

Fach: Automation

##### Thema: Steuerungsgrundlagen

##### 

##### Logische Grundbausteine

##### Kapitel: Grundelemente logischer Schaltungen

##### Autor: Roman Moser

**Inhaltsverzeichnis**

[1. Einleitung 2](#_Toc242927219)

[2. Logik 3](#_Toc242927220)

[2.1 Die Wahrheitstabelle 3](#_Toc242927221)

[2.2 Logikzustände 4](#_Toc242927222)

[2.3 Zeitablaufdiagramme 5](#_Toc242927223)

[2.4 Steuergeräte 7](#_Toc242927224)

[2.5 Vocabulary 8](#_Toc242927225)

[3. Logische Grundbausteine 9](#_Toc242927226)

[3.1 Die UND-Verknüpfung (Konjunktion) 9](#_Toc242927227)

[3.2 Die ODER-Verknüpfung 11](#_Toc242927228)

[3.3 Die NICHT-Verknüpfung (Negation) 12](#_Toc242927229)

[4. Binary logic operations with more than 2 inputs 13](#_Toc242927230)

[4.1 The AND function (conjunction) 13](#_Toc242927231)

[4.2 The OR function (disjunction) 14](#_Toc242927232)

[4.3 Task 15](#_Toc242927233)

[5. Derived Basic Functions 16](#_Toc242927234)

[5.1 The NAND function 16](#_Toc242927235)

[5.2 The NOR function 17](#_Toc242927236)

[5.3 Die XOR-Verknüpfung (Antivalenz oder Exklusiv-ODER) 18](#_Toc242927237)

[5.4 Die XNOR-Verknüpfung (Äquivalenz oder Exklusiv-NOR) 19](#_Toc242927238)

[5.5 Vocabulary 20](#_Toc242927239)

[6. Binary logic operations with inverse inputs 21](#_Toc242927240)

[6.1 Task 21](#_Toc242927241)

# Einleitung

Fast alle Grössen in unserem Leben lassen beliebig viele Zwischenwerte zu. Schon der griechische Philosoph **Aristoteles** (384..322 v. Chr.) versuchte das Denken des Menschen in eindeutige Gesetze zu fassen. Sein „Logiksystem“ arbeitete mit zwei sich ausschliessenden Begriffen **„wahr“** und **„unwahr“** bzw. „ja“ und „nein“.

Der englische Mathematiker **Georges Boole** definierte **1847** für dieses „Logiksystem“ eine Kurzschreibweise, die **„Bool’sche Algebra“.** Die Operanden haben nur zwei mögliche Werte (z.B. **„0“** und **„1“**) und die grundlegenden Operationen heissen **„UND“**, **„ODER“** und **„NICHT“**.

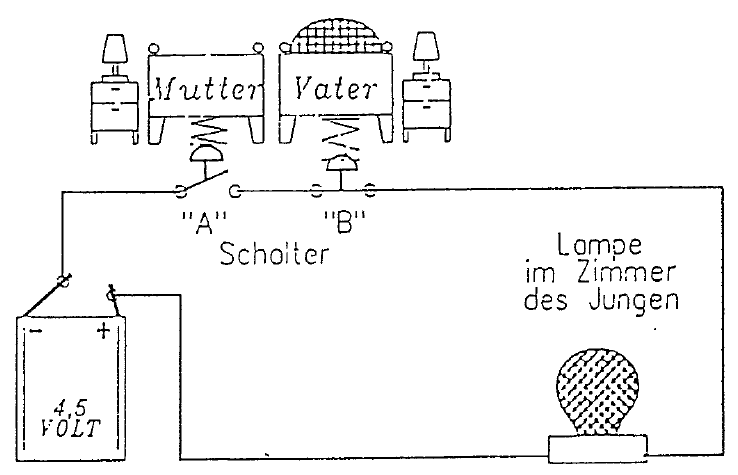
Die Digitaltechnik setzt die Theorie von Boole in die Praxis um. Digitale Systeme lassen sich mit einfachen und ungenauen Bauteilen (z.B. Transistoren) realisieren und erzeugen trotzdem beliebig genaue Resultate! Einige Hunderttausend Transistoren können heute für wenige Franken auf einem wenigen mm2 grossen Siliziumplättchen realisiert werden.

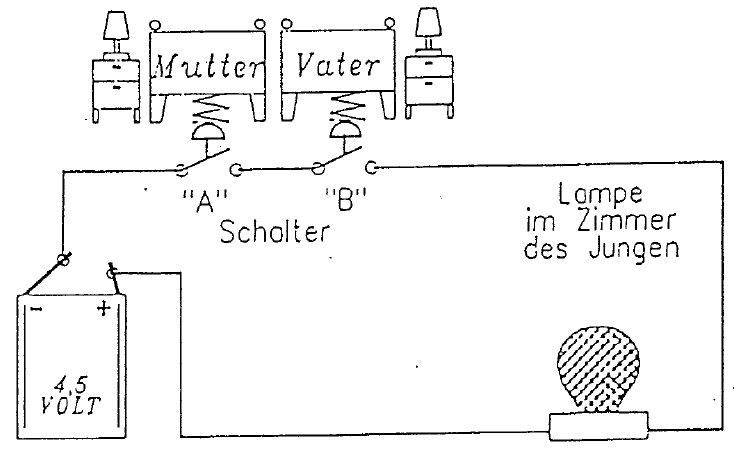
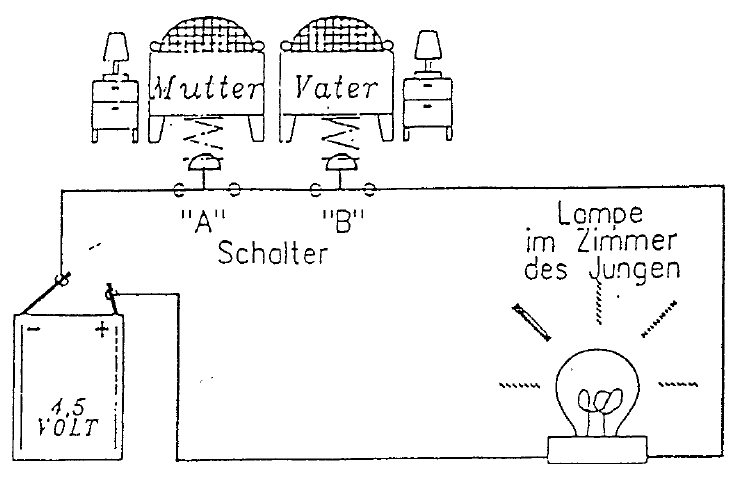
# 

# Logik

## Die Wahrheitstabelle

Wie wird die Arbeitsweise von Steuerungen beschrieben? An einem Beispiel soll dies verdeutlicht werden!

Ein „kleines Schlitzohr“ will nachts gegen den Willen seiner Eltern fernsehen. Er baut sich daher die folgend abgebildete Überwachungs­schaltung. Die Kontroll­lampe in seinem Zimmer brennt nur, wenn beide Eltern im Bett liegen.



|  |  |  |
| --- | --- | --- |
| Schalter | | Lampe |
| A | B |  |
| offen | offen | dunkel |
| offen | zu | dunkel |
| zu | offen | dunkel |
| zu | zu | hell |

Die Funktion der Schaltung kann mit folgender Tabelle beschrieben werden:

In der Digitaltechnik werden solche Tabellen (sogenannte Wahrheitstabellen) in einer verkürzten Schreibweise notiert. Die beiden möglichen Zustände von binären Schalt- und Speicherelementen werden dabei mit den Ziffern „0“ und „1“ beschrie­ben. Was diese Ziffern jeweils bedeuten, wird von Fall zu Fall durch Definition festgelegt und dann konsequent eingehalten.

**Tasks**

1. Draw the truth table as a function of the definitions given!

Definitions: Truth table:

|  |  |  |
| --- | --- | --- |
| switch | | lamp |
| A | B |  |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

If the switch A is open then A = 0  
If the switch A is closed then A = 1

If the switch B is open then B = 0  
 If the switch B is closed then B = 1  
  
If the lamp doesn‘t light up then „lamp“ = 0   
 If the lamp lights up then „lamp“ = 1

1. Complete the table.

|  |  |
| --- | --- |
| English: | German: |
| truth table | Wahrheitstabelle |
| Definition | Definition |
| (to) light up | aufleuchten |
| switch | Schalter |
| … | … |

## Logikzustände

Im Beispiel unter Kapitel 2.1 wurden die Zahlen „0“ und „1“ als Abkürzungen für die beiden möglichen Zustände der binären Schaltungselemente eingeführt. In der Digitaltechnik sind noch andere Bezeichnungen üblich:

|  |  |  |
| --- | --- | --- |
| Logikpegel | 0 | 1 |
| Logikpegel (amerikanisch) | L (low) | H (high) |
| Umgangssprache | inaktiv | aktiv |
| Wahrheitsaussage | falsch (false) | wahr (true) |

**Tasks**

1. Complete the table using an online dictionary. (<http://www.dict.cc/>)

|  |  |
| --- | --- |
| English: | German: |
| energized, pulled in | Angezogen, unter Spannung |
| bright | Hell |
| Open | offen |
| conducts | Leitet |
| dark | Dunkel |
| closed | geschlossen |
| de-energized, dropped out | Abfallen, nicht angezogen |
| closes | sperrt |

1. Decide which English word from task 1 fits best in each cell of the table below.

|  |  |  |
| --- | --- | --- |
| electrical equipment: | logic level „0“ (inactive) | logic level „1“ (active) |
| switch | Open | close |
| lamp | Bright | Drak |
| relay | Drop out  De-Energized | Pulled in  Energized |
| diode | Closed | Conducts |

## Zeitablaufdiagramme

Das Zeitablaufdiagramm zeigt den zeitlichen Verlauf von Ein- und Ausgangssignalen.

Das obige Beispiel zeigt das Signal U. Es ist während den ersten 5 Sekunden "logisch 0", wird dann für 1 Sekunde "logisch 1" und bleibt dann wieder 7 Sekunden lang "logisch 0".

Meistens interessiert der Zusammenhang von mehreren Signalen. Dann werden die Diagramme für die einzelnen Signale "zusammengepackt" mit einer gemeinsamen Zeitachse.

