedene Ansteuerungssignale zu erfassen und aufzulisten.

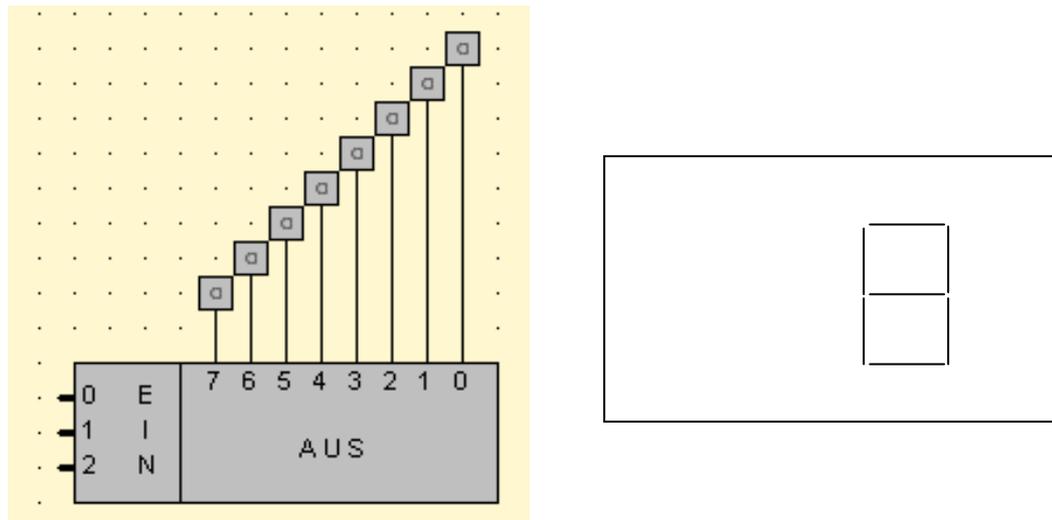


Abb. 5.5-4 Ansteuerung im Programm und Ergebnis in der Anzeige

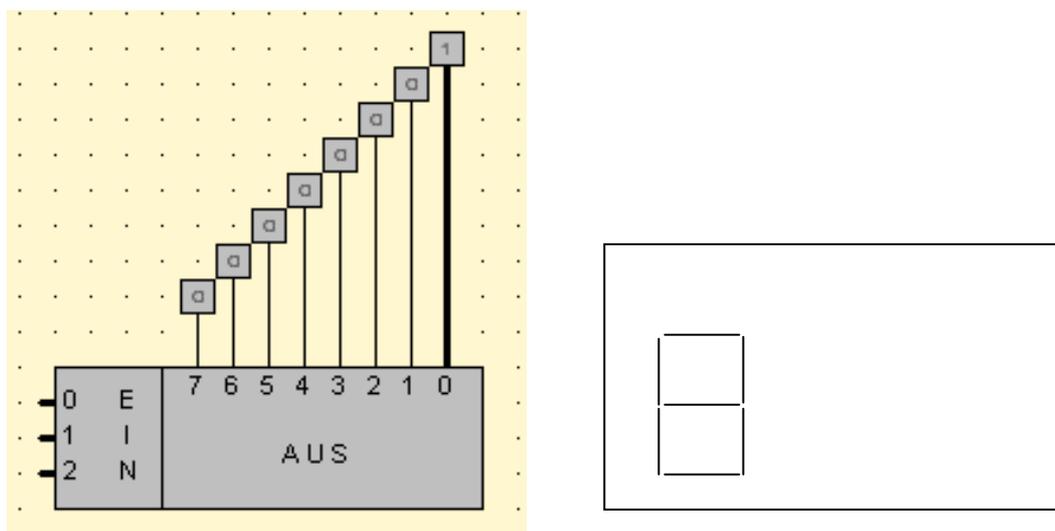


Abb. 5.5-5 Ansteuerung im Programm und Ergebnis in der Anzeige

Das Signal auf der Leitung 0 dient also zum Umschalten zwischen den Stellen. Liegt an der Leitung 0 ein Signal an, so wird die Ziffer auf die Zehnerstelle geschrieben. Die Schüler und Schülerinnen werden schnell merken, dass es zur systematischen Erfassung des Zusammenhangs zwischen der Ansteuerung und dem Ergebnis mehr Sinn macht, erst alle Signale zu setzen und dann Signale wegzulassen. Damit ergibt sich die folgende Tabelle. Ein X bedeutet, dass das Signal gesetzt ist. Die fett

markierten Segmente der Anzeige leuchten auf. Wegen der besseren Übersicht sind auch die anderen Segmente angegeben.

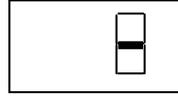
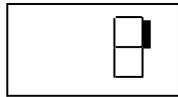
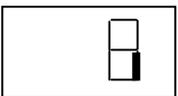
7 6 5 4 3 2 1	
X X X X X X X	
Nichts	
- X X X X X X	
Mitte	
X - X X X X X	
Oben, Rechts	
X X - X X X X	
Unten, Rechts	
X X X - X X X	
Oben, Mitte	
X X X X - X X	
Oben, Links	
X X X X X - X	
Unten, Links	
X X X X X X -	
Unten, Mitte	

Abb. 5.5-6 Tabelle zur Ansteuerung der zweistelligen 7-Segmentanzeige, X bedeutet Signal gesetzt und – bedeutet kein Signal

Die Schüler und Schülerinnen sollen nun die Ziffern von 1 bis 9 mit Locad2002 erzeugen, d.h. der Benutzer wählt eine Ziffer aus und Locad2002 muss dafür sorgen, dass die richtigen Signale ausgegeben werden. So müssen zum Beispiel zur

Darstellung der Ziffer 3 die Leitungen 7, 6, 5, 4, 1 auf Null gesetzt werden, also kein Signal haben. Anders formuliert braucht man also Strom auf den Leitungen 2 und 3.

Dazu wird ausprobiert, welche „Stromzuführung“ in Locad2002 zu welcher Ziffer führt. Das ergibt folgende Tabelle.

Ziffer der 7-Segmentanzeige	Leitungen beim Parallelportbaustein in Locad2002 eingeschaltet
0	7
1	7,4,3,2,1
2	5,3
3	2,1
4	4,2,1
5	6,2
6	6
7	7,3,2,1
8	Keine
9	2

Abb. 5.5-7 Tabelle zur Ziffernausgabe der 7-Segmentanzeige

Die Schüler und Schülerinnen können nun mit Hilfe von Koppeldioden eine Schaltung entwerfen, welche die Ausgabe der Ziffern automatisiert.

