Steuerungstechnik Workshop

Teil I

Logikfunktionen

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Ziel des Workshops

Der Workshop Logikfunktionen soll die-Grundkenntnisse der logischen Grundschaltungen vermitteln.

Als Schwerpunkt werden die logischen Grundschaltungen mit Schaltzeichen und Funktionstabelle vorgestellt. Zum besseren Verständnis über das Zusammenspiel von Logikfunktionen steht der Einsatz von programmierbaren Logikmodulen und die Simulation der Übungsaufgaben. Es werden die Programmiermethoden Funktionsplan, Kontaktplan und Strukturierter Text vorgestellt.

Mit den vermittelten Kennnissen ist es dem Teilnehmer abschließend möglich, einfache Schaltungen zu erstellen, diese in ein Logikmodul zu laden und zu testen.

Teil I

Logik-Grundschaltungen

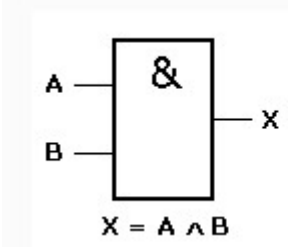
In Digitalschaltungen werden logische Steuersignale zusammengeführt und nach der Booleschen Algebra ausgewertet. George Boole entwickelte um 1850 eine Algebra, die mit zwei Konstanten 0 und 1 und den Operatoren UND, ODER sowie NICHT auskommt. Jede Variable kann nur zwischen den Werten 0 und 1 wechseln. Das Ergebnis ist ebenfalls als Variable zu betrachten und kann daher auch nur die Werte 0 oder 1 annehmen.

Operator		Symbol	Beispiel
UND	AND	∧	A ∧ B
ODER	OR	∨	A ∨ B
NICHT	NOT	¯	$\overline{A}$

UND-Verknüpfung

Für eine UND-Verknüpfung, im englischen Sprachgebrauch AND, sind mindestens zwei Eingangsvariablen notwendig. Die Funktion liefert als Ergebnis eine Ausgangsvariable. Bei zwei binären Zuständen am Eingang sind 4 unterschiedliche Kombinationen möglich. Sie werden nach Eingangsvariablen getrennt in einer Funktionstabelle mit 0 und 1 erfasst. In der letzten Spalte ist das Funktionsergebnis notiert.

Bei einem logischen UND (AND) müssen alle Eingangswerte wahr oder 1 sein, damit das Ergebnis wahr oder 1 ergibt. Stellt man sich die Eingangsvariablen als in Reihe liegende Schalter vor, so kann bei anliegender Spannung nur dann Strom fließen, wenn in diesem Beispiel beide Schalter gleichzeitig geschlossen sind.



Schaltzeichen

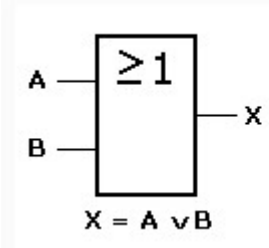
Elektromechanisches Schaltprinzip

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

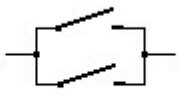
Funktionstabelle

ODER-Verknüpfung

Für eine ODER-Verknüpfung, im englischen Sprachgebrauch OR, sind mindestens zwei Eingangsvariable notwendig. Es gibt ebenso viele unterschiedliche Eingangskombinationen wie beim UND-Gatter. Die Ausgangsvariable ist immer dann wahr oder 1, wenn mindestens eine Eingangsvariable wahr oder 1 ist. Im elektromechanischen Schaltprinzip kann das Verhalten durch zwei im Stromkreis parallel angeordnete Schalterelemente erreicht werden. Bei anliegender Spannung fließt Strom, wenn einer der Schalter geschlossen ist.



Schaltzeichen



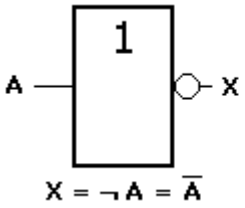
Elektromechanisches Schaltprinzip

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

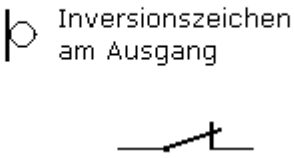
Funktionstabelle

NICHT-Verknüpfung

Die dritte Grundverknüpfung ist die Negation. Für ein NICHT-Gatter, im englischen Sprachgebrauch NOT, ist nur eine Eingangsvariable notwendig. Das Ausgangssignal kehrt den Eingangszustand um. Das NICHT-Gatter funktioniert als Inverter. Im elektromechanischen Schaltprinzip liegt im Stromkreis ein Öffner, der ein genormtes Schaltzeichen hat.