Wenn nur der Ablauf der Signalfolge interessiert, die Signaldauer aber nicht wichtig ist, kann auf die Skala der Zeitachse verzichtet werden. Auch die Beschriftung der Pegel "0" und "1" kann auf der senkrechten Achse entfallen.

### Zusammenhang zwischen Zeitablaufdiagramm und Wahrheitstabelle

(das „Zeitablaufdiagramm“ wird auch „Impulsdiagramm“ oder „Signal-Zeit-Plan“ genannt)

Die Wahrheitstabelle zeigt für jede mögliche Kombination der Eingangszustände die resultierenden Ausgangszustände. Das Zeitablaufdiagramm enthält noch mehr Information als die Wahrheitstabelle! Es zeigt auch noch den in der Anwendung vorkommenden zeitlichen Ablauf der Signale.

#### **Zeitablaufdiagramm aus der Wahrheitstabelle herleiten**

Rezept:

1. Die Eingangssignale müssen im Zeitablaufdiagramm gegeben sein!
2. Für jeden Zeitabschnitt den Zustand der Eingangssignale bestimmen.
3. Die entsprechende Zeile der Wahrheitstabelle bestimmen   
   und darin die zugehörigen Ausgangssignale ablesen.
4. Die Ausgangssignale im Impulsdiagramm einzeichnen.

Beispiel:



Beachten Sie die Abkürzungsmöglichkeit:

Im obigen Beispiel wird Z nur dann „1“, wenn U=„0“ und V=„1“ ist.

**Es genügt, diese Eingangskombination zu suchen, in allen anderen**

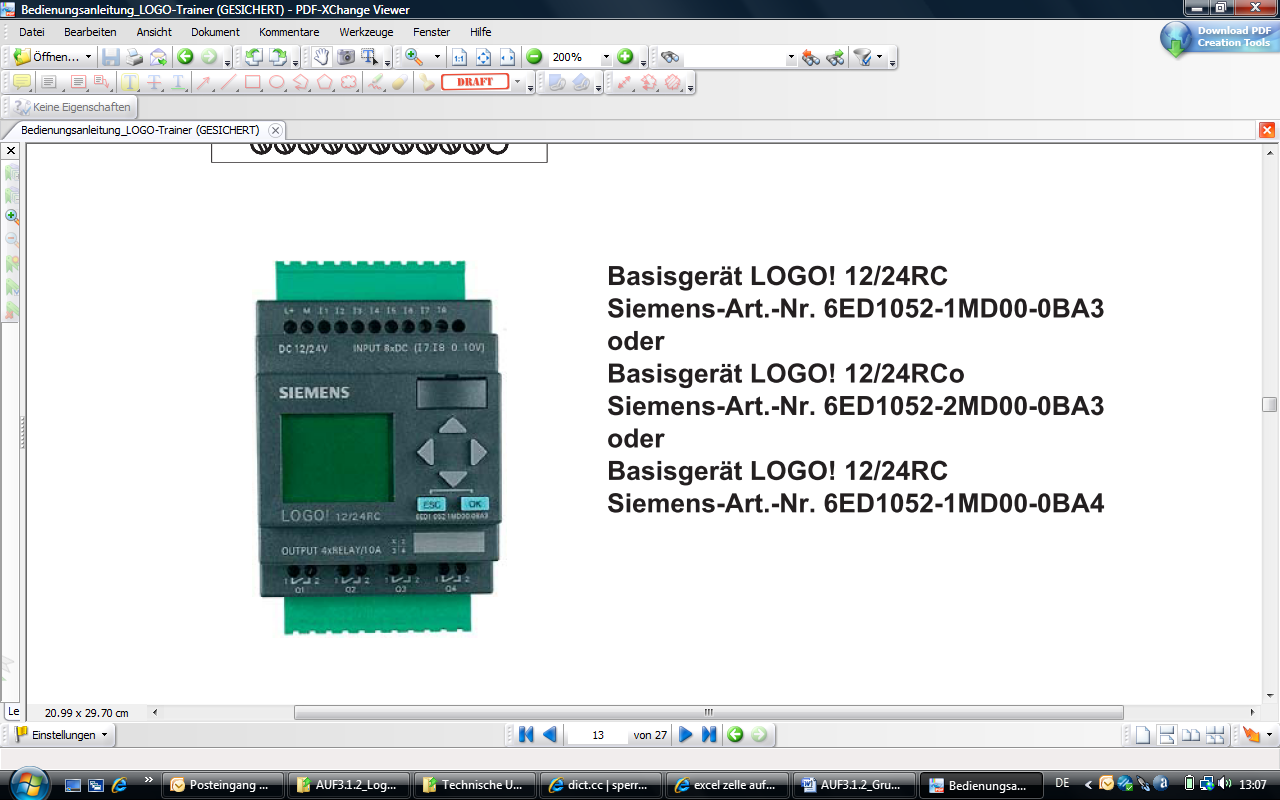
**Fällen ist Z = „0“!**

#### **Wahrheitstabelle aus dem Zeitablaufdiagramm herleiten**

Rezept: Umgekehrtes Vorgehen wie oben!

## Steuergeräte

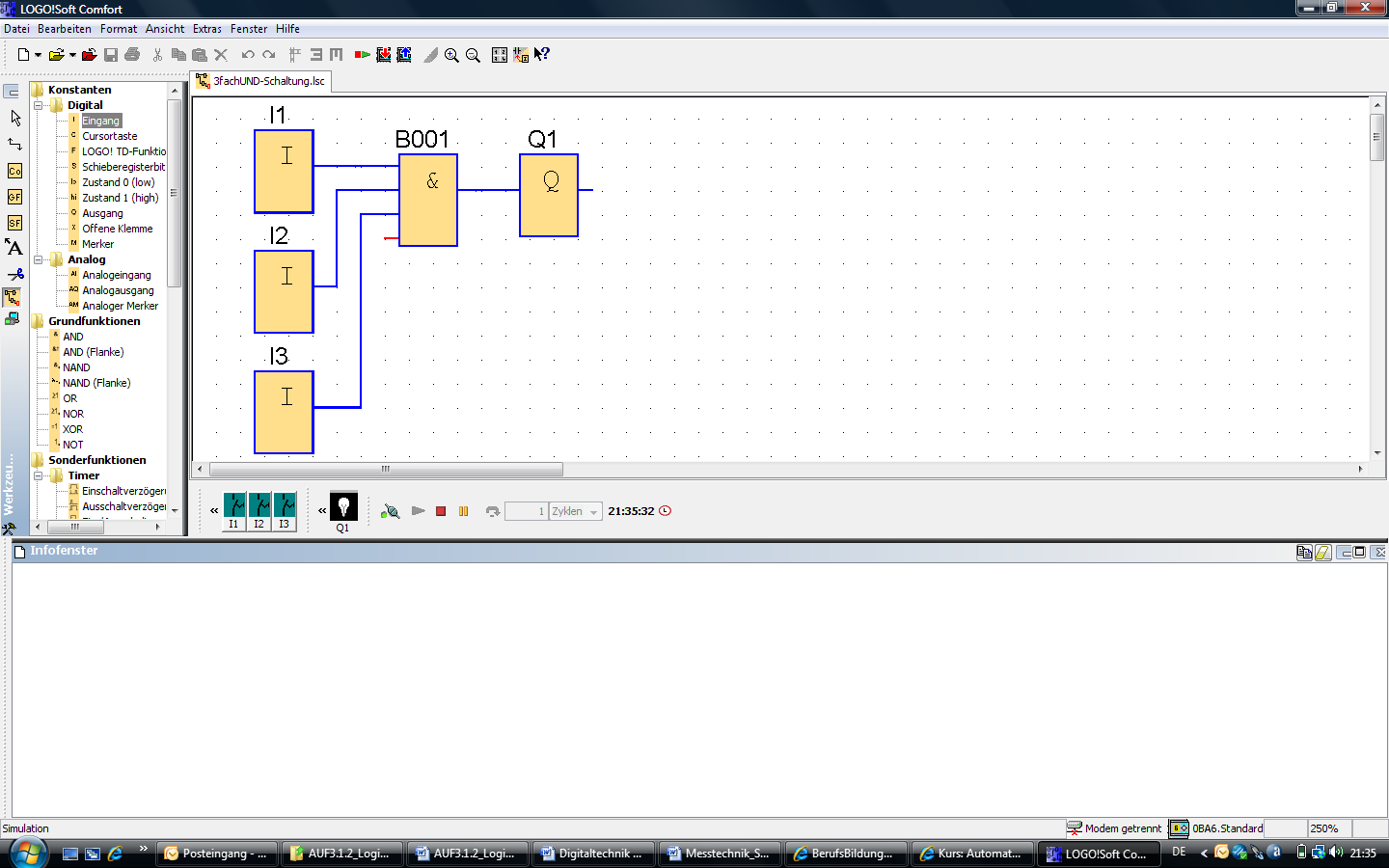
Digitale Steuerungen werden in der Praxis mit verschiedensten Technologien realisiert. Angefangen bei "Relaisschränken" über "Elektronikplatinen" (mit "Integrierten Schaltungen") bis zu "Speicher­program­mier­­baren Steuerungen" ("SPS") kommen die unterschiedlichsten Bauarten vor.

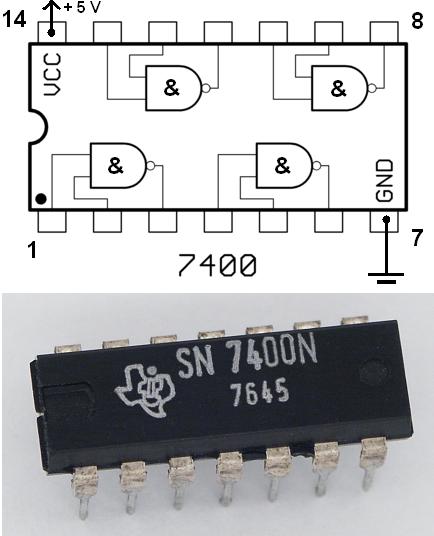
Eine vielseitig einsetzbare und programmierbare Steuerung ist die LOGO!, eine Kleinsteuerung von der Firma Siemens. Die LOGO! ist geeignet zur Lösung einfacher Automatisierungs­aufgaben wie z.B. die Steuerung eines Garagentors.