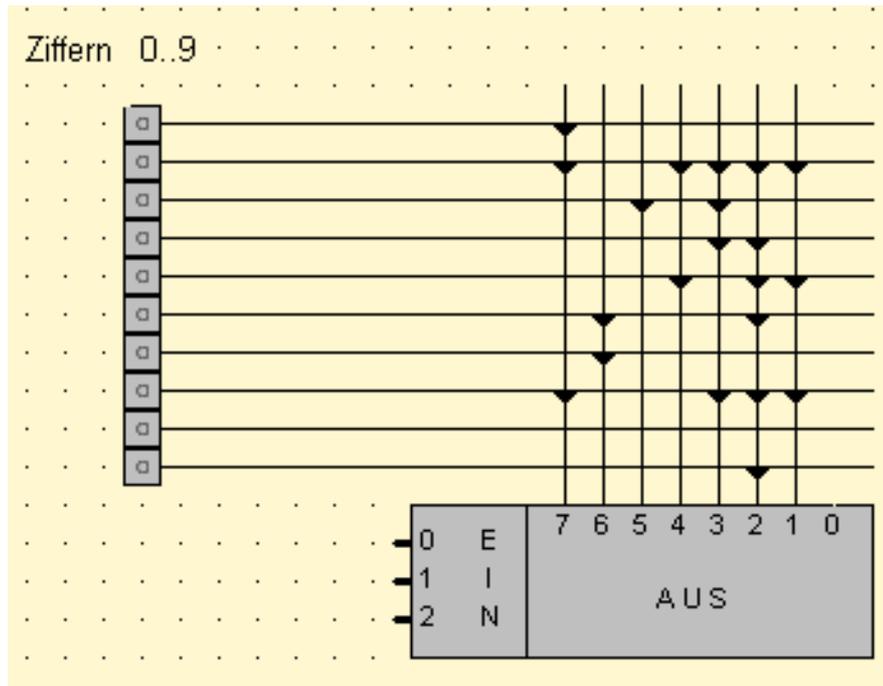


Abb. 5.5-8 Schaltung zur Ziffernausgabe der 7-Segmentanzeige

Die Schaltung kann jetzt so erweitert werden, dass die Ziffer an die Zehnerstelle gesetzt wird. Soll eine zweistellig Zahl angezeigt werden, so muss der Eingang Null ständig kurze Impulse bekommen. Zum Setzen der Zehnerstelle muss der Eingang Null des Parallelportbausteins das Signal eins erhalten. Das ständige hin- und herschalten am Eingang Null erfolgt am besten mit einem Taktgenerator, der als Bauteil in Locad2002 vorhanden ist. Weiteren Aktivitäten sind nun möglich. Zum Beispiel das Konstruieren einer Schaltung zur Sekundenangabe. Dazu muss der Taktgenerator entsprechend eingestellt und Zählwerke verwendet werden.

## Hardware-Modul

Der Bau des Interfaces ist für eine Lerngruppe möglich. Die nötigen Teile und eine ausführliche Anleitung können unter der angegebenen Kontaktadresse bestellt werden. Darunter sind Bausätze mit geätzten und gebohrten Platinen und allen erforderlichen Bauteilen einschließlich einer ausführlichen Anleitung zum Aufbau und Test. Beim Entwurf der Platinen wurde darauf geachtet, dass diese auch von einer Schülergruppe mit unterschiedlichen Fertigkeiten im Lötten ohne Schwierigkeiten aufgebaut werden können (breite Leiterbahnführungen, ausreichender Abstand der Bauteile voneinander).

Kontaktadresse: Eberhard Eube  
Klößgeskamp 19  
41564 Kaarst  
Tel. 02131 65923



Abb. 5.5-9 vorgefertigte Platine mit den dazugehörigen Bauteilen

Das Lötten einer solchen Platine kann auch innerhalb von Projektwochen oder in einer Arbeitsgruppe erfolgen. Gegen einen geringen Betrag können die in Bild 5.5-9

gezeigten Bauteile und die Platine bestellt werden. Es kann für die Schüler und Schülerinnen nur von Vorteil sein einmal in ihrer Schullaufbahn ein funktionierendes Interface gebaut zu haben.

## **6 Anhang**

### **6.1 Bezugsquelle LOCAD 2002**

#### 6.1.1 Lizenzarten

Das CAD-System *LOCAD2002* kann in folgenden Lizenzen erworben werden:

- Einzellizenz

Die Einzellizenz berechtigt Sie zur Nutzung des Programms analog zur Benutzung eines Buches. Das Softwareprodukt darf nicht gleichzeitig von verschiedenen Personen an verschiedenen Orten und auf verschiedenen Geräten benutzt werden. Kopien dürfen lediglich zum Zweck der Datensicherung angefertigt werden.

- Mehrfachlizenz

Die Mehrfachlizenz gestattet die gleichzeitige Nutzung des Programms auf allen Computern der Ausbildungsstätte.

- Campuslizenz

Die Campuslizenz gestattet wie die Mehrfachlizenz die gleichzeitige Nutzung des Programms auf allen Computern der Ausbildungsstätte. Darüber hinaus erlaubt sie, Arbeitskopien in aus-reichender Anzahl anzufertigen, die während der Nutzung des Systems im Unterricht an die beteiligten Auszubildenden zur Unterrichtsvor- und – Nachbereitung bzw. zum Üben zuhause ausgeliehen werden dürfen.

Um eine funktionsfähige Arbeitskopie des CAD-Systems zu erstellen, müssen nur die Dateien

...  
*Locad2002.exe*, *Locad2002.dat* und *Locad2002.hlp* auf eine Diskette kopiert werden.

☞ **Bitte unbedingt beachten:**

Nach Abschluss der Unterrichtsreihe müssen die ausgeliehenen Arbeitskopien von der Ausbildungsstätte zurückgefordert werden.

- **Zusatzlizenz**

Zur Unterrichtsvor- und -nachbereitung bzw. zum Üben zuhause können die Auszubildenden einer Ausbildungsstätte, die eine Mehrfach- oder eine Campuslizenz besitzt, eine besonders günstige Zusatzlizenz erwerben.

Die Zusatzlizenz kostet 20% des Einzellizenzpreises. Bei der Bestellung ist eine schriftliche Bestätigung der Ausbildungsstätte erforderlich.

## **Lieferumfang und Systemvoraussetzungen**

Das CAD-System *LOCAD2002* wird auf einer CD geliefert, die im Stammverzeichnis das Installationsprogramm **Setup.exe** und in den Verzeichnissen *BuchTexte*, *Programme* und *Schaltungsbeispiele* die Texte der Begleitbücher als Word-Dokumente und PDF-Dateien, die Beispielprogramme zum Themenbereich Messen, Steuern und Regeln als Delphi-Projekte und die Schaltungsbeispiele aus den Begleitbüchern enthält.

Hierbei handelt es sich um die Bücher „Technische Informatik“ [1], „Messen, Steuern und Regeln“ [2] und „Mikroprozessortechnik“ [3].

Zum Betrieb des CAD-Systems werden ein Betriebssystem ab Windows 95, ein Prozessor ab Pentium 90, eine Grafikkarte mit einer Mindestauflösung von 800 x 600 Punkten und  
ca. 3 MB Festplattenspeicher benötigt.

### 6.1.2 Installation und Konfiguration

Durch Starten des Programms **Setup.exe** wird das System nach Bestätigung des Lizenzvertrages auf der Festplatte eingerichtet.

Das Installationsprogramm führt folgende Aktionen durch:

- im Verzeichnis *Programme* wird ein neues Verzeichnis mit dem Namen *LOCAD2002* angelegt und darin die Unterverzeichnisse *MPT*, *MSR* und *TI* eingerichtet
- die notwendigen Programmdateien werden kopiert
- die Schaltungsbeispiele aus [1] werden im Unterverzeichnis *TI* abgelegt
- die Schaltungsbeispiele aus [2] werden im Unterverzeichnis *MSR* abgelegt
- die Schaltungsbeispiele aus [3] werden im Unterverzeichnis *MPT* abgelegt.

Die Dateinamen der Schaltungsbeispiele sind so gewählt, dass eine einfache Zuordnung über den Buchtitel und die Seitenzahl möglich ist. Die Schaltung auf Seite 62 in [1] hat den Datei-namen *TI062.LWS*, die Schaltung auf Seite 72 in [2] finden Sie unter dem Namen *MSR072.LWS* und die Schaltung auf Seite 77 in [3] unter *MPT077.LWS*.

Nach der Installation können Sie eine Konfiguration des System nach Ihren Wünschen vornehmen. Im einzelnen haben Sie die Möglichkeit

- die Darstellungsform der Schaltsymbole zu ändern, wobei Sie zwischen den alten und den neuen DIN-Symbolen wählen können
- die Farben für Arbeitsblatt, Linien, Bauteile und Texte zu verändern
- die Parallelschnittstelle für das Interface festzulegen
- die Bauelemente auszuwählen, die im Bauteilmenü zur Wahl stehen sollen

{ Beim Einsatz des Systems im Unterricht ist es unter Umständen aus didaktischen Gründen sinnvoll, bei den einfachen Bauelementen nur die Grundverknüpfungen UND, ODER und INVERTER zuzulassen und bei den komplexeren nur die selbst-definierten IC's ("geschrumpfte" Schaltungen). }

- den Standardwert für die Simulationsverzögerung zu ändern
- die Druckbildgröße sowie die Horizontal- und Vertikalverschiebung festzulegen
- den Warnton ein- bzw. auszuschalten

Die durchgeführte Konfiguration wird in der Datei *Locad2002.dat* gespeichert und steht so lange zur Verfügung, bis eine Konfigurationsänderung vorgenommen wird.

