

Bitverknüpfungen

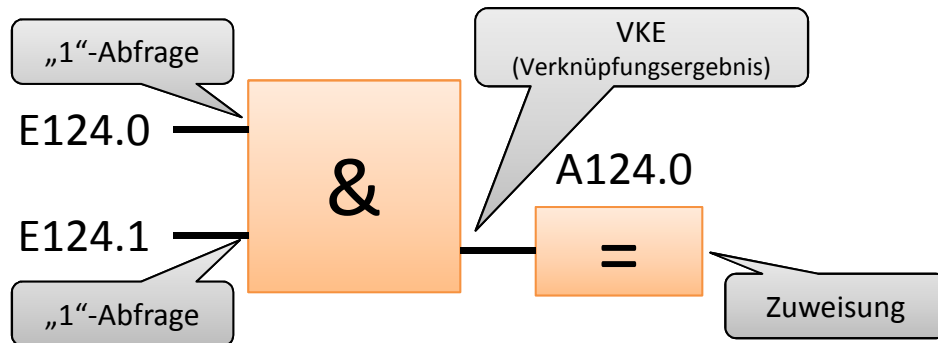
Mit Übungen
und
zeitlichen Signalverläufen

Von Michael Hartinger
Dipl.-Ing. (FH)

Bitverknüpfungen

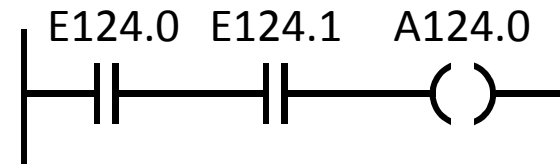
Das logische **UND** (AND)

Programm in FUP (Funktionsplan):



Das VKE beim logischen UND ist „1“, wenn alle Eingangsabfragen erfüllt sind.

Programm in KOP (Kontaktplan):



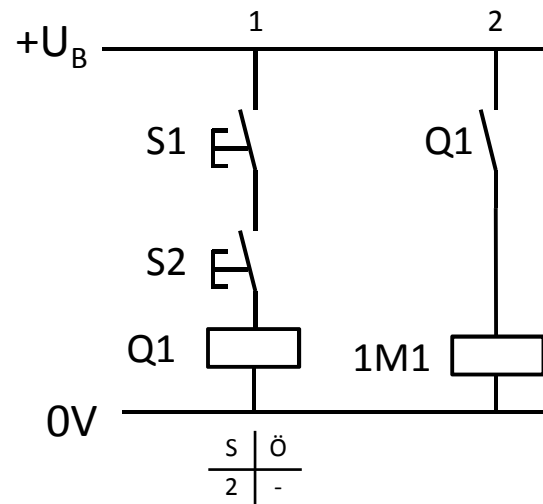
Programm in AWL (Anweisungsliste):

```
U  E124.0
U  E124.1
=  A124.0
```

Wahrheitstabelle:

E124.0	E124.1	A124.0
0	0	0
0	1	0
1	0	0
1	1	1

Elektroschaltplan:



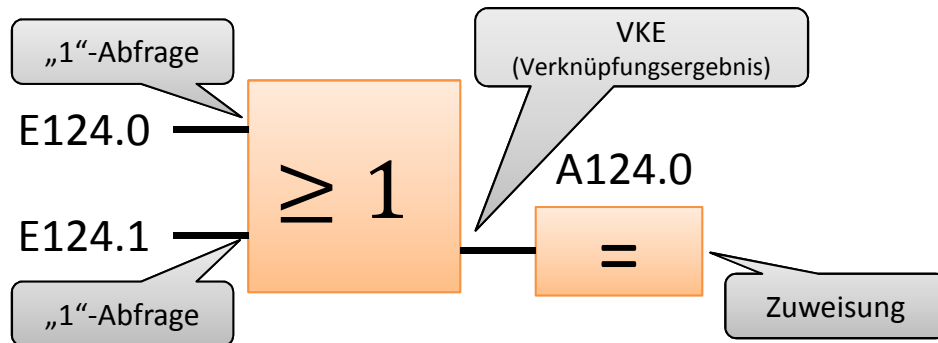
Boolesche Algebra:

$$S1 \wedge S2 = 1M1$$

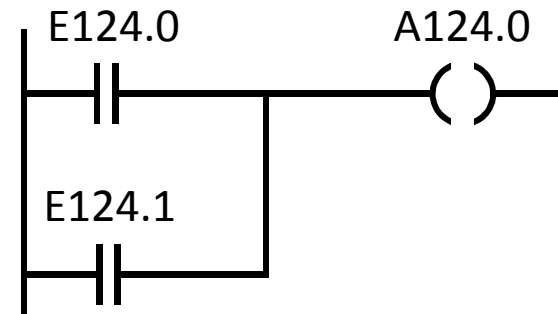
Bitverknüpfungen

Das logische **ODER** (OR)

Programm in FUP (Funktionsplan):



Programm in KOP (Kontaktplan):



Das VKE beim logischen ODER ist „1“, wenn mindestens eine Eingangsabfrage erfüllt ist.

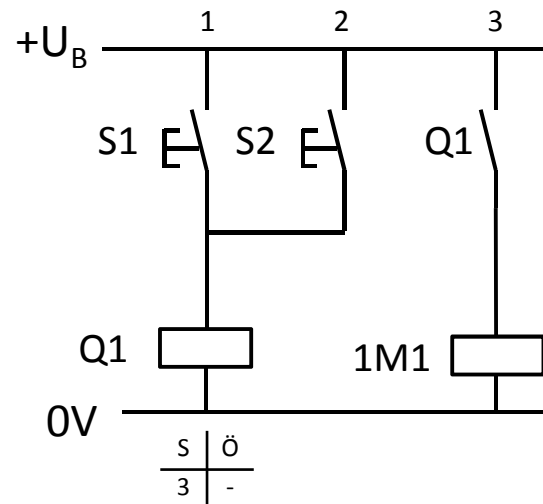
Programm in AWL (Anweisungsliste):

```
O E124.0
O E124.1
= A124.0
```

Wahrheitstabelle:

E124.0	E124.1	A124.0
0	0	0
0	1	1
1	0	1
1	1	1

Elektroschaltplan:



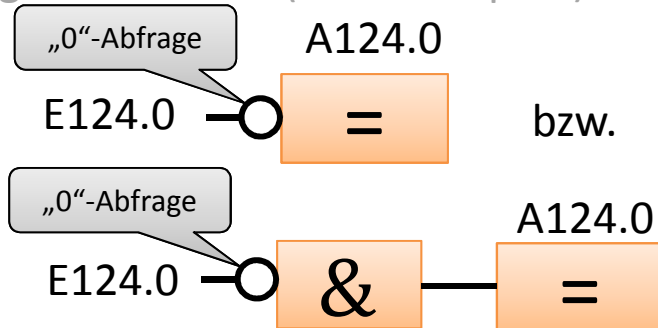
Boolesche Algebra:

$$S1 \vee S2 = 1M1$$

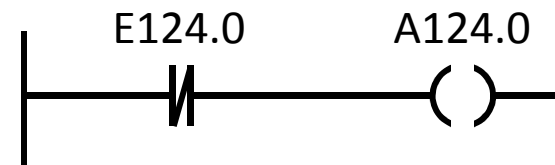
Bitverknüpfungen

Das logische NICHT (NOT)

Programm in FUP (Funktionsplan):



Programm in KOP (Kontaktplan):



Programm in AWL (Anweisungsliste):