Das abgebildete Grundmodul verfügt über 8 digitale Eingänge für den Anschluss von Sensoren (Schlüsselschalter, Lichtschranken, Berührungsmelder, ....). 2 dieser Eingänge können wahlweise auch als Analogeingänge (0 bis 10 V) verwendet werden. Zur Ansteuerung von Aktoren stehen 4 Relaisausgänge zur Verfügung.

Für komplexere Aufgaben kann die LOGO! durch Module erweitert werden (max. 24 digitale Eingänge, 16 digitale Ausgänge, 8 analoge Eingänge, 2 analoge Ausgänge, ASI/EIB-Kommunikationsmodule). Zusätzlich kann ein Textdisplay angesteuert werden.

Dank der entsprechenden Programmierung kann die universelle Steuerung die verschiedensten Aufgaben erfüllen.

Das Programm kann direkt mit den Pfeiltasten des LOGO!-Moduls eingegeben werden. Wesentlich komfortabler und empfehlenswert ist die Programmierung am Laptop. Im Unterricht verwenden wir die Programmiersoftware **LOGO!Soft Comfort**. Am Bildschirm kann die Schaltung direkt als Schema eingegeben werden. Nebst den 8 Grundfunktionen (UND, ODER, usw.) sind auch 28 Sonderfunktionen (z.B. Einschaltverzögerung, Ausschaltverzögerung, Betriebsstundenzähler, Zähler, Selbsthalterelais, PI-Regler, usw.) verfügbar.

In some practicals we will use transistor-transistor logic (TTL) integrated circuits. One of the most common TTL integrated circuits is the 7400. It may also be listed as 74LS00, 74HC00, 74HCT00. It is sometimes identified as a quad two-input NAND as it consists of four two-input gates. When using any TTL integrated circuits it is important that the supply voltage is maintained at 5 V and connected with the correct polarity.

# http://upload.wikimedia.org/wikipedia/commons/2/26/7400.jpg

## http://www.homestead.com/~media/elements/Clipart/office/laptop_work.jpgVocabulary

Complete the wordlist using an online dictionary. Use the website <http://www.dict.cc/>.

|  |  |
| --- | --- |
| English: | German: |
| Prctical | Praktikum |
| Integrated circuits | integrierte Schaltungen |
| Quad | vierfach |
| (to) consist of | bestehen aus |
| Gate | Gatter |
| (to) maintain | beibehalten |
| Digital Technology | Digitaltechnik |
| binary | binär |
| time sequence chart | Zeitablaufdiagramm |
| Logic | Logik |
| logic level | Logikzustand |
| Dual in-line package (DIP) | DIP-Gehäuse |
| Dual in-line | Doppelreihig |
| Pluug-in package | Steckgehäuse |
| rectangular | rechteckig |
| housing | Gehäuse |
| Parallel rows | Parallele Reihen |
| Connecting pins | Anschlusspins |
| Pin | Beinchen, Stift, Pin |
| Socket | Sockel,Fassung |
| Counter- clockwise | Entgegen dem Uhrzeigersinn |
| Conclusion | Zusammenfassung |
| … | … |
| … | … |
| … | … |
| … | … |
| … | … |
| … | … |
| … | … |

# Logische Grundbausteine

## Die UND-Verknüpfung (Konjunktion)

### Funktionsbeschreibung

Ein UND – Glied liefert nur dann am Ausgang das 1-Signal, wenn alle Eingänge gleichzeitig 1-Signal führen.

### Schaltzeichen

Ein UND – Glied wird durch dieses Symbol wiedergegeben:

&

Q1

I1

I2

### Funktionsgleichung

*(lies: Q1 gleich I1 UND I2)*

∧ bedeutet: UND – Verknüpfung mit ….

### Zeitablaufdiagramm

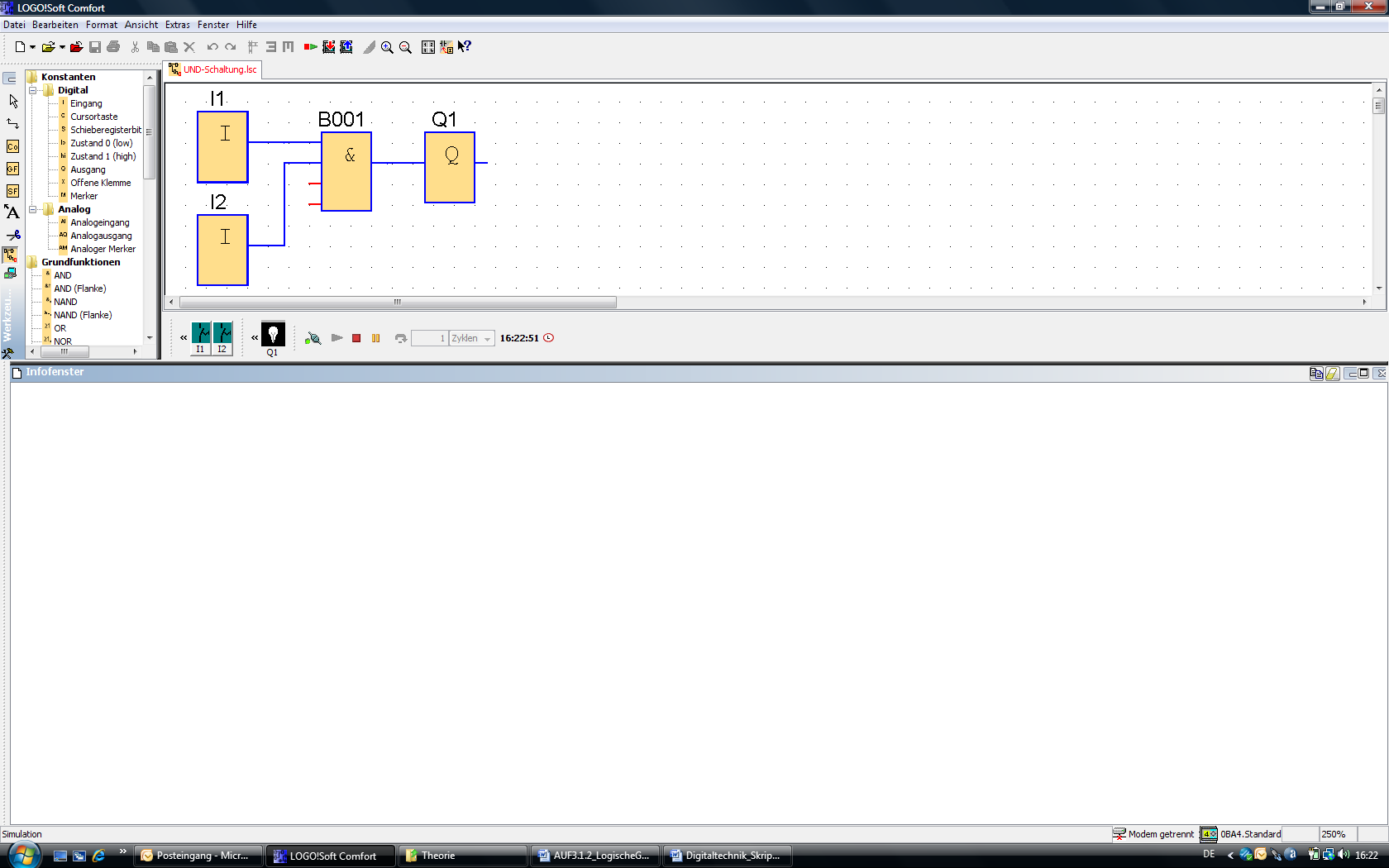
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I1 |  |  |  |  |  |
|  |  |  |  |  |  |
| I2 |  |  |  |  |  |
|  |  |  |  |  |  |
| Q1 |  |  |  |  |  |

### Experimentelle Überprüfung

Programmieren Sie mit LOGO!Soft Comfort eine UND-Verknüpfung mit 2 Eingängen. Prüfen Sie durch Simulation nach, ob die Funktionsbeschreibung der UND-Verknüpfung stimmt. Erarbeiten Sie die vollständige Wahrheitstabelle, indem alle möglichen Eingangskombinationen eingestellt werden und die Reaktion des Ausganges beobachtet wird.

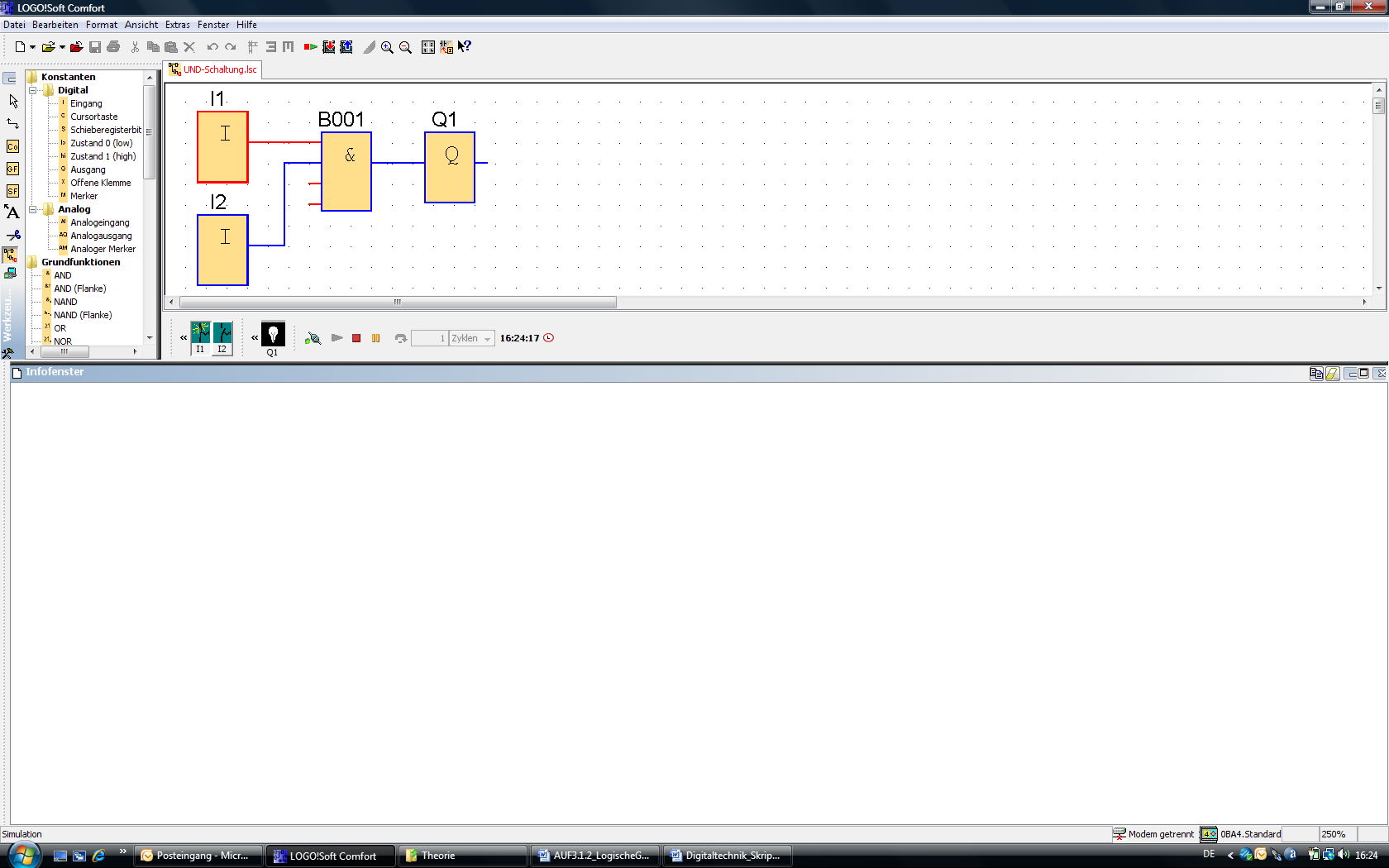
Überprüfung von Zeile 1 der Wahrheitstabelle:

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 0 |
|  |  |  |
|  |  |  |
|  |  |  |

Beide Eingänge sind nicht betätigt und führen somit 0-Signal. Der Ausgang führt ebenfalls 0-Signal. Die „Lampe“ am Ausgang Q1 leuchtet nicht.

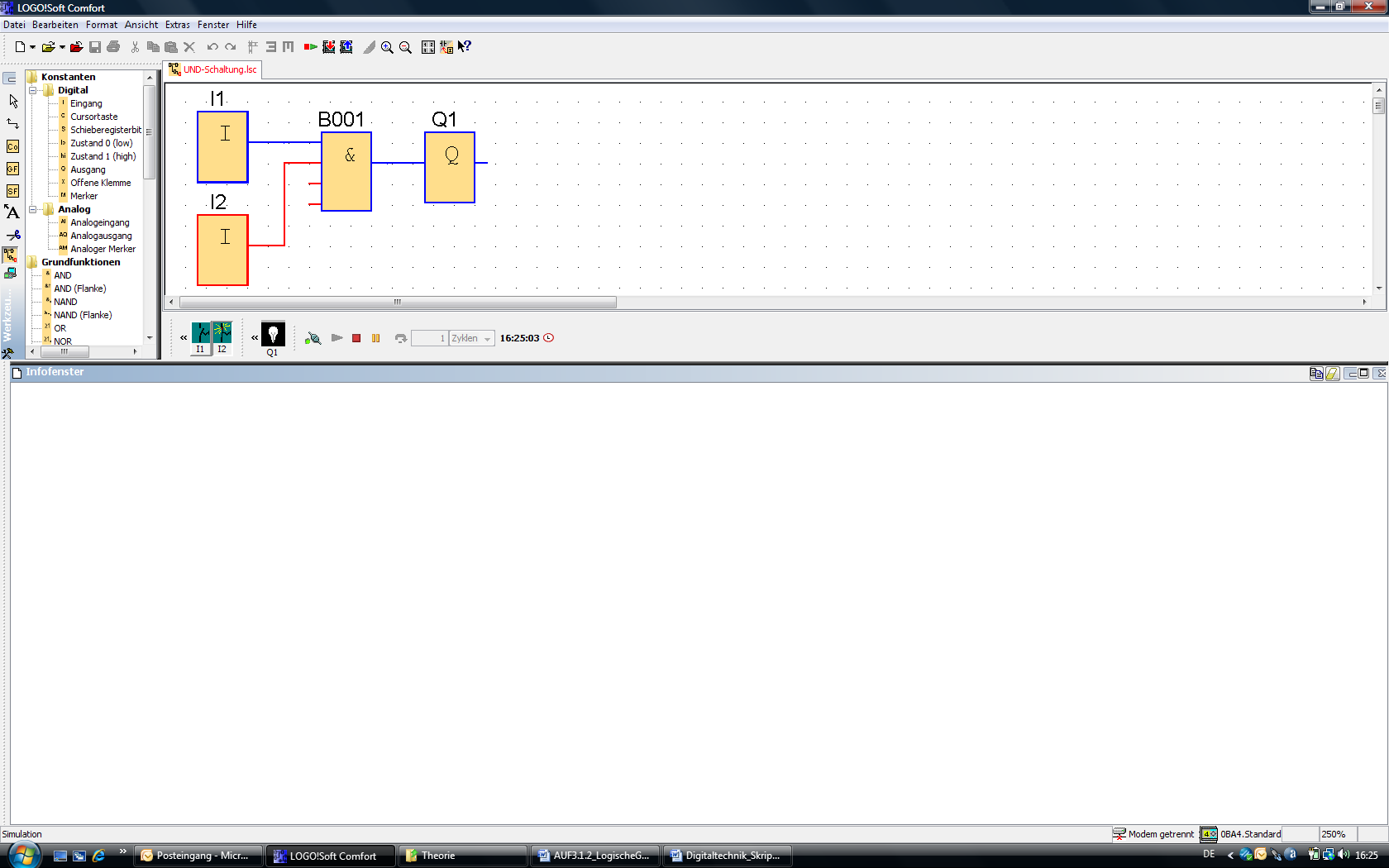
Überprüfung von Zeile 2 der Wahrheitstabelle:

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
|  |  |  |
|  |  |  |

Nun wird Eingang I1 betätigt und führt somit 1-Signal. I2 bleibt unbetätigt. Der Ausgang führt immer noch 0-Signal. Die „Lampe“ am Ausgang Q1 leuchtet nicht.

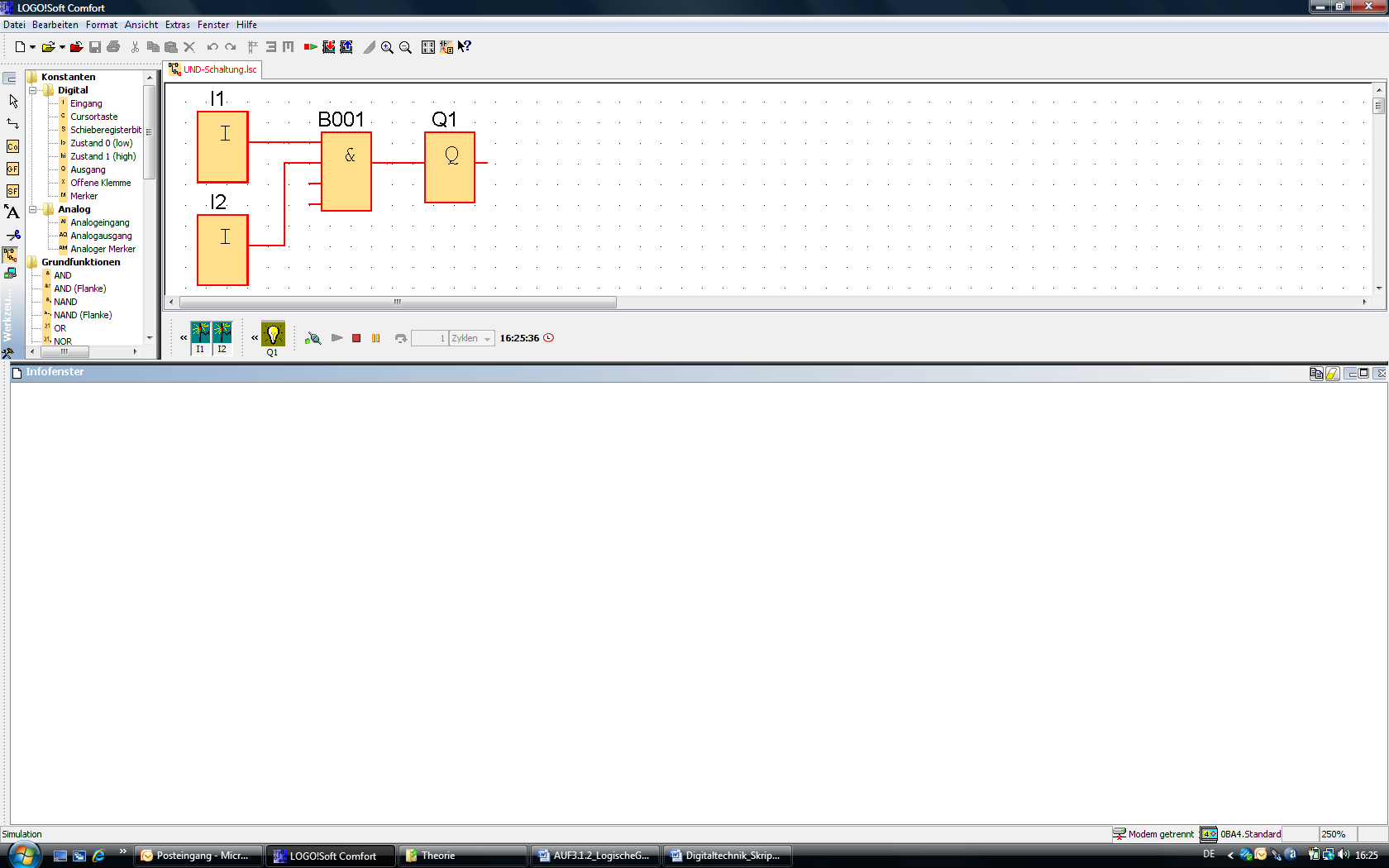
Überprüfung von Zeile 3 der Wahrheitstabelle:

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
|  |  |  |

Nun wird Eingang I2 betätigt und führt somit 1-Signal. I1 bleibt unbetätigt. Der Ausgang führt immer noch 0-Signal. Die „Lampe“ am Ausgang Q1 leuchtet nicht.