Mit dem Programm *L3\_in\_LfW.exe* können Sie Schaltungen und ICs, die mit *LOCAD 2.x* oder *LOCAD 3.x* erstellt wurden, automatisch in das für *LOCAD2002* benötigte Format umwandeln lassen. Mit *LOCAD für Windows* erstellte Schaltungen können ohne Umwandlung verwendet werden.

### 6.1.3 Einsatz von *LOCAD2002* unter Windows NT, 2000, XP

#### **Mausposition und Fadenkreuz**

Sollte es bei Ihrem System vorkommen, dass beim Ziehen von Leitungen das Fadenkreuz nicht auf den nächstgelegenen Rasterpunkt springt bzw. die Linie nicht mit dem Fadenkreuz beginnt, sondern dazu versetzt erscheint, liegt das aller Wahrscheinlichkeit nach daran, dass eine zu große Schriftart eingestellt ist.

Ändern Sie in diesem Fall die Schriftart folgendermaßen:

Start -> Einstellungen -> Systemsteuerung -> Anzeige -> Einstellungen -> Weitere Optionen-> Schriftgröße und dort "kleine Schriftarten".

#### **Parallelport-Baustein**

Da die Betriebssysteme Windows NT, 2000 und XP Win32-Applikationen keine direkten Zugriffe auf die parallelen Ports des PC erlauben, kommt es bei Benutzung eines Parallelport-Bausteines zur Ausgabe von Daten über einen LPT-Port beim Einschalten einer Schaltung, die einen Port-Baustein enthält, zu der Fehlermeldung „Privilegierte Anweisung“.

Beenden Sie in diesem Fall das Programm über das Anklicken des „Abbruch“-Buttons .

Sie können das Problem dann folgendermaßen lösen:

- Lassen Sie das Programm unter Windows 95, Windows 98 oder Windows ME laufen  
*oder*

- Im Internet finden Sie unter der Adresse <http://www.lvr.com/files/> die Datei „userport.zip“ (34 KB groß), die eine von Tomas Franzon entwickelte Möglichkeit zum Hardwarezugriff bietet. Sie benötigen aus der ZIP-Datei nur die beiden Dateien "UserPort.sys" und "UserPort.exe".

Kopieren Sie die Datei "UserPort.sys" in das Verzeichnis *WINNT\SYSTEM32\DRIVERS*.

Starten Sie das Programm **UserPort.exe** und klicken Sie dort auf **Start** und anschließend auf **Exit**. Dadurch wird der Hardwarezugriff auf die voreingestellten Adressen ermöglicht.

Wenn Sie den Hardwarezugriff wieder deaktivieren wollten, müssen Sie das Programm nochmals starten und **Stop** anklicken.

## 6.2 *Das Interface*

von Eberhard Eube

Der Grundgedanke bei der Entwicklung des Interface war, durch einen modularen Aufbau eine übersichtliche und erweiterungsfähige Schaltung zu erhalten, mit der Steuerungen und Regelungen über die Centronics-Schnittstelle eines Rechners realisiert werden können.

Hierzu werden auf der Hauptplatine die Datenleitungen gepuffert und die erforderlichen Versorgungsspannungen zur Verfügung gestellt. An den beiden Steckerleisten können sowohl die unten beschriebene EIN-AUSGABE-Platine und die ADDA-Platine als auch beliebige Schaltungen für weitergehende Anwendungen (Schrittmotorsteuerung, Relaischaltungen o.ä.) angeschlossen werden.

### 6.2.1 Hauptplatine

Die folgenden Abbildungen zeigen die fertig aufgebaute Hauptplatine und das Blockschaltbild dazu.

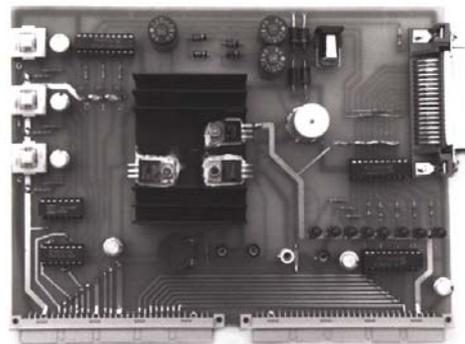


Abb. 6.2.-1 Hauptplatine des Interface

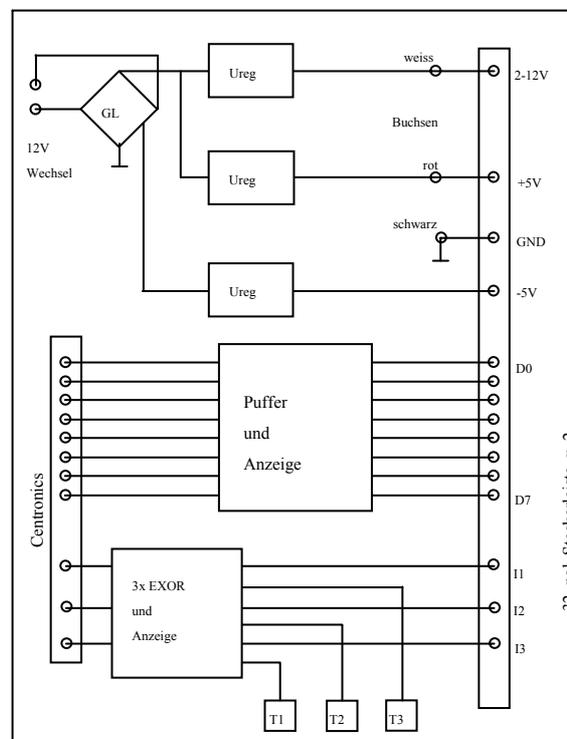


Abb. 6.2-2 Blockschaltbild der Hauptplatine

Im Blockschaltbild erkennt man die drei Hauptgruppen der Platine:

Spannungsversorgungen

Ausgangsdaten

Eingangsdaten

Als Spannungsversorgung stehen zwei Festspannungsregler für +5V und -5V sowie ein einstellbarer Spannungsregler für den Bereich 2-12V zur Verfügung.

Das zugehörige Potentiometer befindet sich unterhalb des Kühlkörpers. An dem Buchsenpaar (schwarz-weiß) neben dem Potentiometer kann die eingestellte Spannung gemessen werden. Das rot-schwarze Buchsenpaar liefert die Festspannung +5V.

In Abb. 6.2-1 sehen Sie rechts oben die Centronics-Buchse zum Anschluss des Interface über ein Standard-Drucker-kabel an den Rechner. Die Datenausgangsleitungen des Datenregisters laufen über eine Treiberstufe zu den Leuchtdioden als Anzeige und über einen weiteren Puffer zu den beiden Steckerleisten.

Von den 5 Eingangsleitungen des Status-Register werden hier 3 benutzt. Sensorsignale, die über die Steckerleisten zugeführt werden, lassen sich mit den drei Tastern invertieren. Der am Parallelport des Rechners anliegende Zustand wird durch die drei grünen Leuchtdioden links oben angezeigt.

### 6.2.2 EIN-AUSGABE-Platine

Die nächsten Abbildungen zeigen die EIN-AUSGABE-Platine, die an eine der Steckerleisten angeschlossen werden kann.

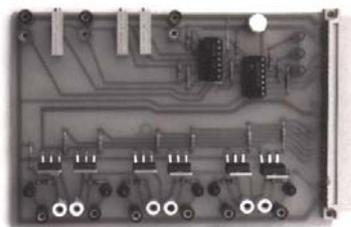


Abb. 6.2-3 EIN-AUSGABE-Platine

### 6.2.3 ADDA-Platine

Die beiden folgenden Abbildungen zeigen die ADDA-Platine und das Blockschaltbild dazu.