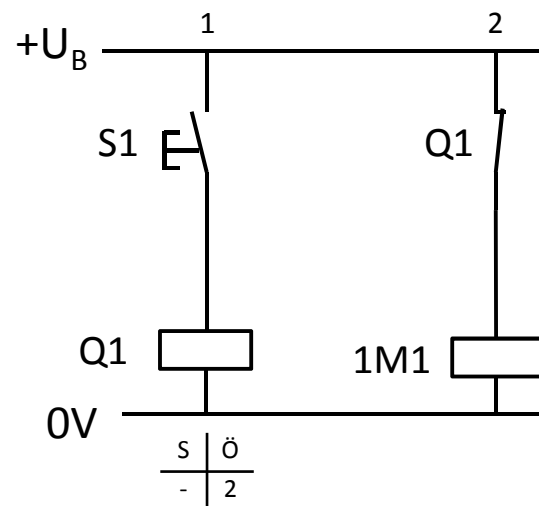
UN E124.0
= A124.0

Das VKE beim logischen NICHT ist „1“, wenn die Nullabfrage erfüllt ist.

Wahrheitstabelle:

E124.0	A124.0
0	1
1	0

Elektroschaltplan:



Boolesche Algebra:

$$\overline{S1} = 1M1$$

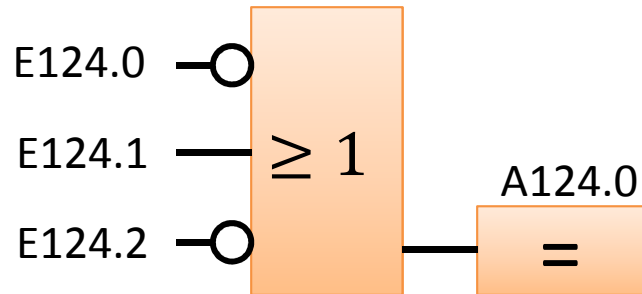
bzw.

$$S1 = \overline{1M1}$$

Bitverknüpfungen

Übungen

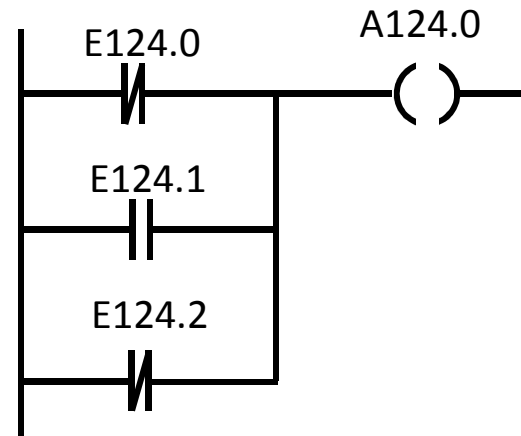
Gegeben: FUP



Gesucht: Wahrheitstabelle

E124.0	E124.1	E124.2	A124.0
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Gesucht: KOP



Gesucht: AWL

```

ON  E124.0
O   E124.1
ON  E124.2
=   A124.0
  
```

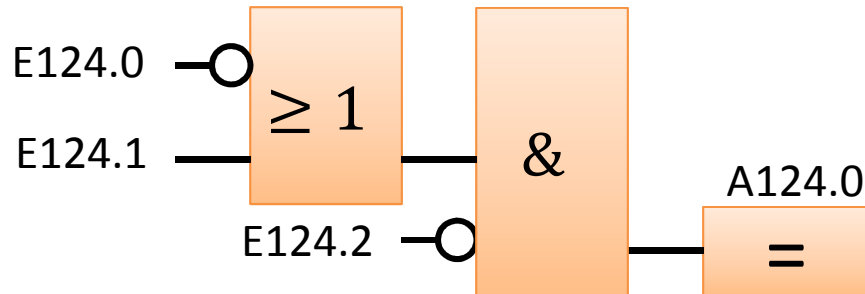
Gesucht: Boolesche Algebra:

$$\overline{E124.0} \vee E124.1 \vee \overline{E124.2} = A124.0$$

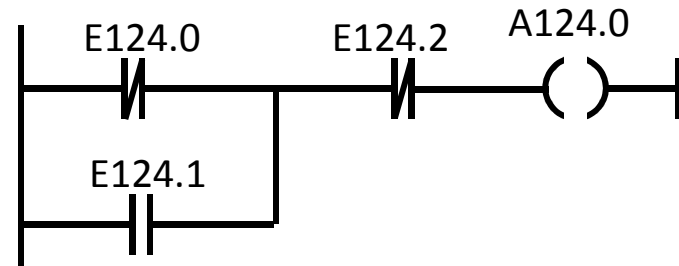
Bitverknüpfungen

Übungen

Gegeben: FUP



Gesucht: KOP



Gesucht: Wahrheitstabelle

E124.0	E124.1	E124.2	A124.0
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Gesucht: AWL

```

U(
ON  E124.0
O   E124.1
)
UN  E124.2
=   A124.0
    
```