Überprüfung von Zeile 4 der Wahrheitstabelle:

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Nun werden beide Eingänge betätigt und führen somit 1-Signal. Der Ausgang wechselt nun auf 1-Signal und die „Lampe“ am Ausgang Q1 leuchtet.

## Die ODER-Verknüpfung

### Funktionsbeschreibung

Der Ausgang führt nur dann 1-Signal, wenn ein oder mehrere Eingänge 1-Signal führen.

### Schaltzeichen

≥1

Q1

I1

I2

### Funktionsgleichung

*(lies: Q1 gleich I1 ODER I2)*

v bedeutet: ODER – Verknüpfung mit ….

### Zeitablaufdiagramm

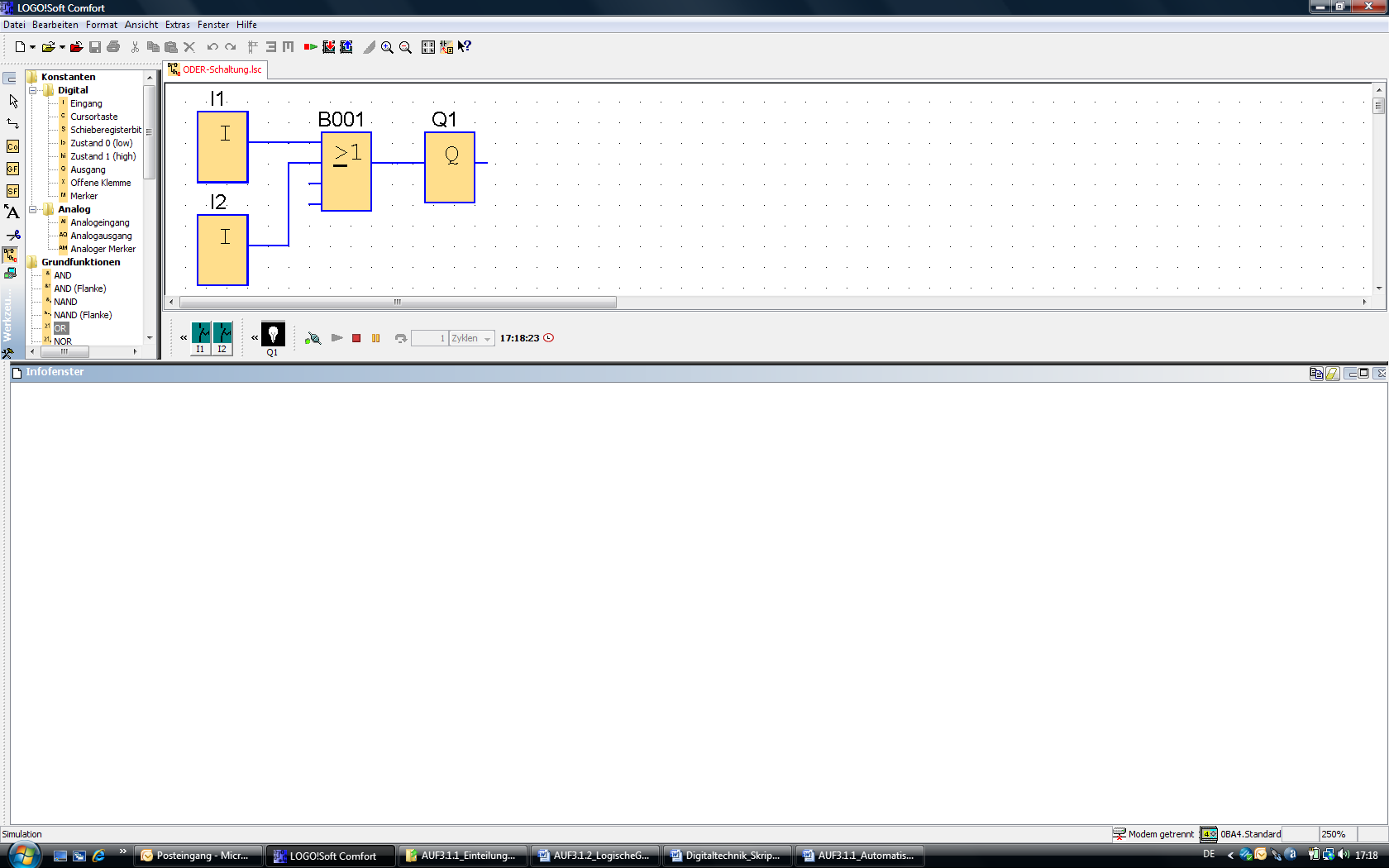
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I1 |  |  |  |  |  |
|  |  |  |  |  |  |
| I2 |  |  |  |  |  |
|  |  |  |  |  |  |
| Q1 |  |  |  |  |  |

### Experimentelle Überprüfung

Programmieren Sie mit LOGO!Soft Comfort eine ODER-Verknüpfung mit 2 Eingängen. Prüfen Sie durch Simulation nach, ob die Funktionsbeschreibung der ODER-Verknüpfung stimmt. Erarbeiten Sie die vollständige Wahrheitstabelle, indem alle möglichen Eingangskombinationen eingestellt werden und die Reaktion des Ausganges beobachtet wird.

Ergänzen Sie die Wahrheitstabelle:

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



## Die NICHT-Verknüpfung (Negation)

Die NICHT – Verknüpfung ist einfach, jedoch unentbehrlich. Die Schaltung heisst auch **INVERTER**.

### Funktionsbeschreibung

Der Ausgang führt 1-Signal, wenn der Eingang 0-Signal führt und umgekehrt.

### Schaltzeichen

1

Q1

I1

### Funktionsgleichung

*(lies: Q1 gleich NICHT I1)*

Der Strich bedeutet: Variable wird invertiert.

### Zeitablaufdiagramm

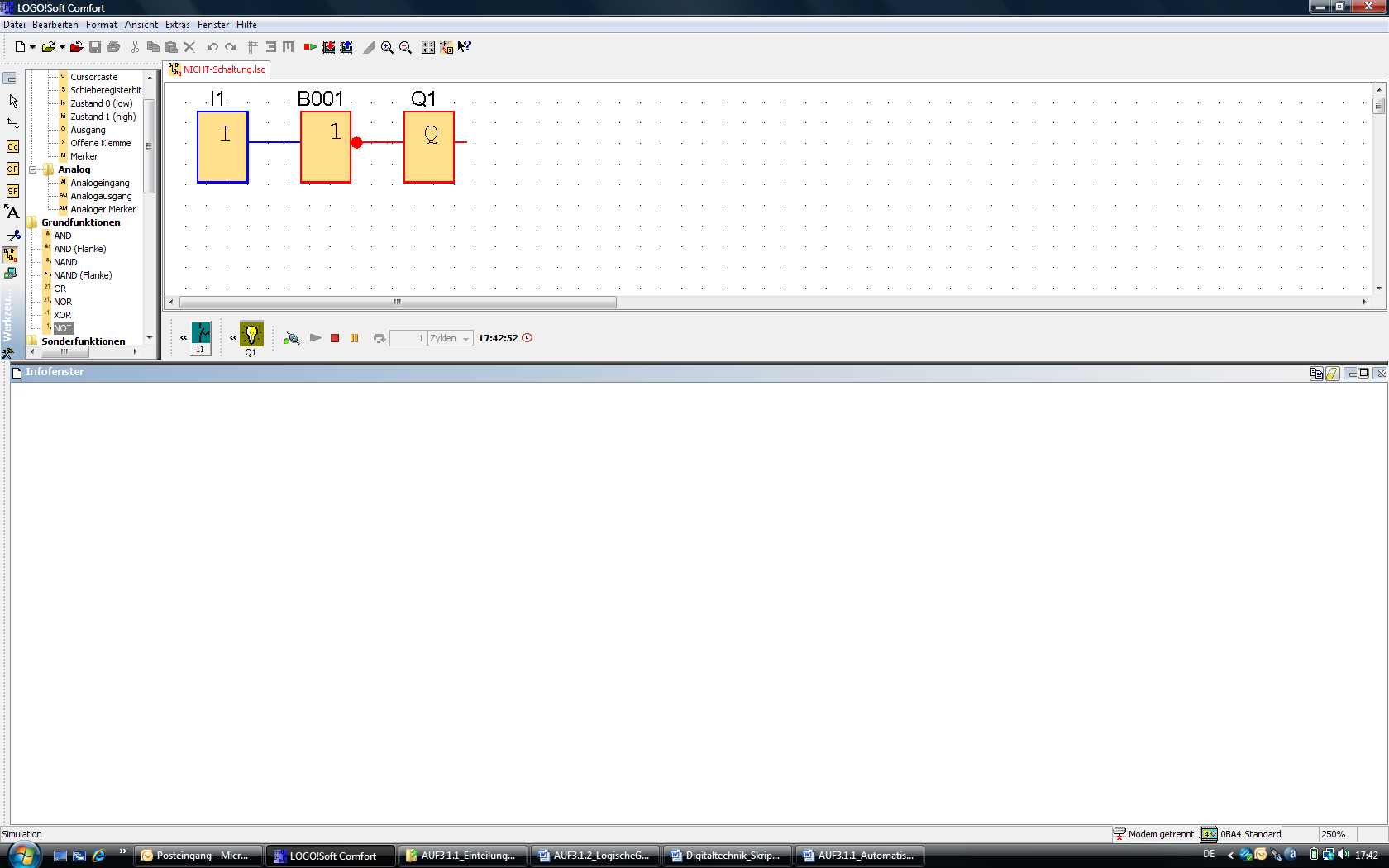
|  |  |  |  |
| --- | --- | --- | --- |
| I1 |  |  |  |
|  |  |  |  |
| Q1 |  |  |  |

### Experimentelle Überprüfung

Programmieren Sie mit LOGO!Soft Comfort einen Inverter. Prüfen Sie durch Simulation nach, ob die Funktionsbeschreibung des Inverters stimmt. Erarbeiten Sie die vollständige Wahrheitstabelle, indem alle möglichen Eingangskombinationen eingestellt werden und die Reaktion des Ausganges beobachtet wird.

