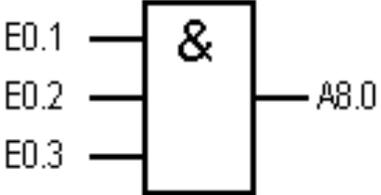
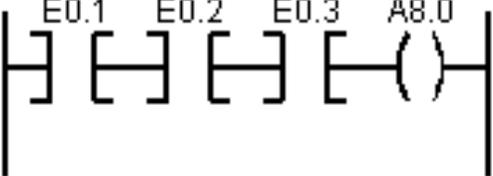


# SPS S7

# Grundverknüpfungen

© U. Ohm, BBS 4, Hannover

# UND-Verknüpfungen

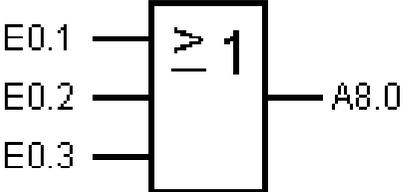
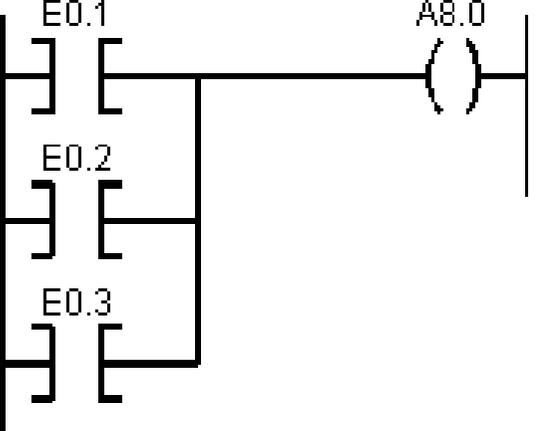
Programmdarstellung		
FUP	KOP	AWL
		<pre> U E0.1 U E0.2 U E0.3 = A8.0                     </pre>

## Funktionsbeschreibung

Die UND-Verknüpfung entspricht der Reihenschaltung von Kontakten(Vorlage). Am Ausgang

A 8.0 erscheint Signalzustand "1", wenn alle Eingänge gleichzeitig den Signalzustand "1" aufweisen. Wenn mindestens einer der Eingänge den Signalzustand "0" aufweist erscheint am Ausgang Signalzustand "0". Die Anzahl der Abfragen und die Reihenfolge der Programmierung ist beliebig.

# ODER-Verknüpfung

Programmdarstellung		
FUP	KOP	AWL
		<pre>O E0.1 O E0.2 O E0.3 = A8.0</pre>

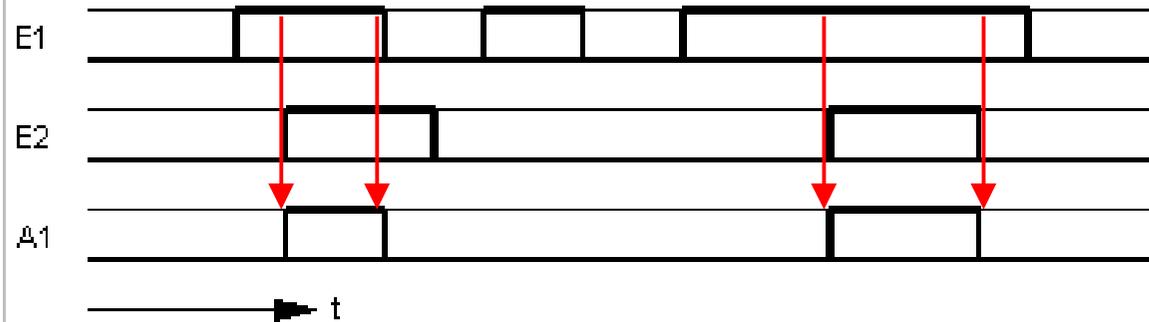
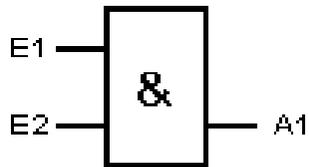
## Funktionsbeschreibung

Die ODER-Verknüpfung entspricht der Parallelschaltung einzelner Kontakte(Vorlage). Am Ausgang A 8.0 erscheint Signalzustand "1" wenn mindestens einer der Eingänge den Signalzustand "1" aufweist. Am Ausgang erscheint Signalzustand "0", wenn alle Eingänge gleichzeitig den Signalzustand "0" aufweisen.