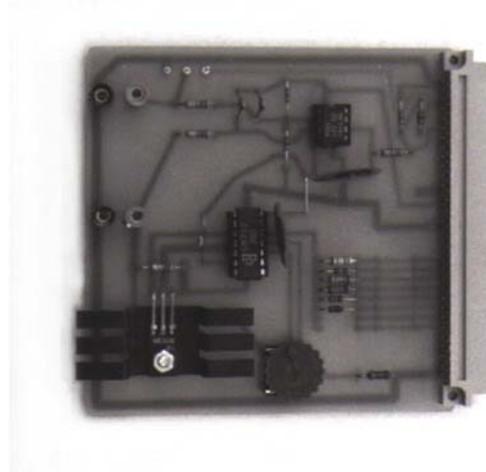


Abb. 6.2.-5 ADDA-Platine

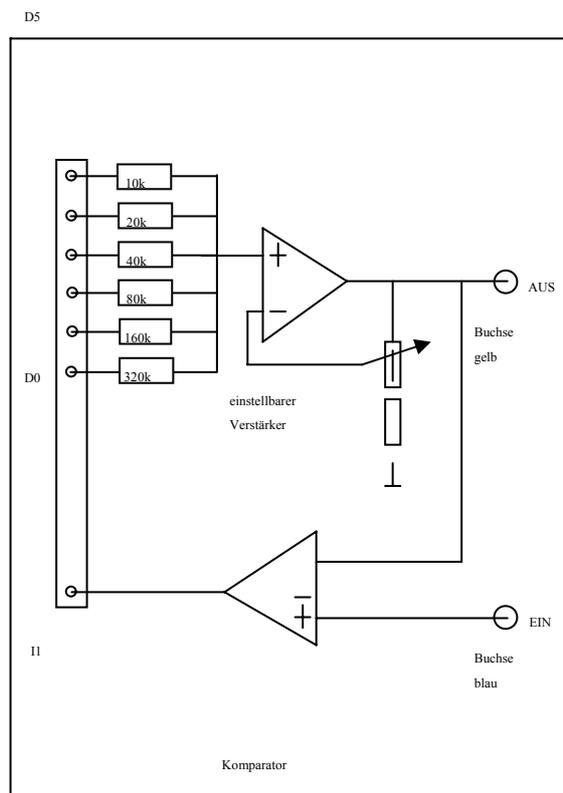


Abb. 6.2.-6 Blockschaltbild zur ADDA-Platine

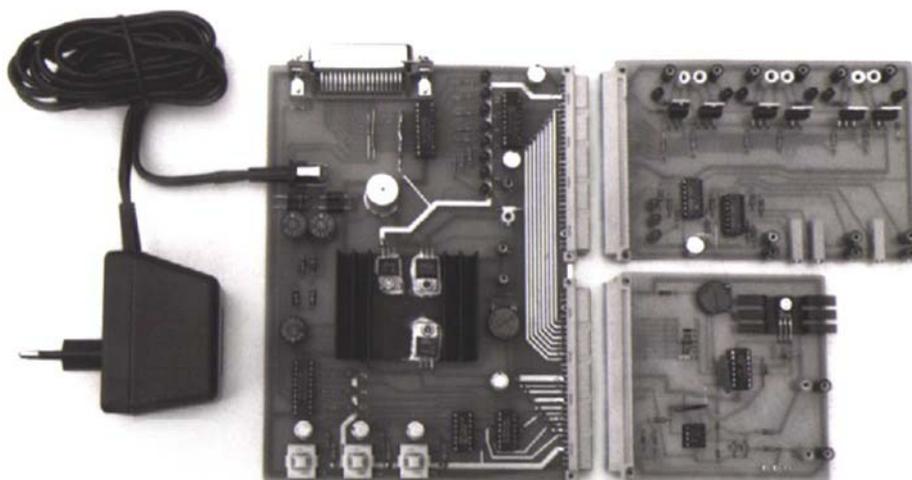
Auf der ADDA-Platine ist ein Digital/Analog-Wandler mit diskreten Widerständen realisiert (siehe Kapitel 5). Über die Datenausgänge des Rechners kann hiermit eine

Analogspannung festgelegt werden, die über einen Verstärker mit einstellbarem Verstärkungsfaktor zur gelben Ausgangsbuchse geführt wird.

Die Analog/Digital-Wandlung wird über einen Komparator durchgeführt, der die am blau-schwarzen Buchsenpaar anliegende Eingangsspannung mit der Ausgangsspannung des Verstärkers vergleicht und ein entsprechendes Signal über eine Dateneingangsleitung an den Rechner zurückliefert.

Die drei Lötstifte oberhalb des Eingangsbuchsenpaares können dazu benutzt werden, mit Hilfe eines Potentiometers eine Eingangsspannung zu erzeugen, falls keine externe Spannungsquelle zur Verfügung steht.

Die folgende Abbildung zeigt das komplette Interface mit Netzteil.



Zur Direktansteuerung des Interface steht das Programm PP\_STRG auf der CD im Verzeichnis *Programme/Parallelport* zur Verfügung.

#### 6.2.4 Bezugsquelle des Interface

Die drei beschriebenen Baugruppen sind als Bausätze mit geätzten und gebohrten Platinen und allen erforderlichen Bauteilen einschließlich einer ausführlichen Anleitung zum Aufbau und Test bei der unten angegebenen Kontaktadresse erhältlich.

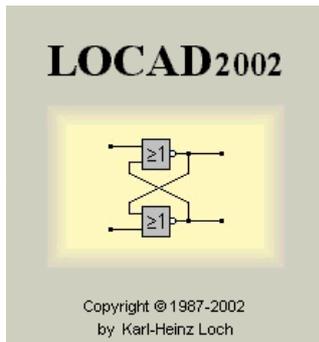
Beim Entwurf der Platinen wurde darauf geachtet, dass diese auch von einer Schülergruppe mit unterschiedlichen Fertigkeiten im Lötten ohne Schwierigkeiten

aufgebaut werden können (breite Leiterbahnführungen, ausreichender Abstand der Bauteile voneinander).

Kontaktadresse: Eberhard Eube  
Klößgeskamp 19  
41564 Kaarst  
Tel. 02131 65923

### ***6.3 Arbeitsblätter zu den einzelnen Kapiteln***

Die folgenden Arbeitsblätter sind als Kopiervorlagen oder als Anregung für Aufgaben gedacht. Zum schnelleren Arbeiten sind die Lösungen immer auf der folgenden Seite angegeben.