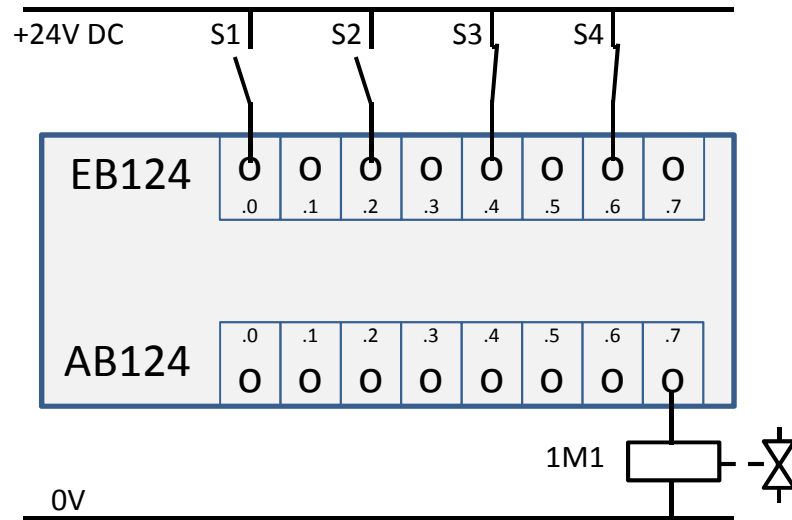
Gesucht: Boolesche Algebra:

$$(\overline{E124.0} \vee E124.1) \wedge \overline{E124.2} = A124.0$$

Bitverknüpfungen

Übungen

Gegeben: Schaltplan



Hilfe: Wahrheitstabelle

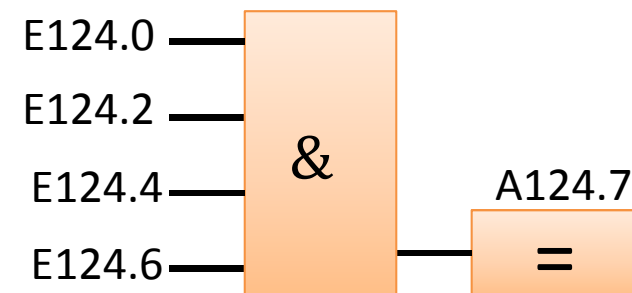
E124.0	E124.2	E124.4	E124.6	A124.7
1	1	1	1	1

Alle weiteren fünfzehn Zeilen liefern „0“!