Schaltzeichen



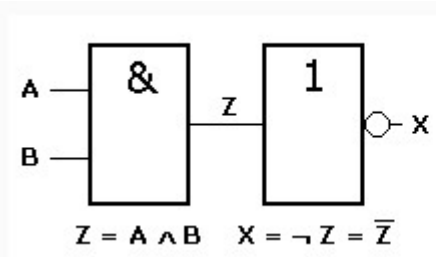
Elektromechanisches Schaltprinzip

A	X
0	1
1	0

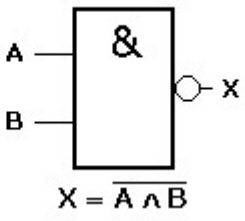
Funktionstabelle

NAND-Verknüpfung

Eine Kombination aus einem UND-Gatter mit nachfolgendem NICHT-Gatter ergibt ein NAND-Gatter. Die Ausgangsvariable Z des UND-Gatters wird durch das NICHT-Gatter negiert und erzeugt die Ausgangsvariable X des NAND-Glieds. Viele logische Funktionen lassen sich durch den Einsatz von NAND-Gattern lösen. Oft steigt dadurch die Anzahl der notwendigen Gatter, mit dem wirtschaftlichen Vorteil nur einen Gattertyp zu verwenden.



UND-NICHT Schaltung

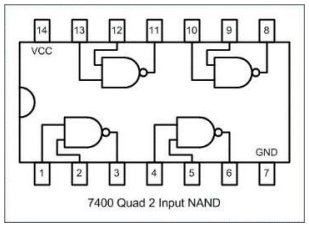


NAND-Schaltzeichen

A	B	Z	X
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

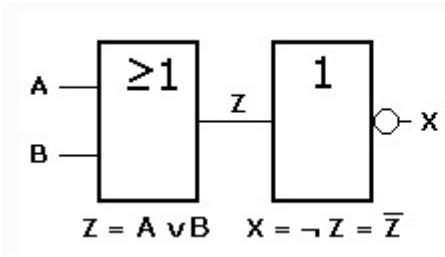
Funktionstabelle

Datenblatt oder auch Datasheet von einem SN7404N von Texas Instruments mit vier NAND-Gatter.

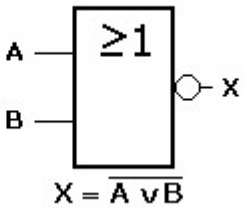


NOR-Verknüpfung

Eine Kombination aus einem ODER-Gatter mit nachfolgendem NICHT-Gatter ergibt ein NOR-Gatter. Die Ausgangsvariable Z des ODER-Gatters wird durch das NICHT-Gatter negiert und erzeugt die Ausgangsvariable X des NOR-Glieds. NOR-Glieder haben in logischen Schaltungen die gleiche wichtige Bedeutung wie NAND-Glieder.



OR-NICHT Schaltung



NOR-Schaltzeichen

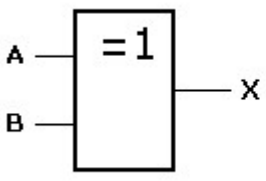
A	B	Z	X
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Funktionstabelle

XOR-Verknüpfung

Die logische Verknüpfung des XOR-Gatters mit zwei Eingangsvariablen kann mit einem „entweder – oder“ umschrieben werden. Die Ausgangsvariable wird immer dann „true“ liefern, wenn die Eingangsvariablen unterschiedliche Zustände haben. Die Wahrheitstabelle des XOR-Gatters entspricht dem ODER-Gatter mit dem Ausschluss gleicher Eingangszustände. Dieses Verhalten wird als **Antivalenz** bezeichnet. Es gibt ein nach IEC 60617-12 genormtes Schaltzeichen der Antivalenz-Funktion.

Wird in der Funktionsgleichung für die ODER Verknüpfung das v als Zeichen verwendet, kann die XOR Verknüpfung durch ein unterstrichenes v gekennzeichnet werden. Bei Verwendung des + Zeichens für ODER kennzeichnet ein Plus im Kreis ⊕ für die XOR-Verknüpfung.