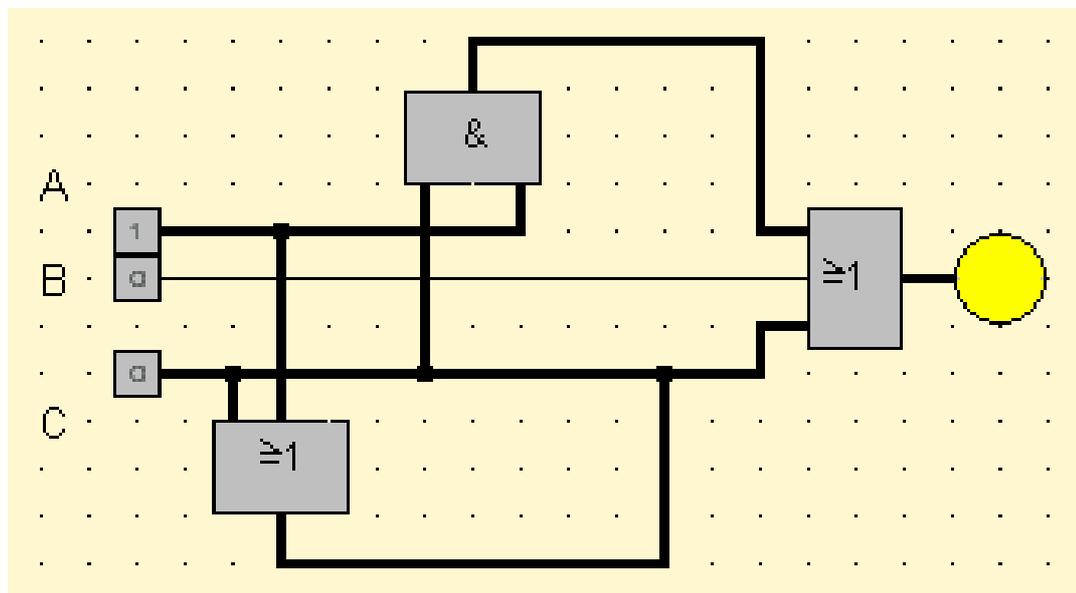


### 6.3.1 Arbeitsblatt Grundlagen 1 von 2

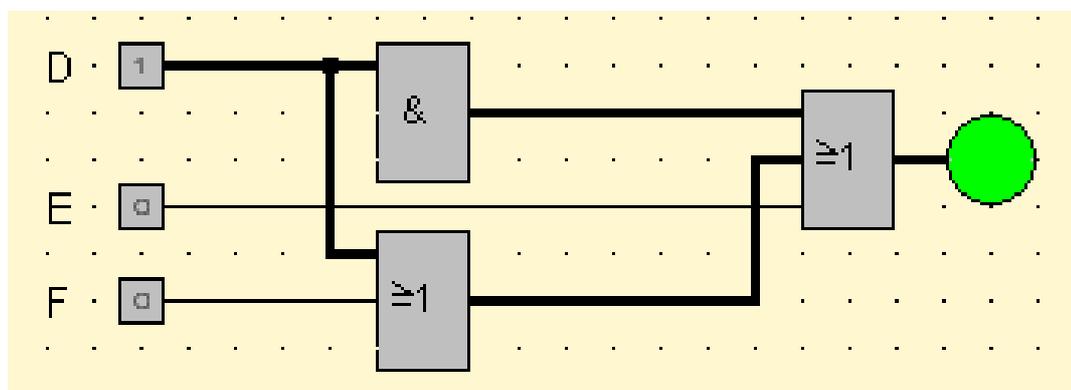
1) Starte LOCAD2002 und baue die angegebene Schaltung unter A nach und speichere Sie unter deinem Namen ab.

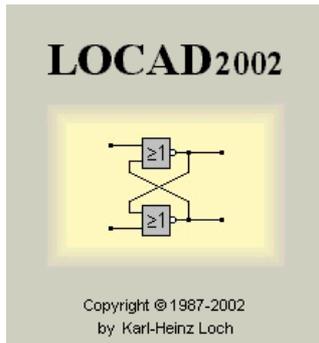
2) Ändere deine Schaltung ab, damit sie so aussieht wie unter B gezeigt und speichere sie unter B\_dein Name ab.

#### Schaltung A:



#### Schaltung B:

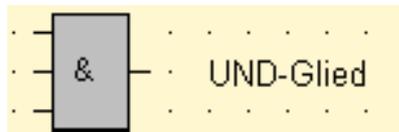




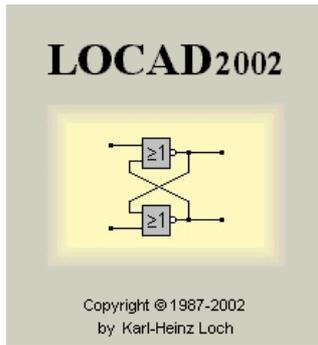
### 6.3.2 Arbeitsblatt Grundlagen 2 von 2

1) Wähle die ersten 10 Bauteile aus der Bauteilliste und platziere sie auf dem linken Rand . Schreibe jeweils rechts dazu den passenden Namen und speichere dein Ergebnis ab.

Das soll dann so aussehen:

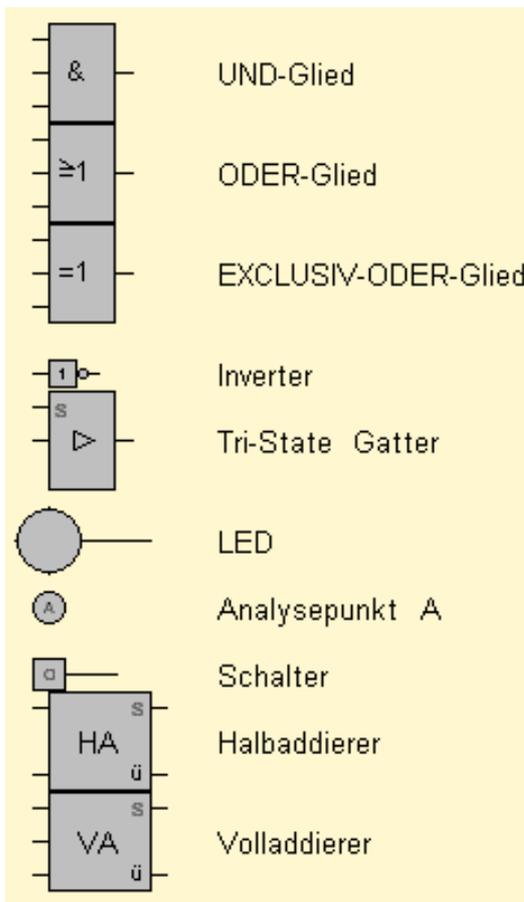


- 2) Kopiere die Bauteile in Word und beschrifte sie dort.
- 3) Versucht in Partnerarbeit die Funktion der Bauteile herauszubekommen und schreibt diese Funktion in kurzen Sätzen auf.



### 6.3.3 Lösungen für Grundlagen 2 von 2

Auflistung der ersten 10 Bauteile mit Bezeichnung



Signaldurchgang, wenn alle anliegenden Eingangsleitungen Strom führen

Signaldurchgang, wenn mindestens eine anliegende Eingangsleitung Strom führt

Signaldurchgang, wenn höchstens eine anliegende Eingangsleitung Strom führt

Invertierung des Eingangssignals

Steht Steuereingang S auf 1, wird der Wert des zweiten Eingangs durchgestellt

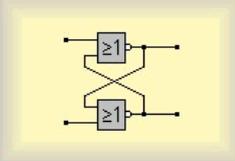
Leuchtet wenn Leitung Strom führt

Punkt zum Untersuchen im Oszilloskop

Ermöglicht den Stromfluss in einer Leitung

Summe aus zwei binären Ziffern

Summe aus drei binären Ziffern

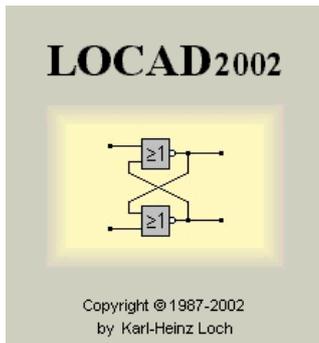
**LOCAD2002**

Copyright © 1987-2002  
by Karl-Heinz Loch

**6.3.4 Arbeitsblatt Einfache Schaltungslogik 1 von 3**

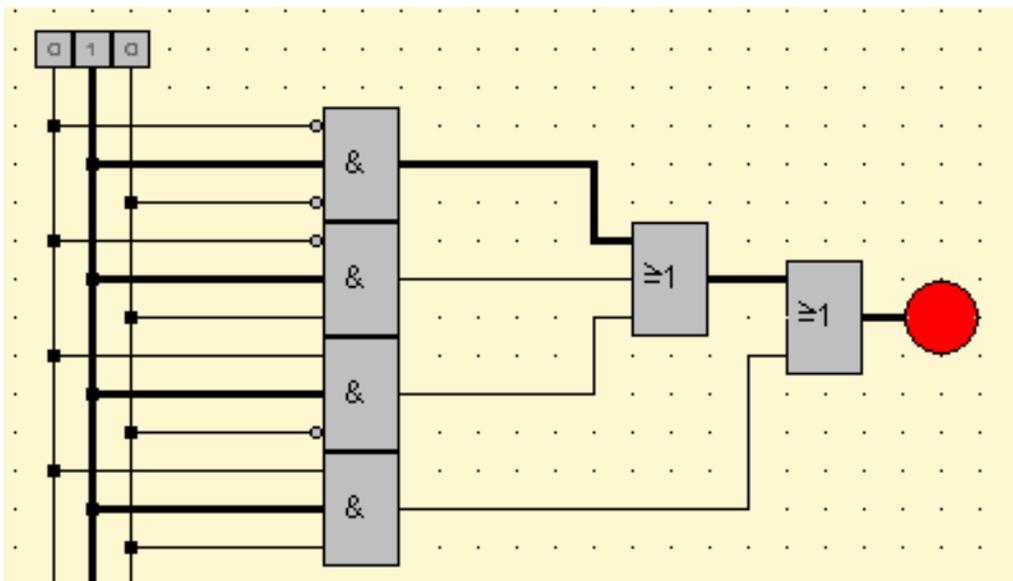
- 1) Baue eine Schaltung nach folgender Wahrheitstabelle auf und überprüfe jede Zeile.
- 2) Formuliere in Worten, wann das LED leuchtet. Suche möglichst eine einfache Formulierung.
- 3) Stelle einen mathematischen Ausdruck für das Leuchten der LED auf. Beginne so:  $L = \dots$