Lösung: SPS-Programm in FUP

Gegeben: Funktionsbeschreibung

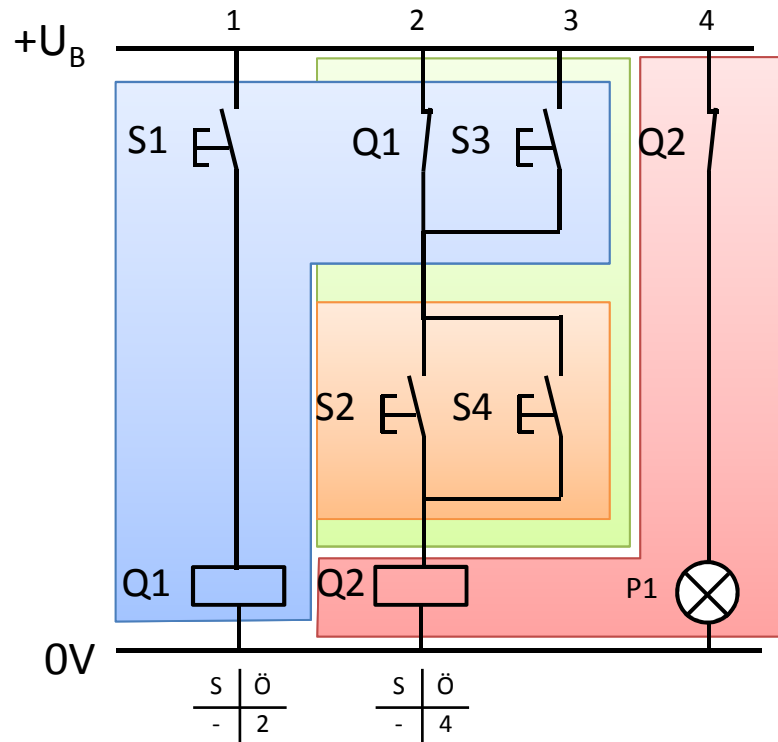
Das Ventil 1M1 soll angesteuert werden, wenn S1 und S2 betätigt sind und S3 und S4 dabei nicht betätigt sind!



Bitverknüpfungen

Übungen

Gegeben: Konventioneller Schaltplan

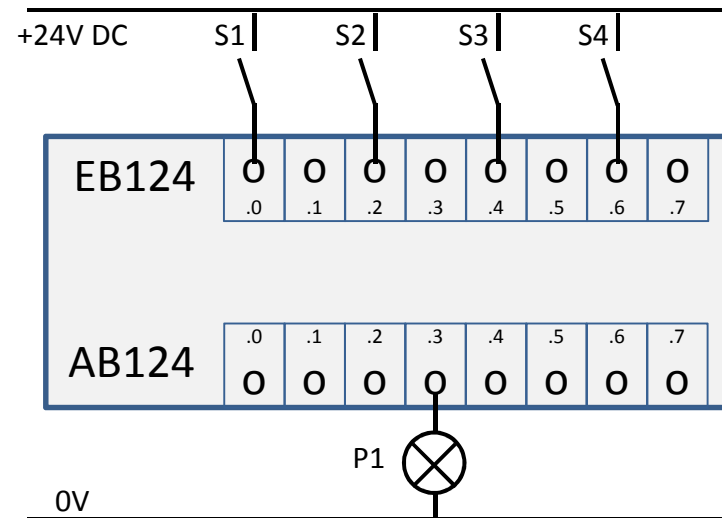


Gegeben: Aufgabe

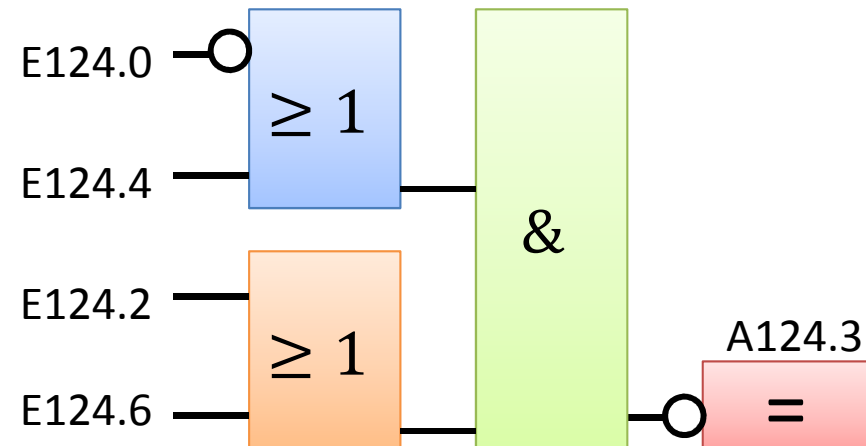
Die konventionelle festverdrahtete Schaltung wird durch eine SPS ersetzt. Die Taster S1 bis S4 werden dort an die Eingänge und die Lampe P1 an einen Ausgang angeschlossen.

Ein entsprechendes SPS-Programm soll die ursprüngliche Funktion wieder herstellen!