Ergänzen Sie die Wahrheitstabelle:

|  |  |
| --- | --- |
| I1 | Q1 |
| 0 | 1 |
| 1 | 0 |



# Binary logic operations with more than 2 inputs

## The AND function (conjunction)

### Functional description

If we examine the truth table for the AND function, we can see that the output of logic 1 is obtained only when all the inputs are logic 1.

### Circuit symbol

Common symbol used for AND gates:

&

Q1

I1

I3

I2

### Boolean expression

*(read: Q1 equal I1 AND I2 AND I3)*

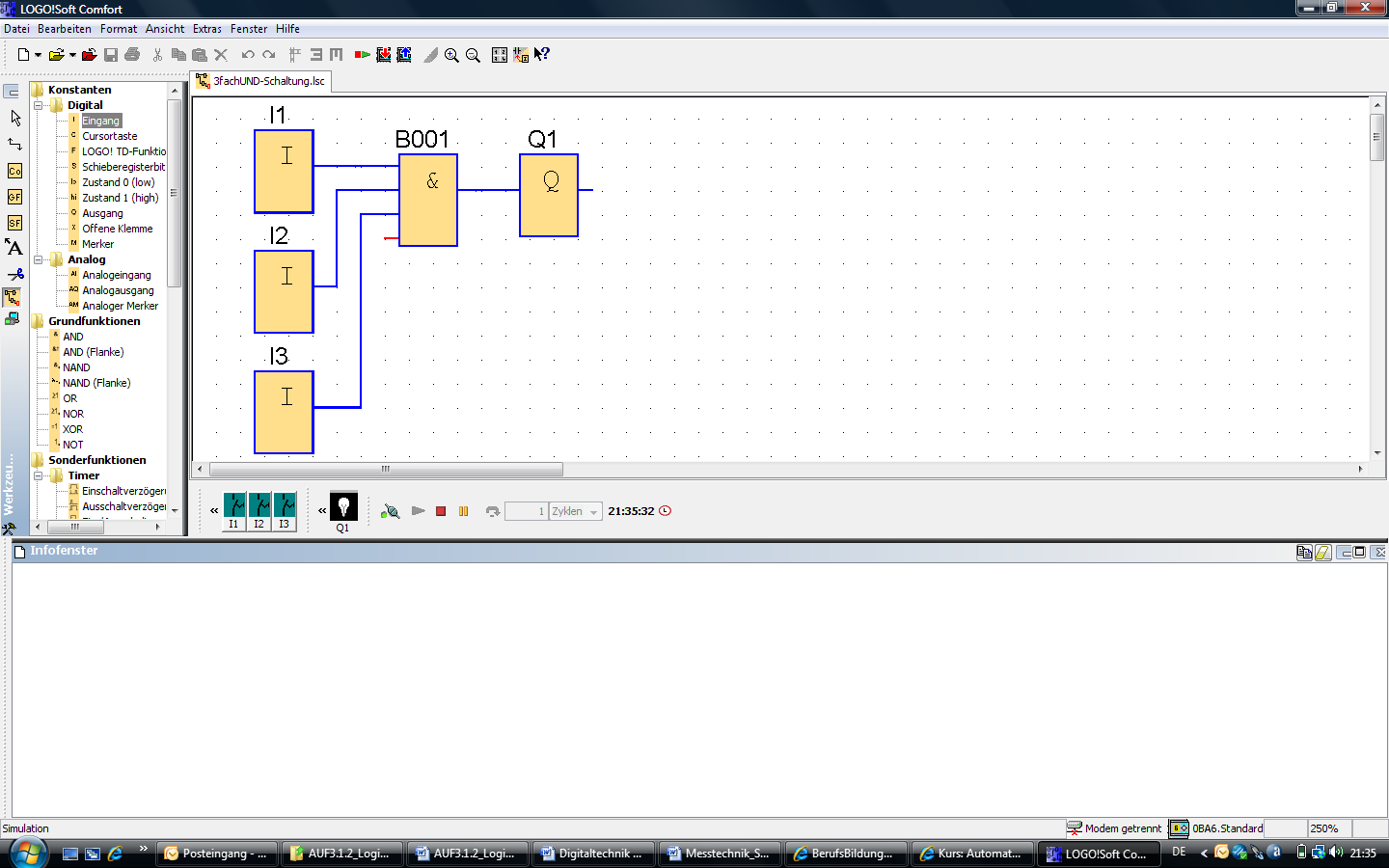
AND is denoted in Boolean algebra by a **∧** between the relevant variables.

### Time sequence chart

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I1 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| I2 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| I3 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| Q1 |  |  |  |  |  |  |  |  |  |

### Experimental check

Use LOGO!Soft Comfort and draw the circuit of an AND function with 3 inputs. Use I1, I2 and I3 to simulate 0 or 1 at the input and complete the truth table!

Circuit: Truth table:

|  |  |  |  |
| --- | --- | --- | --- |
| I3 | I2 | I1 | Q1 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

## The OR function (disjunction)

### Functional description

If we examine the truth table for the OR function, we can see that an output of logic 1 is obtained when any of its inputs is at logic 1.

### Circuit symbol

≥1

I1

I2

Q1

I3

I4

### Boolean expression

*(read: Q1 equal I1 OR I2 OR I3 OR I4)*

OR is denoted in Boolean algebra by a **v** between the relevant variables.

### Time sequence chart

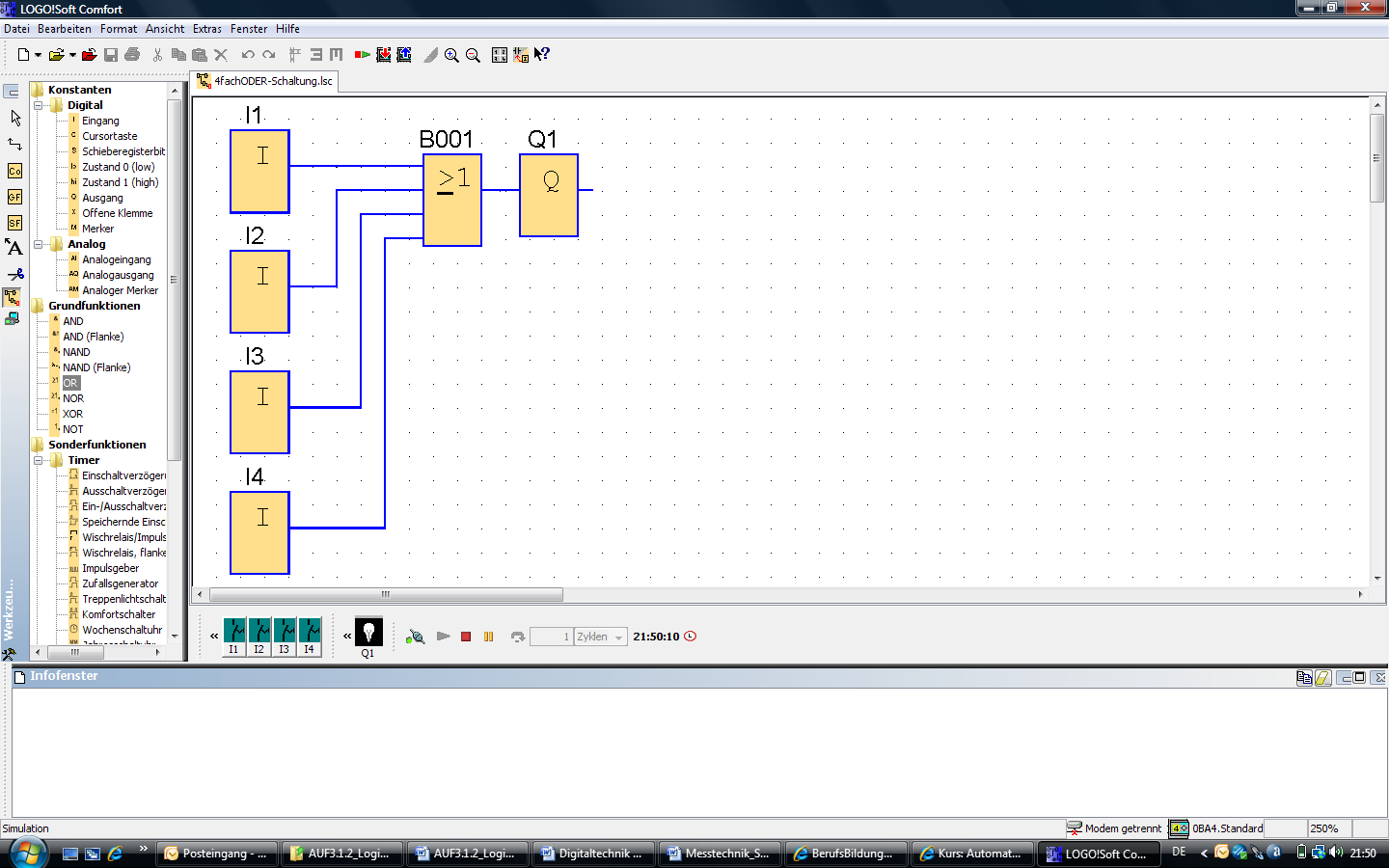
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| I2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| I3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| I4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Q1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### Experimental check

Use LOGO!Soft Comfort and draw the circuit of an OR function with 4 inputs. Use I1, I2, I3 and I4 to simulate 0 or 1 at the input and complete the truth table!

Circuit: Truth table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| I4 | I3 | I2 | I1 | Q1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |



## Task

A truth table for a function with two input variables has 4 lines because there are 4 possible combinations of the input variables. How many lines are there in a truth table for a function with 3,4,5 and 6 input variables?



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| number of inputs | 2 | 3 | 4 | 5 | 6 |
| number of combinations | 4 | 8(23 | 1624 | 2225 | 6426 |

# Derived Basic Functions

## The NAND function

The NAND function gives us a truth table which is the inverse of the AND function. The operation may be thought of as NOT AND.

### Functional description

AND function with inverse output.