<b>A</b>	<b>B</b>	<b>C</b>	<b>LED (L)</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>
<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>
<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>
<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>



### 6.3.5 Lösungen Einfache Schaltungslogik 1 von 3

Die geforderte Schaltung sieht folgendermaßen aus:



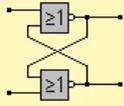
Eine möglichst einfache Formulierung lautet: Das LED leuchtet, wenn auf Leitung B Strom fließt.

Der mathematische Ausdruck lautet:  $L = \bar{A} B \bar{C} + \bar{A} B C + A B \bar{C} + A B C$

Hinweis zum mathematischen Ausdruck:

Aufgrund der Rechenregeln fallen drei Terme weg und es ergibt sich  $L = B$ .

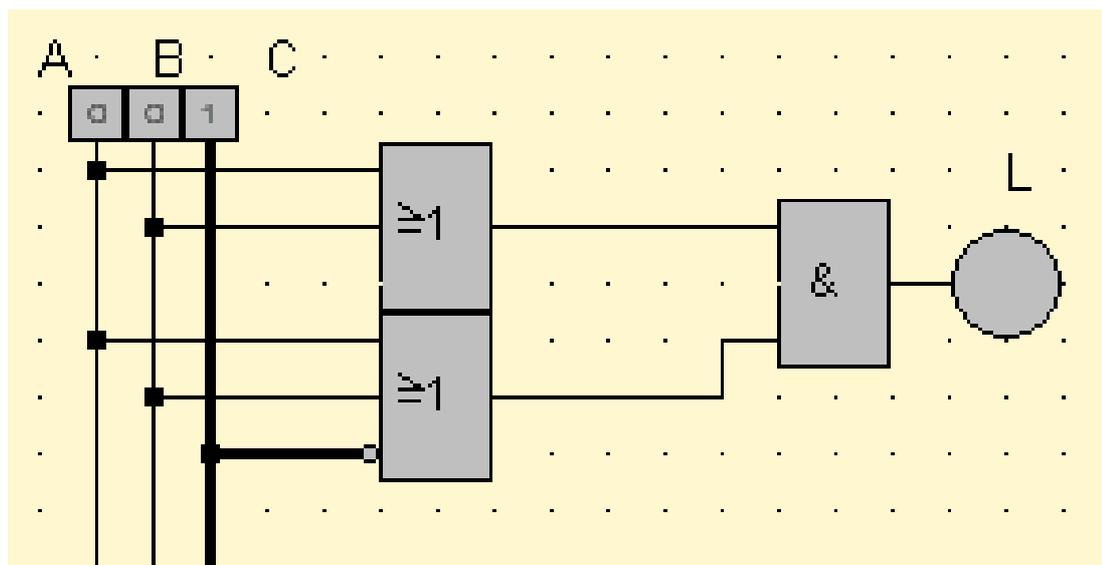
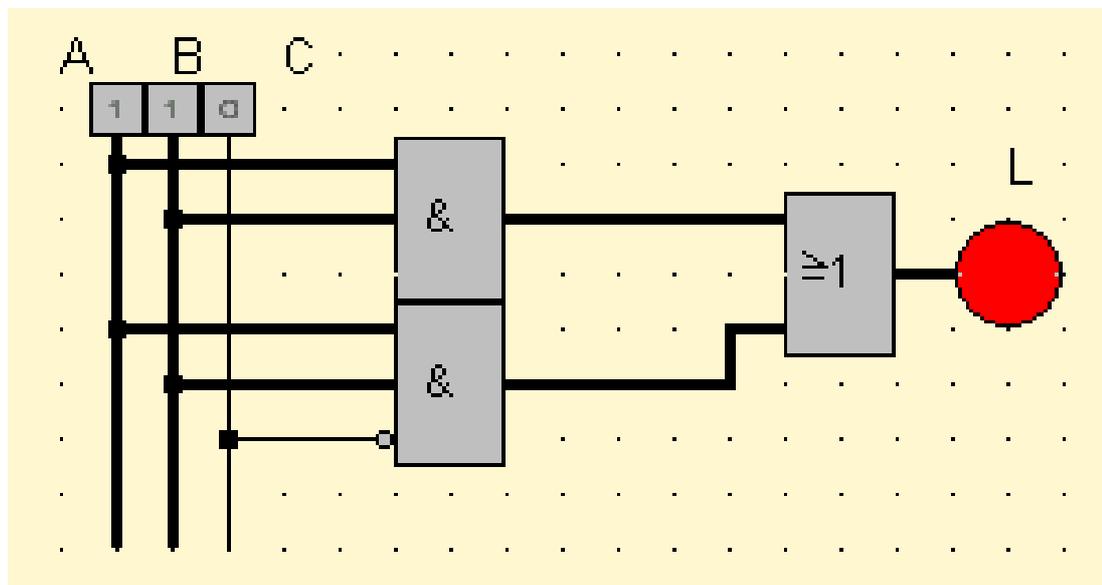
## LOCAD2002

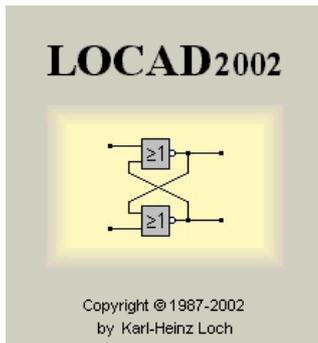


Copyright © 1987-2002  
by Karl-Heinz Loch

### 6.3.6 Arbeitsblatt Einfache Schaltungslogik 2 von 3

- 1) Baue die folgenden Schaltungen auf
- 2) Finde zu jeder Schaltung eine vollständige Wahrheitstabelle
- 3) Gib jeweils einen mathematischen Ausdruck für das Leuchten der Lampe an und vereinfache soweit wie möglich





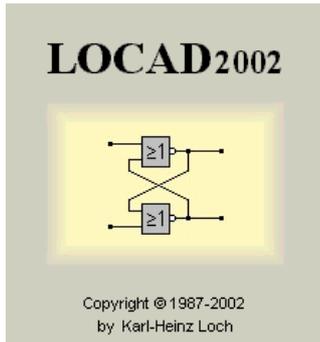
### 6.3.7 Lösungen zu Einfache Schaltungslogik 2 von 3

**Schaltung 1)**  $L = AB + AB\bar{C}$  vereinfacht zu  $L = AB$

A	B	C	LED (L)
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1
1	1	1	1

**Schaltung 2)**  $L = (A + B) (A + B + \bar{C})$  vereinfacht zu  $L = A + B$

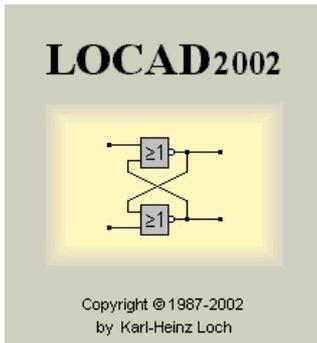
A	B	C	LED (L)
0	0	0	0
0	0	1	0
0	1	0	1
1	0	0	1
0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	1