Exklusiv ODER

$X = (A \wedge \bar{B}) \vee (\bar{A} \wedge B)$

$X = A \underline{\vee} B$

$X = A \oplus B$

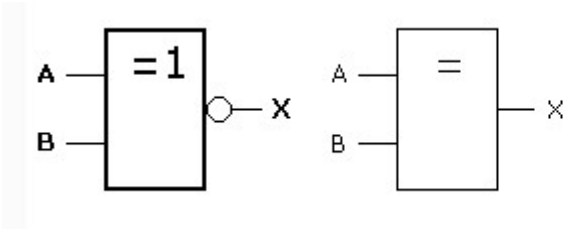
Funktionsgleichung

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Wahrheitstabelle

NXOR-Verknüpfung

Die Bezeichnung **Äquivalenz** bedeutet Gleichwertigkeit. Beim XNOR-Gatter mit zwei Eingangsvariablen ist der Zustand der Ausgangsvariable 1, wenn beide Eingangsvariablen den gleichen Zustand 0 oder 1 haben. Die Funktion kann auch durch eine Schaltung mit vier NOR-Gattern erreicht werden. Es sind zwei unterschiedliche Schaltzeichen zu finden. Nach IEC 60617-12 gilt das genormte Schaltzeichen des XOR-Gatters mit negiertem Ausgang.



XNOR Äquivalenz

$$X = (A \wedge B) \vee (\overline{A} \wedge \overline{B})$$

$$X = \overline{A \veebar B}$$

$$X = \overline{A \oplus B}$$

Funktionsgleichung

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

Funktionstabelle

Logikmodule

Logikmodule sind programmierbare Bausteine mit denen wir die Logikgatter umsetzen und testen können. Logikmodule sind häufig in der REG-Bauform von unterschiedlichen Herstellern zu finden. Für die praktischen Übungen arbeiten wir mit dem Logikmodul LOGO! von Siemens und easy von EATON.

LOGO! von Siemens

easy von EATON

Beide Logikmodule gibt es in den unterschiedlichsten Ausführungen und Ausbaustufen. Die Logikmodule sind sehr einfach über die Tasten am Gerät programmierbar, für beide Module steht auch eine Software für die Programmierung zur Verfügung. Durch diese Kombination aus Soft- und Hardware sind die Logikmodule die Vorstufe der Speicherprogrammierbaren Steuerungen (SPS), die sehr häufig in der technischen Automatisierung zu finden sind.

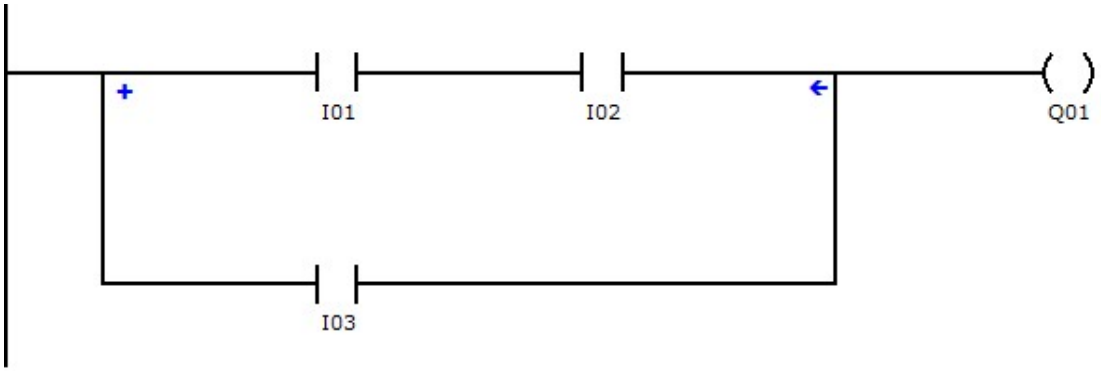
Die Programmierung der Logikmodule mit der entsprechenden Software ist mit drei unterschiedlichen Methoden möglich:

- Kontaktplan
- Funktionsplan und
- Strukturierter Text (AWL)