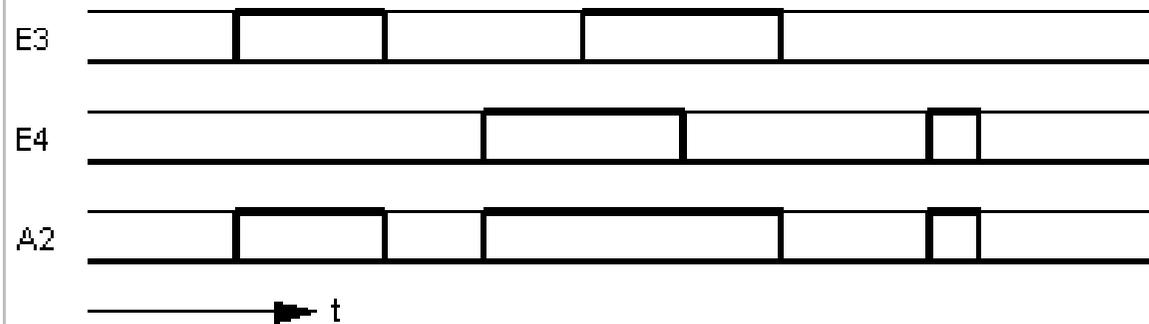
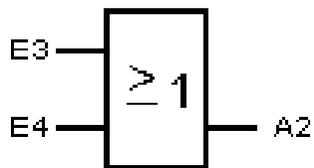
Die Anzahl der Abfragen und die Reihenfolge der Programmierung ist beliebig.

# Zeitablauf-Diagramm

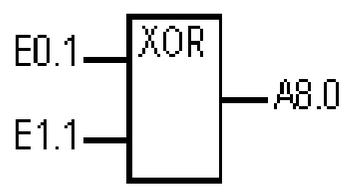
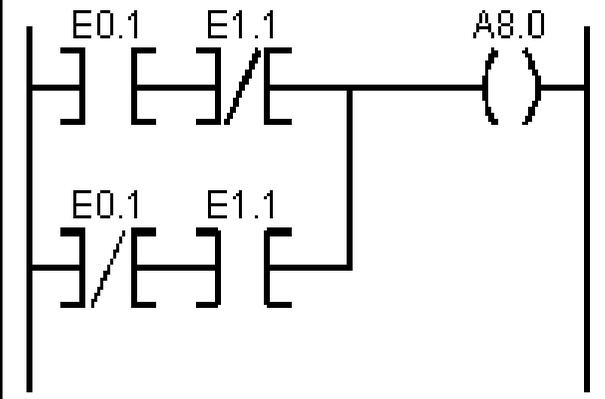
## UND-Funktion



## ODER-Funktion



# XOR-Verknüpfung

Programmdarstellung		
FUP	KOP	AWL
		<pre> X E 0.1 X E 1.1 = A 8.0                     </pre>

## Funktionsbeschreibung

Die XOR-Verknüpfung liefert am Ausgang A 8.0 Signalzustand "1", wenn nur einer der Eingänge den Signalzustand "1" aufweist. Am Ausgang A8.0 erscheint Signalzustand "0", wenn alle Eingänge gleichzeitig den Signalzustand "0" oder "1" aufweisen. Bei XOR-Verknüpfungen können nur 2 Eingänge angelegt werden.

## XOR-Verknüpfung von UND-Funktionen

Programmdarstellung:	
FUP	AWL
<pre> graph LR     E0_0[E0.0] --- AND1[&amp;]     E0_1[E0.1] --- AND1     E1_0[E1.0] --- AND2[&amp;]     E1_1[E1.1] --- AND2     AND1 --- XOR[XOR]     AND2 --- XOR     XOR --- A8_0[A8.0]         </pre>	<pre> UE 0.0 UE 0.1 X ( UE 1.0 UE 1.1 ) = A 8.0         </pre>

