

Übung 2

1. Assembler-Programm

Gegeben ist ein Assembler-Programm.

Für die Übung stehen Ihnen zwei Hilfsblätter zur Verfügung, welche den Hauptspeicher und die CPU schematisch darstellen.

```

1 READ S20
2 READ S21
3 READ S22
4 LOAD S20, R1
5 LOAD S21, R2
6 LOAD S22, R3
7 MULT R1, R2, R4
8 MULT R4, R3, R5
9 ADD R5, R3, R6
10 STORE R6, S33
11 WRITE S33
12 STOP
    
```

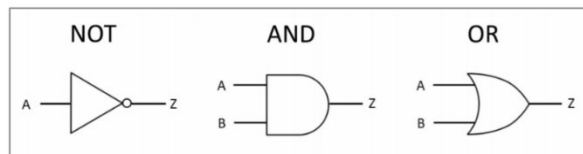
- Für das Programm steht der folgende **Adressbereich** im Hauptspeicher zur Verfügung: **S1-S43**
- Das Programm nimmt 3 **ganze Zahlen** als Input.
- Zuerst wird die Zahl **7** über die Tastatur eingegeben.
- Danach wird die Zahl **3** eingegeben.
- Abschließend wird noch die Zahl **5** eingegeben.

Welches Ergebnis wird berechnet bzw. ausgegeben?

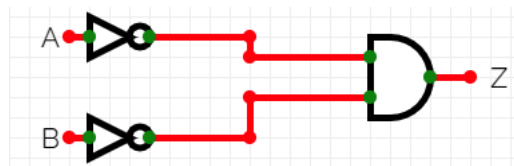
2. Logik

Eine wichtige Anwendung der Booleschen Algebra ist der Entwurf von logischen Schaltungen.

Gegeben sind drei Kombinationen (a., b. und c.) von Logikgattern mit den binären Inputs A, B und C und dem binären Output Z. Es werden dabei die folgenden Gattersymbole für die Operationen NOT, AND und OR verwendet:

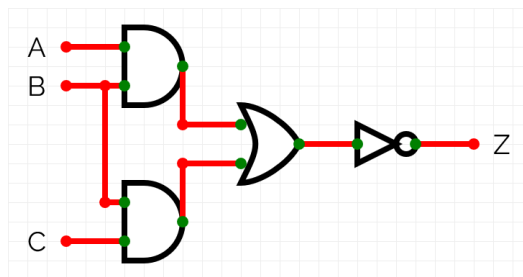


- a. Vervollständigen Sie die **Wahrheitstabelle** (mit den Wahrheitswerten 0 und 1) für die folgende **logische Schaltung**:



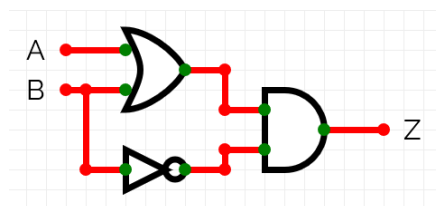
Wahrheitstabelle		
A	B	Z

- b. Geben Sie die **boolesche Funktion** an, die durch die folgende logische Schaltung realisiert wird. Verwenden Sie ausschließlich die folgenden Symbole: \neg Negation, \wedge Konjunktion, \vee Disjunktion, ((Klammer auf),) (Klammer zu)



Welchen Wert hat Z, wenn A und B den Wert 1 haben?
Welchen Wert hat Z, wenn A und B den Wert 0 haben?

- c. Geben Sie die **boolesche Funktion** an, die durch die folgende logische Schaltung realisiert wird. Verwenden Sie ausschließlich die folgenden Symbole: \neg Negation, \wedge Konjunktion, \vee Disjunktion, ((Klammer auf),) (Klammer zu)



3. Zahlensysteme

Sie sehen hier 7 Glühbirnen, mit denen insgesamt $2^7 = 128$ verschiedene (Binär-)Zahlen dargestellt werden können.

Sind alle Glühbirnen **ausgeschaltet (Status 0)**, dann ergibt das die Dezimalzahl **0**.

STATUS	0	0	0	0	0	0	0
Wert	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

Dezimalzahl: 0 (= 0 + 0 + 0 + 0 + 0 + 0 + 0)

Darstellung als 7-Bit-Binärzahl: 0000000








Sind alle Glühbirnen **eingeschaltet (Status 1)**, dann ergibt das die Dezimalzahl **127**.

STATUS	1	1	1	1	1	1	1
Wert	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$








Dezimalzahl: 127 (= 1 + 2 + 4 + 8 + 16 + 32 + 64)

Darstellung als 7-Bit-Binärzahl: 1111111

- a. Welche Glühbirnen müssen eingeschaltet werden, um die Dezimalzahl **23** als Binärzahl darzustellen? Tragen Sie die notwendigen Statusänderungen in der Tabelle ein!

							
STATUS							
Wert	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

- b. Welche Glühbirnen müssen eingeschaltet werden, um die Dezimalzahl **124** als Binärzahl darzustellen? Tragen Sie die notwendigen Statusänderungen in der Tabelle ein!

							
STATUS							
Wert	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

- c. Wandeln Sie die **Dezimalzahl 2022** in eine **Hexadezimalzahl** um.
Beschreiben Sie Ihren Rechenweg!
- d. Bilden Sie „händisch“ das **Produkt** der Binärzahlen 1000_2 und 1010_2 .
Geben Sie die Rechenschritte und Zwischenergebnisse an!
- e. Welcher **logischer Verknüpfung** entspricht die **Multiplikation** zweier **einstelliger** Binärzahlen?