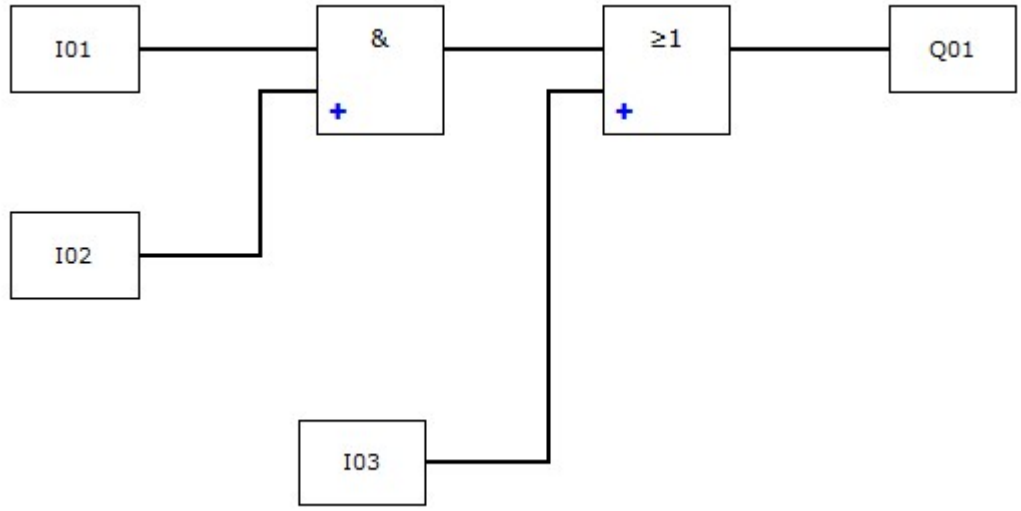
Der strukturierte Text (engl. Structured Text, Abkürzung: ST; auch engl. Structured Control Language, Abkürzung SCL) ist eine Programmiersprache für Speicherprogrammierbare Steuerungen (SPS). Die Norm EN 61131-3 legt neben anderen auch den Sprachumfang von ST fest. Dabei ist die Syntax der Sprachelemente ähnlich denen der Hochsprache Pascal und es wird wie bei allen Sprachen der EN 61131-3 bei Schlüsselwörtern keine Unterscheidung zwischen Groß- und Kleinschreibung gemacht (Case Insensitive).

ST bietet mehr Strukturierungsmöglichkeiten als die AWL und hat diese daher abgelöst. Komplexe Algorithmen und mathematische Funktionen lassen sich in ST übersichtlicher und schneller programmieren.

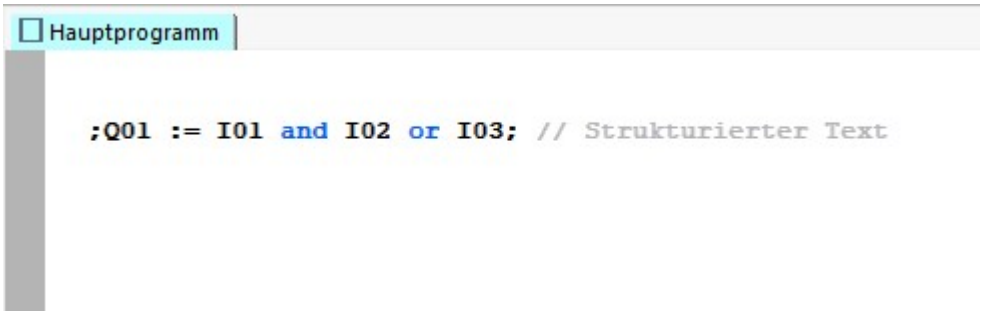
Programmiermethoden:



Logik-Darstellung im Kontaktplan (KOP)



Logik-Darstellung im Funktionsplan (FUP)



Logik-Darstellung mit strukturiertem Text (ST)

Übung:

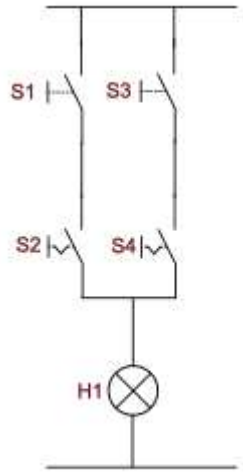
Erstelle die drei Logik-Grundfunktionen UND, ODER und NICHT in einem Logikmodul und teste sie (manuelle Eingabe).

Programmiere die drei Logik-Grundfunktionen UND, ODER und NICHT in der Software LOGO! Soft Comfort oder easySOFT7 und simuliere die Funktionen.

Als Hilfe für die Übungen stehen die entsprechenden Handbücher und Bedienungsanleitungen zur Verfügung.

Übung

Erstelle in der LOGO!Soft den Funktionsplan anhand des nachstehenden Stromlaufplans und simuliere deinen FUP.



Viel Erfolg!