### Funktionsbeschreibung

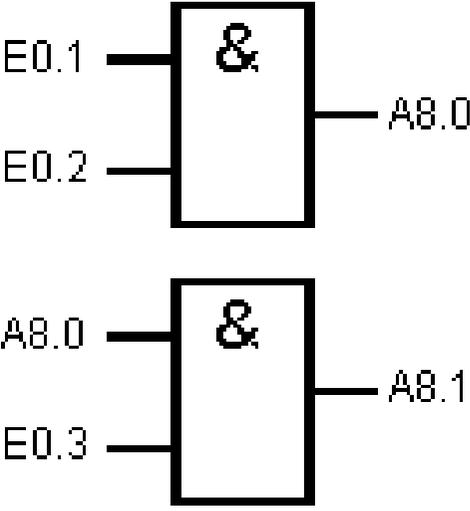
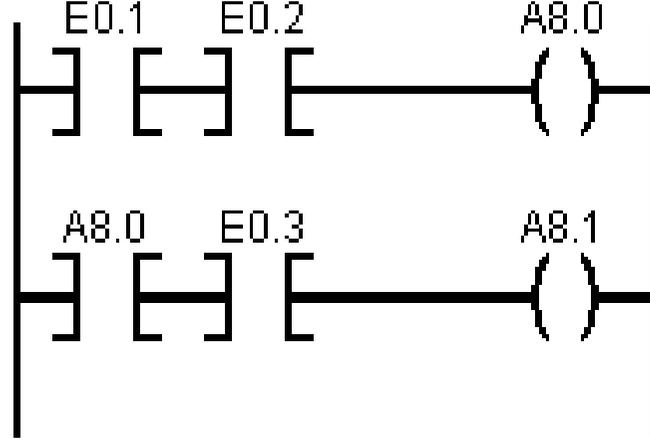
Die Verknüpfungsergebnisse der beiden UND-Verknüpfung werden durch die Klammerfunktion Exklusiv- ODER verknüpft.

In der ersten UND- Verknüpfung wurde keine Klammer gesetzt, da die Exklusiv ODER-Funktion eine höhere Priorität hat.

# Negation von Klammerausdrücken

Programmdarstellung:		
FUP	AWL mit negierter Klammer	AWL mit der Operation NOT
<p>The diagram shows a ladder logic network. It starts with two parallel branches. The upper branch contains an OR gate with inputs E0.0 and E0.1. The lower branch contains an XOR gate with inputs E1.0 and E1.1. The outputs of these two gates are connected to the inputs of an AND gate. The output of the AND gate is labeled A8.0.</p>	<pre> UN (   O E 0.0   O E 0.1 ) UN(   X E 1.0   X E 1.1 ) = A 8.0 </pre>	<pre> U (   O E 0.0   O E 0.1 ) NOT U(   X E 1.0   X E 1.1 ) NOT ) = A 8.0 </pre>

# Abfragen von Ausgängen

Programmdarstellung		
FUP	KOP	AWL
		<pre> Netzwerk 1 U  E0.1 U  E0.2 =  A8.0  Netzwerk 2 U  A8.0 U  E0.3 =  A8.1                     </pre>

## Funktionsbeschreibung

Für das Einschalten der Ausgänge A 8.0 und A 8.1 gelten unterschiedliche Bedingungen. In diesen Fällen muss für jeden Ausgang ein eigenes Verknüpfungssymbol vorgesehen werden. Da das Automatisierungsgerät nicht nur den Signalzustand von Eingängen, sondern auch den von Ausgängen, Merkern usw. abfragen kann, wird in der UND-Verknüpfung für den Ausgang A 8.1 der Ausgang A8.0 abgefragt.