### 6.3.8 Einfache Schaltungslogik 3 von 3

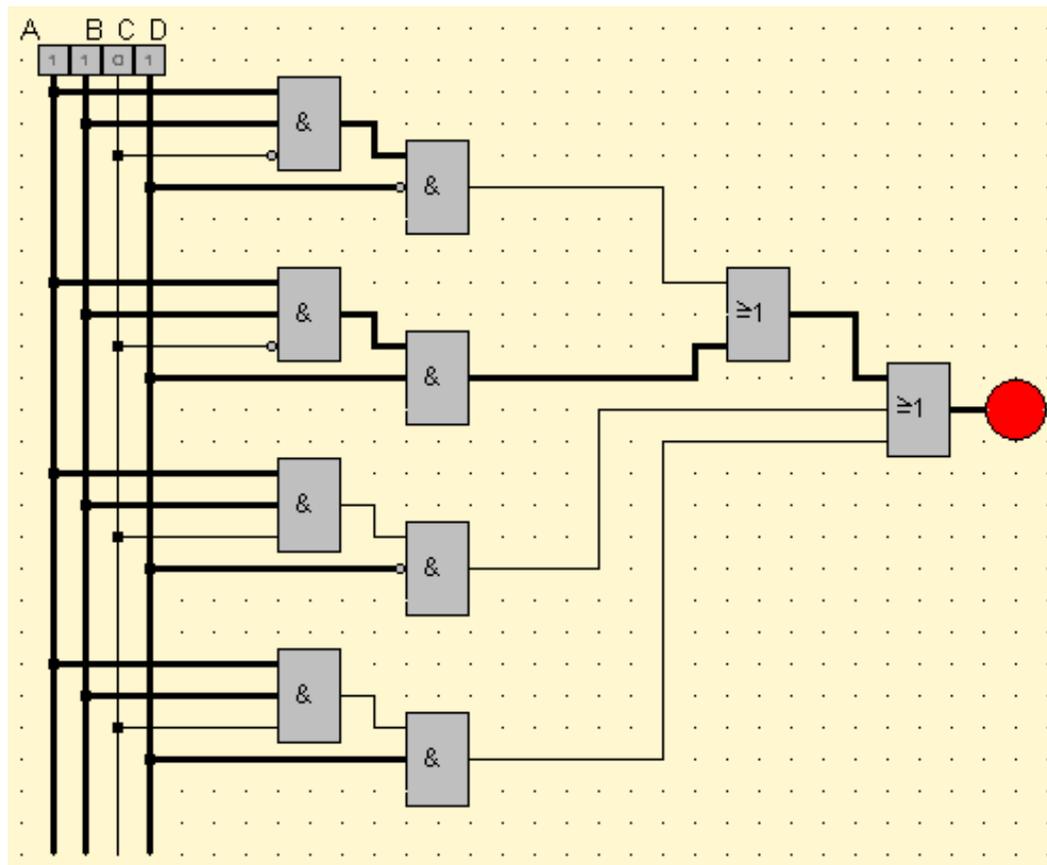
- 1) Baue eine Schaltung nach folgender Wahrheitstabelle auf und überprüfe jede Zeile.
- 2) Formuliere in Worten, wann das LED leuchtet. Suche möglichst eine einfache Formulierung.
- 3) Stelle einen mathematischen Ausdruck für das Leuchten der LED auf. Beginne so:  $L = \dots$

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>L</b>
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



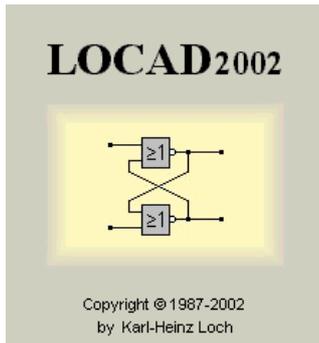
### 6.3.9 Lösungen zu Einfache Schaltungslogik 3 von 3

Die Schaltung sieht folgendermaßen aus:



Der ungekürzte Ausdruck lautet:  $L = ABC\bar{D} + AB\bar{C}D + ABC\bar{D} + ABCD$

Der vereinfachte Ausdruck lautet:  $L = AB$



### 6.3.10 Erweiterte Schaltungslogik 1 von 3

Stelle für folgende Tabellen eine konjunktive und eine disjunktive Normalform auf.

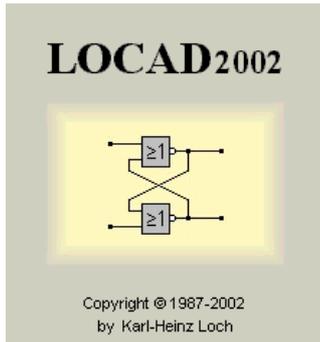
Beginne mit  $L = \dots$

a)

A	B	C	LED (L)
0	0	0	0
0	0	1	1
0	1	0	1
1	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1
1	1	1	1

b)

A	B	C	LED (L)
0	0	0	1
0	0	1	1
0	1	0	1
1	0	0	1
0	1	1	1
1	0	1	0
1	1	0	1
1	1	1	1



### 6.3.11 Lösungen zu erweiterte Schaltungslogik 1 von 3

#### Zu Tabelle a)

Die konjunktive Normalform lautet:  $L = \bar{A} \bar{B} C + \bar{A} B C + A B \bar{C} + A B C$

Die disjunktive Normalform lautet:

$$L = (A+B+C) (\bar{A} + B + C) (A + \bar{B} + \bar{C}) (\bar{A} + B + \bar{C})$$

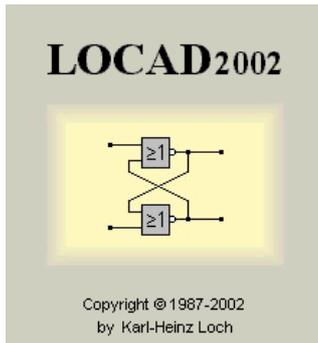
#### Zu Tabelle b)

Die konjunktive Normalform lautet:

$$L = \bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} C + \bar{A} B \bar{C} + A \bar{B} \bar{C} + \bar{A} B C + A B \bar{C} + ABC$$

Die disjunktive Normalform lautet:

$$L = A + \bar{B} + C$$



### 6.3.12 Erweiterte Schaltungslogik 2 von 3

1) Erläutere kurz die Funktion eines PROM-Speichers, was sind Vorteile, was sind Nachteile?

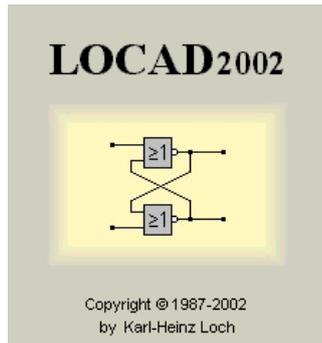
2) Baue für die beiden folgenden Aufgaben a), b) jeweils ein PROM auf, überlege dir die Lösung erst auf einem Blatt.