Gegeben: Neuer Schaltplan mit SPS

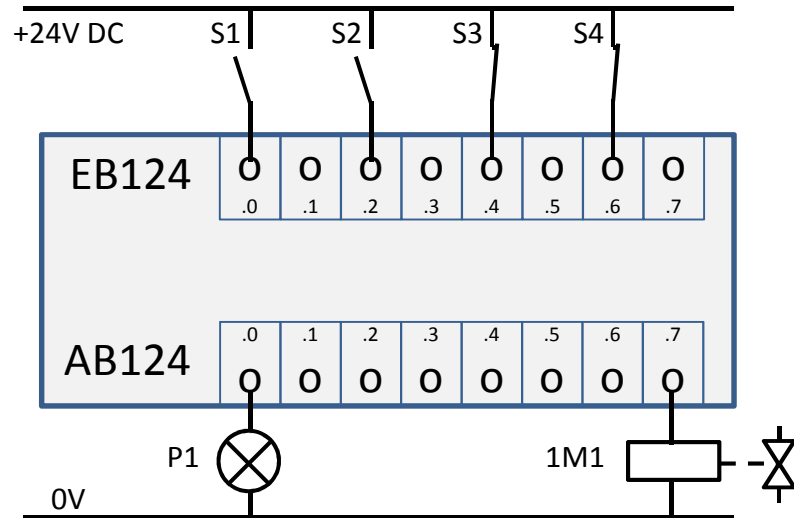


Gesucht: SPS-Programm in FUP



Bitverknüpfungen

Gegeben: Schaltplan



Hilfe: Wahrheitstabelle

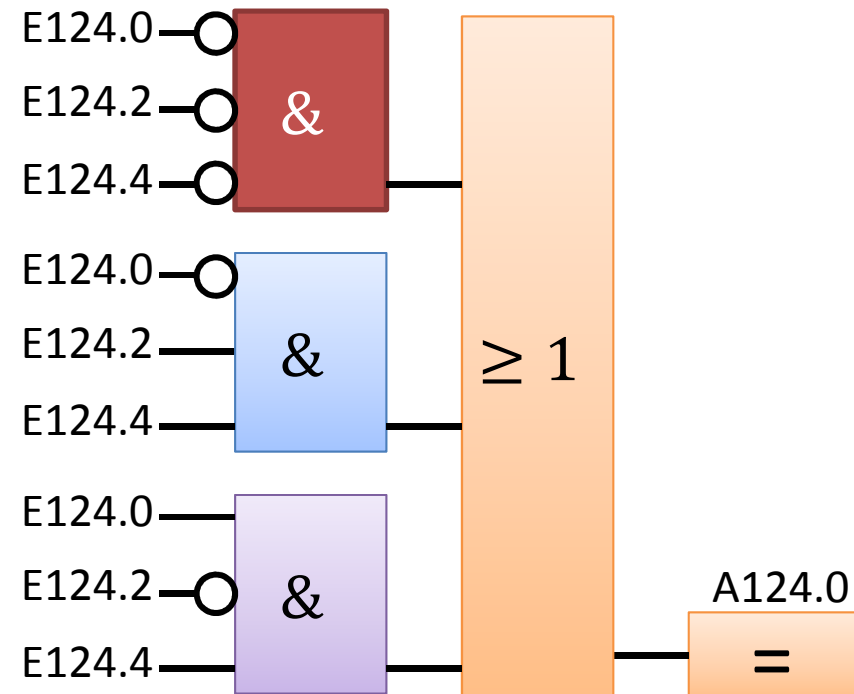
E124.0	E124.2	E124.4	A124.0
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Übungen

Gegeben: Funktionsbeschreibung

Die Lampe P1 soll leuchten, wenn genau einer der drei Taster S1, S2 bzw. S3 betätigt ist!

Lösung: SPS-Program in FUP



Übungen mit Signalverläufen

```

graph LR
    E124.0 --- AND[&]
    E124.1 --- AND
    AND --- A124.0[=]
    style A124.0 fill:#fde9d9,stroke:#333,stroke-width:1px

```