### Circuit symbol

&

Q1

I1

I2

### Boolean expression

*(read: Q1 equal I1NAND I2)*

### Time sequence chart

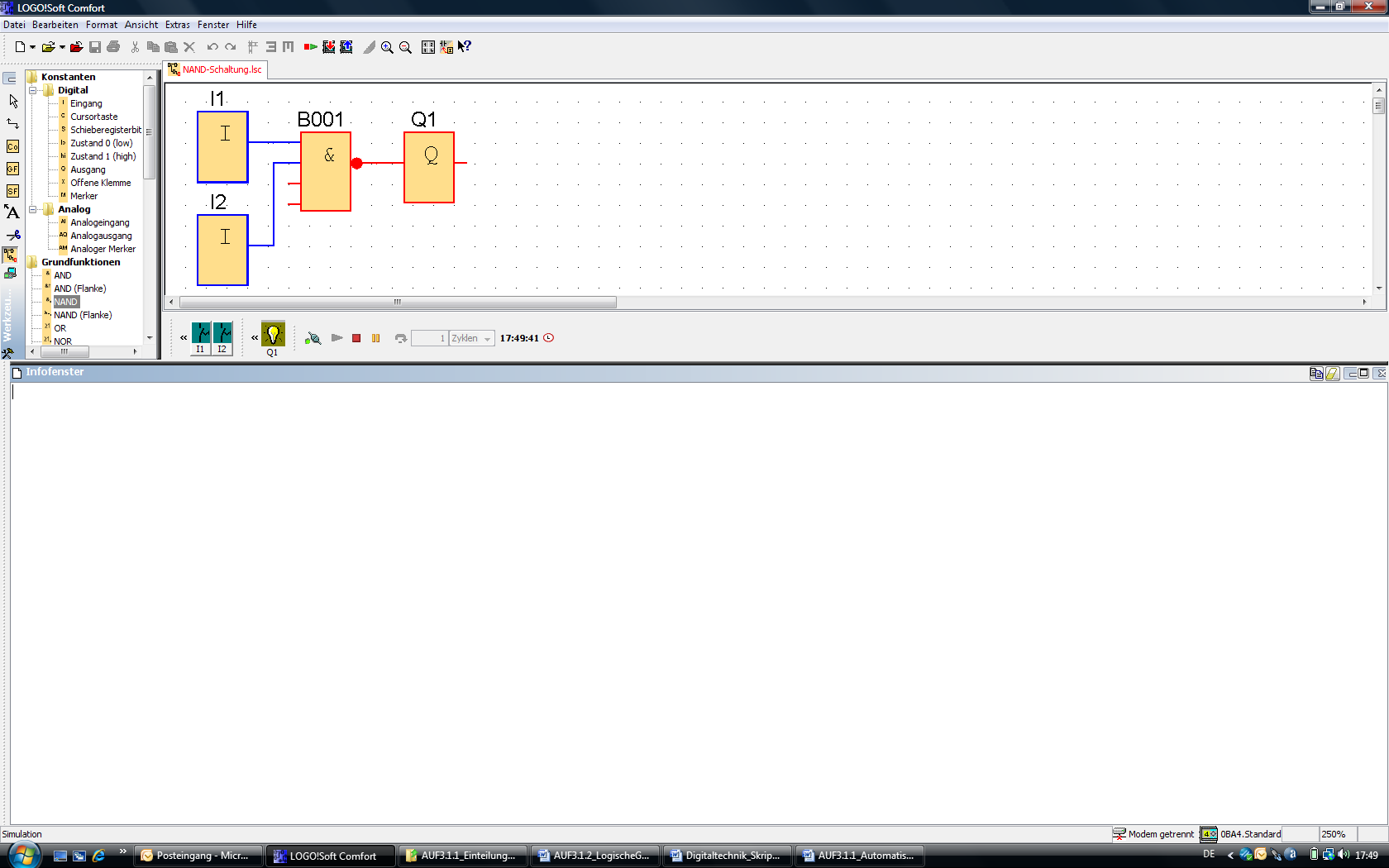
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I1 |  |  |  |  |  |
|  |  |  |  |  |  |
| I2 |  |  |  |  |  |
|  |  |  |  |  |  |
| Q1 |  |  |  |  |  |

### Experimental check

Use LOGO!Soft Comfort and draw the circuit of a NAND function with 2 inputs. Use I1 and I2 to simulate 0 or 1 at the input and complete the truth table!

Circuit: Truth table:

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



## The NOR function

The NOR function is the inverse of the OR function.

### Functional description

OR function with inverse output.

### Circuit symbol

≥1

Q1

I1

I2

### Boolean expression

*(read: Q1 equal I1NOR I2)*

### Time sequence chart

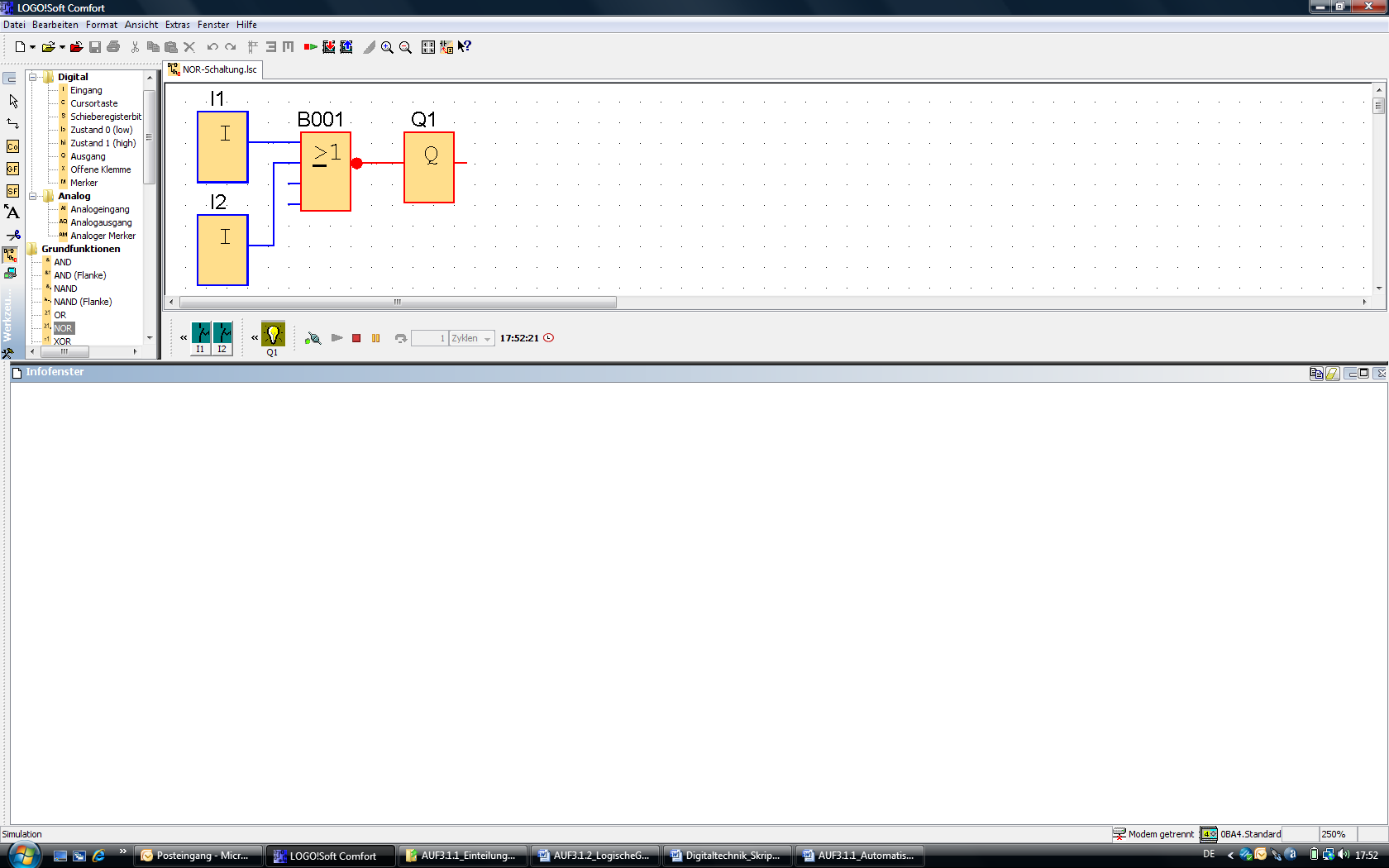
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I1 |  |  |  |  |  |
|  |  |  |  |  |  |
| I2 |  |  |  |  |  |
|  |  |  |  |  |  |
| Q1 |  |  |  |  |  |

### Experimental check

Use LOGO!Soft Comfort and draw the circuit of a NOR function with 2 inputs. Use I1 and I2 to simulate 0 or 1 at the input and complete the truth table!

Circuit: Truth table:

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



## Die XOR-Verknüpfung (Antivalenz oder Exklusiv-ODER)

### Funktionsbeschreibung

Der Ausgang führt nur dann 1-Signal, wenn nur einer der beiden Eingänge 1-Signal führt.