Wie viele Adressleitungen werden jeweils benötigt?

a) Die ersten 4 Quadratzahlen sollen als Ausgabe auf den Datenleitungen angezeigt werden.

b) Der PROM soll folgende Eigenschaften haben:

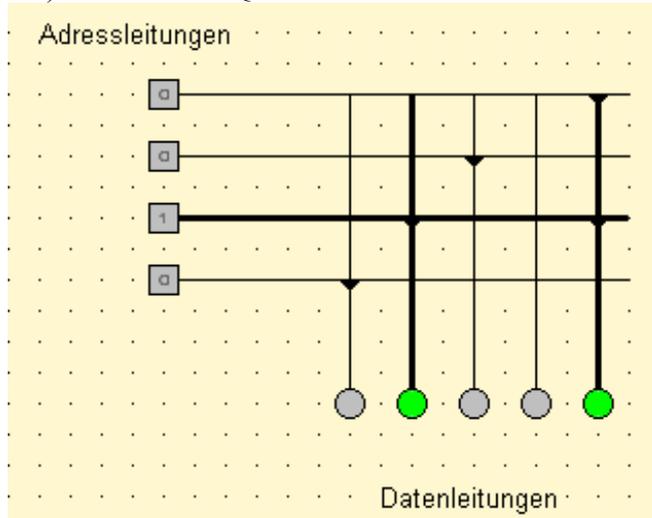
Adressleitungen	Datenleitungen
1	5
2	3
3	12
4	1
5	17



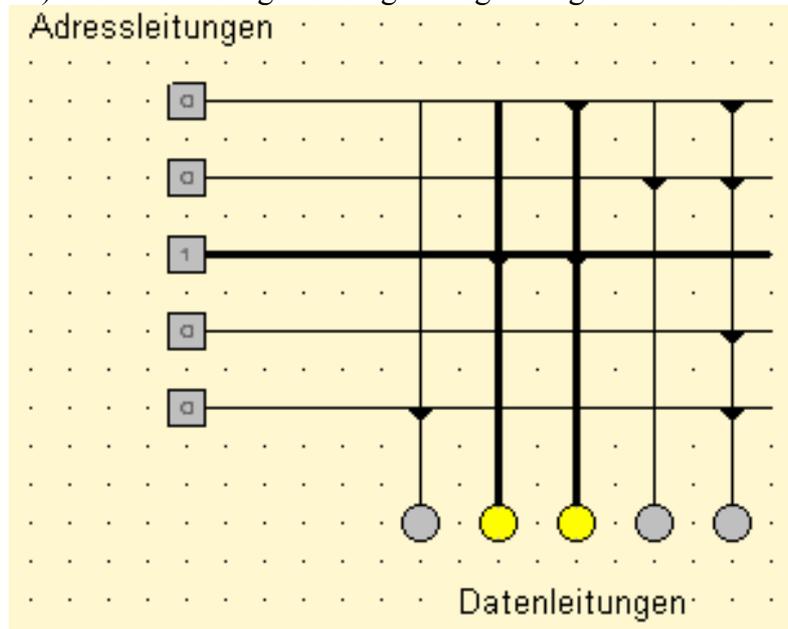
### 6.3.13 Lösungen zu Erweiterte Schaltungslogik 2 von 3

1) In einem PROM-Speicher werden Informationen dauerhaft „eingebrannt“. Der Vorteil dabei ist, dass ein Computer immer auf diese Daten zugreifen kann, auch wenn es vorher zum totalen Systemabsturz gekommen ist. Der Nachteil liegt in der starren Struktur der Speicherung, die nicht mehr geändert werden kann.

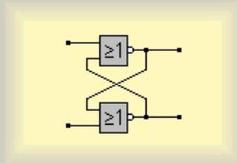
2 a) Die ersten 4 Quadratzahlen werden auf die ersten 4 Adressleitungen gelegt



2 b) Eine Umsetzung der Aufgabe ergibt folgendes PROM



## LOCAD2002

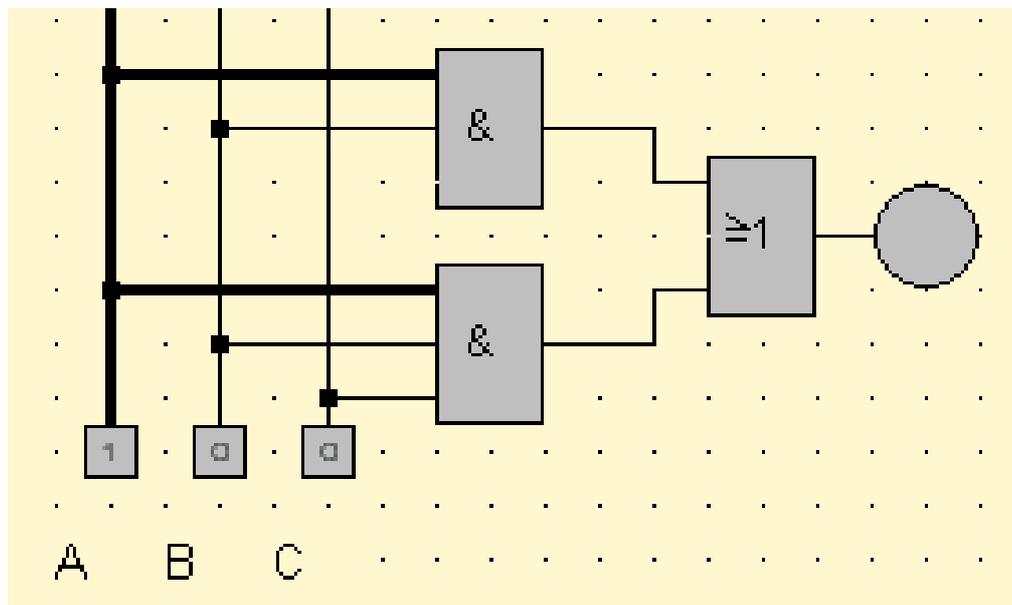


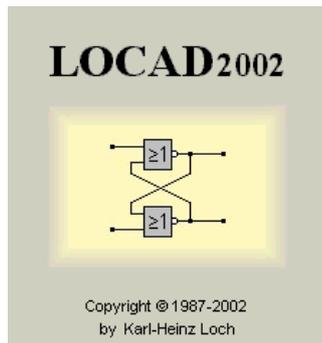
Copyright © 1987-2002  
by Karl-Heinz Loch

### 6.3.14 Erweiterte Schaltungslogik 3 von 3

1) Stelle für die folgende Schaltungen eine Wahrheitstabelle auf.

2) Ersetze die UND-, sowie ODER- Glieder durch NAND-Gatter und überprüfe dein Ergebnis mit einer Wahrheitstabelle



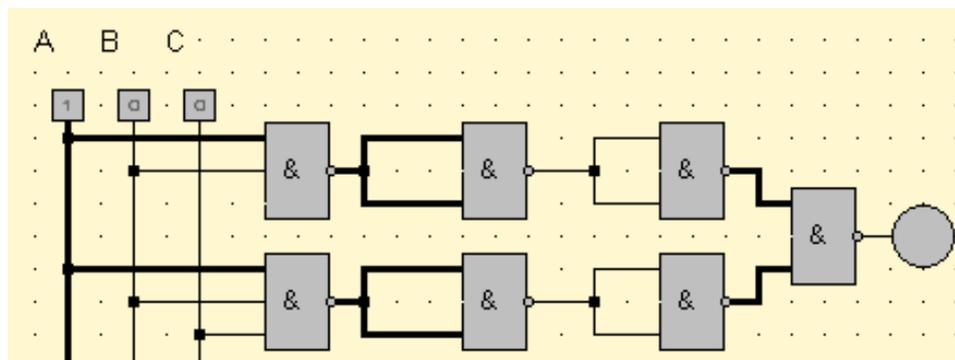


### 6.3.15 Lösungen zu Erweiterte Schaltungslogik 3 von 3

Die Wahrheitstabelle sieht folgendermaßen aus:

A	B	C	LED (L)
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1
1	1	1	1

Das dazugehörige Schaltbild mit NAND-Gattern sieht so aus:



#### **6.4 Literaturverzeichnis**

Kleinere Teile der vorliegenden Arbeit sind aus der folgenden Quelle:

Technische Informatik mit Locad2002

Karl-Heinz Loch  
Weinesch 16  
41812 Erkelenz

alle Arbeitsblätter, Beispiele, Modulfolge und didaktische Überlegungen sind selbstständig verfasst, da ich mit Locad2002 schon im Unterricht gearbeitet habe.

Ich versichere, diese Arbeit selbständig verfasst, keine anderen als die angegebenen Hilfsmittel verwendet und die Stellen, die in anderen Werken im Wortlaut, als Graphik oder dem Sinne nach entnommen sind, mit Quellenangaben kenntlich gemacht zu haben.