### Schaltzeichen

=1

Q1

I1

I2

### Funktionsgleichung

*(lies: Q1 gleich I1XOR I2)*

### Zeitablaufdiagramm

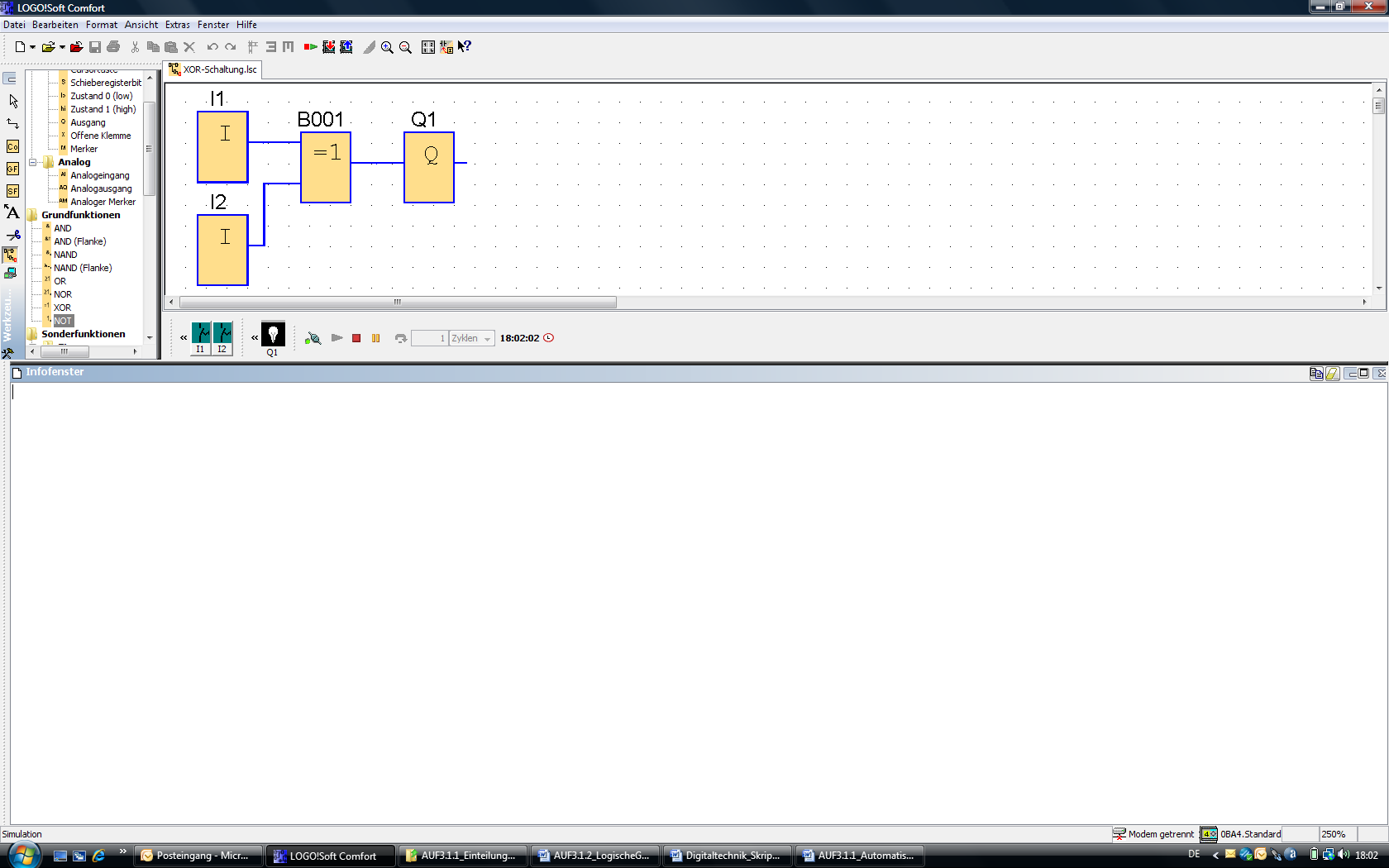
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I1 |  |  |  |  |  |
|  |  |  |  |  |  |
| I2 |  |  |  |  |  |
|  |  |  |  |  |  |
| Q1 |  |  |  |  |  |

### Experimentelle Überprüfung

Programmieren Sie mit LOGO!Soft Comfort eine XOR-Verknüpfung mit 2 Eingängen. Prüfen Sie durch Simulation nach, ob die Funktionsbeschreibung der XOR-Verknüpfung stimmt. Erarbeiten Sie die vollständige Wahrheitstabelle, indem alle möglichen Eingangskombinationen eingestellt werden und die Reaktion des Ausganges beobachtet wird.

Ergänzen Sie die Wahrheitstabelle:

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



## Die XNOR-Verknüpfung (Äquivalenz oder Exklusiv-NOR)

### Funktionsbeschreibung

Der Ausgang führt nur dann 1-Signal, wenn sich alle Eingänge im gleichen Zustand befinden.

### Schaltzeichen

=

Q1

I1

I2

### Funktionsgleichung

*(lies: Q1 gleich I1XNOR I2)*

### Zeitablaufdiagramm

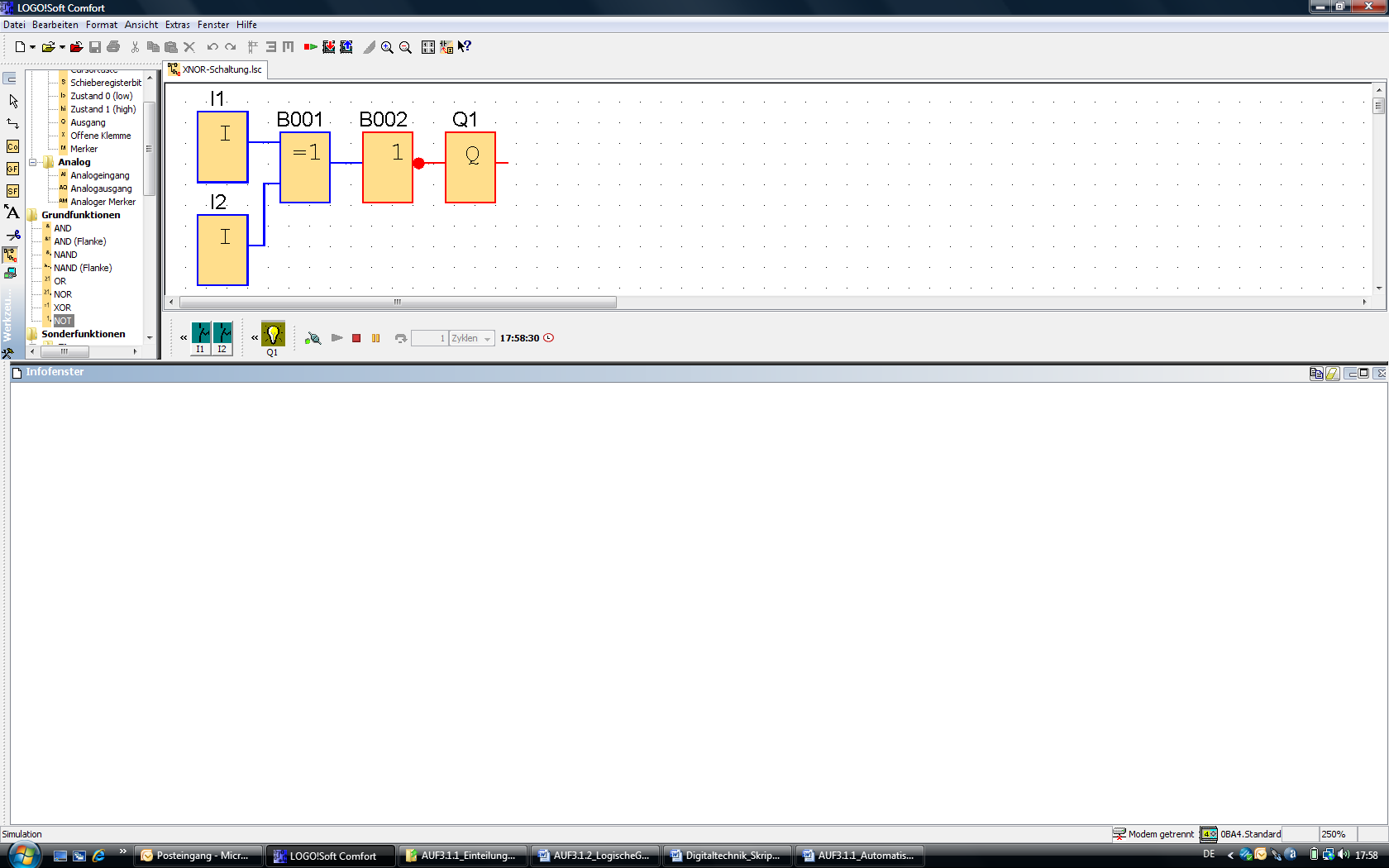
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I1 |  |  |  |  |  |
|  |  |  |  |  |  |
| I2 |  |  |  |  |  |
|  |  |  |  |  |  |
| Q1 |  |  |  |  |  |

### Experimentelle Überprüfung

Programmieren Sie mit LOGO!Soft Comfort die Funktion einer XNOR-Verknüpfung mit 2 Eingängen. **Beachten Sie, dass diese Funktion nicht als „fertige Funktion“ in LOGO!Soft Comfort vorliegt**. Prüfen Sie durch Simulation nach, ob die Funktionsbeschreibung der XNOR-Verknüpfung stimmt. Erarbeiten Sie die vollständige Wahrheitstabelle, indem alle möglichen Eingangskombinationen eingestellt werden und die Reaktion des Ausganges beobachtet wird.

Ergänzen Sie die Wahrheitstabelle:

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



## http://www.homestead.com/~media/elements/Clipart/office/laptop_work.jpgVocabulary

Complete the English and German terms using the book of tables and an online dictionary. (<http://www.dict.cc/>)

|  |  |
| --- | --- |
| English: | German: |
| Binary logic operations | binäre Verknüpfungen |
| AND (conjunction) | UND (Konjunktion) |
| OR (disjunction) | ODER (Disjunktion) |
| NOT (inverter) | NICHT (Inverter) |
| functional description | Funktionsbeschreibung |
| boolean expression | Funktionsgleichung/Boleesche Ausdruck |
| circuit symbol | Schaltzeichen |
| experimental check | Experimentelle Überprüfung |
| equal | Gleich |
| (to) obtain | Erreichen |
| Denoted | gekennzeichnet |
| (to) simulate | simulieren |
| inverse | Negieren |
| NAND | NAND |
| NOR | NOR |
| Exclusive OR, ExOR, XOR | Exclusiv OR, ExOR, XOR |
| Exclusive NOR, ExNOR, XNOR | Exclusiv NOR, ExNOR, XNOR |
| Apparatus | Gerät |
| Method | Vorgehen, Methode |
| (to) derive | Ableiten |
| Derived basic functons | Abgeleitete Grundfunktionnen |
| Pinout | Pinbelegung |
| … | … |
| … | … |
| … | … |
| … | … |
| … | … |
| … | … |

# Binary logic operations with inverse inputs

## Task

Use LOGO!Soft Comfort and draw the binary logic functions with inverse inputs as shown below. Use I1 and I2 to simulate 0 or 1 at the input and complete the truth table! Choose an equivalent logical symbol in the table and mark with „X“!



≥1

Q1

I1

I2

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| … | X | … |

&

Q1

I1

I2

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| … | … | X |

≥1

Q1

I1

I2

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| X | … | … |