# Merker / UND-vor-ODER

## Funktionsbeschreibung

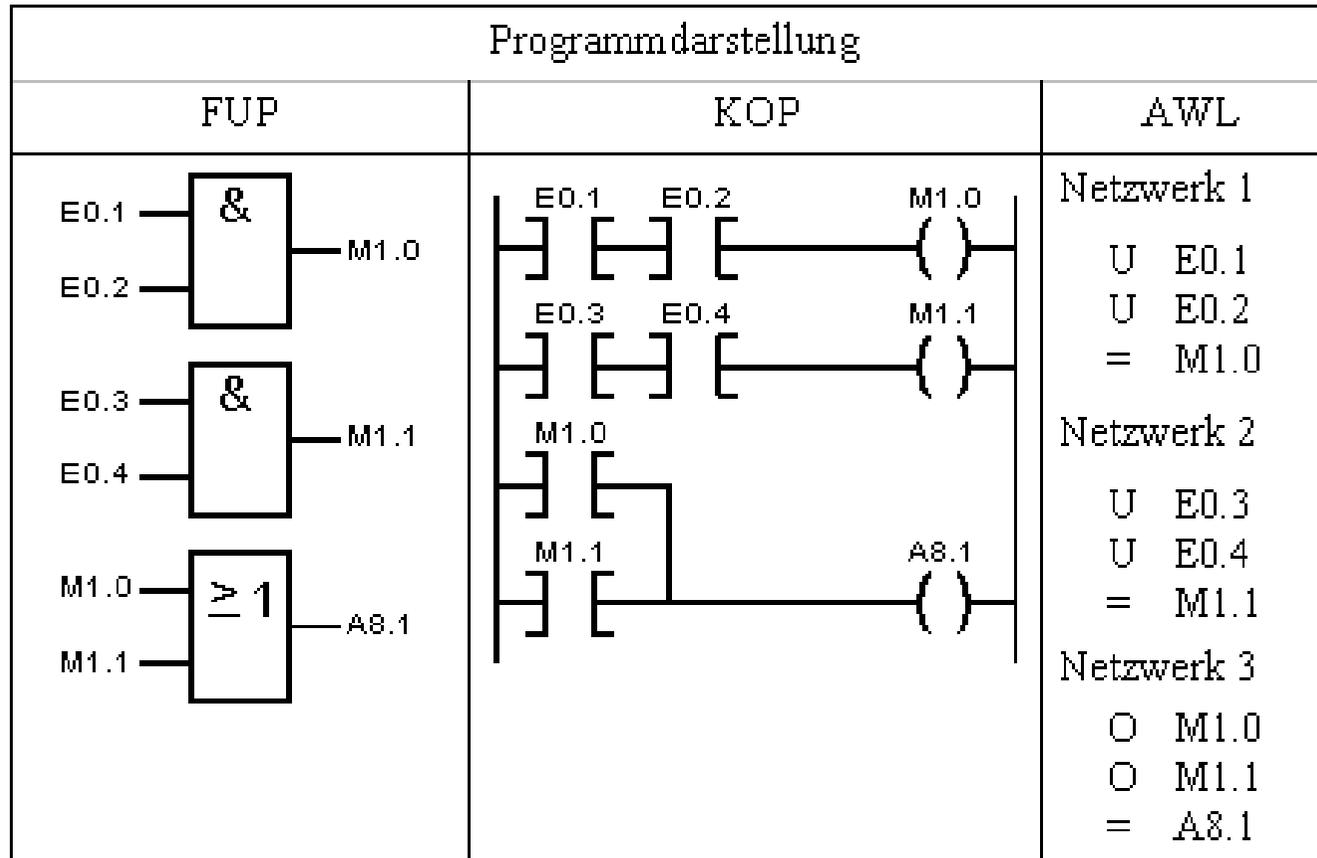
Bei der UND-vor-ODER-Verknüpfung können, je nach Automatisierungsgerät, unterschiedliche Programmierungen vorgenommen werden.

Die einfachste, an allen Automatisierungsgeräten anwendbare, aber an Speicherplätzen und Bearbeitungszeit aufwendigere Programmierung ist über Merker möglich. Dabei wird jede UND-Verknüpfung über einen Merker abgeschlossen. Anschließend werden die Merker nach ODER verknüpft.

Versteht ein Steuergerät die Operation 0 (= ODER-Verknüpfung von UND-Funktionen, wird ohne zusätzlichen Operanden programmiert) kann die UND-vor-ODER-Verknüpfung auch ohne Merker programmiert werden. Die Operation 0 wird immer dann verwendet, wenn nach einer ODER-Bedingung die nächste ODER-Bedingung eine UND-Funktion ist.

Siehe nächste Seite

# UND-vor-ODER / Beispiel mit Merker



# UND-vor-ODER / Beispiel ohne Merker

Programmdarstellung		
FUP	KOP	AWL
		<pre> U  E0.1 U  E0.2 O U  E0.3 U  E0.4 =  A8.1 </pre>

# ODER-vor-UND

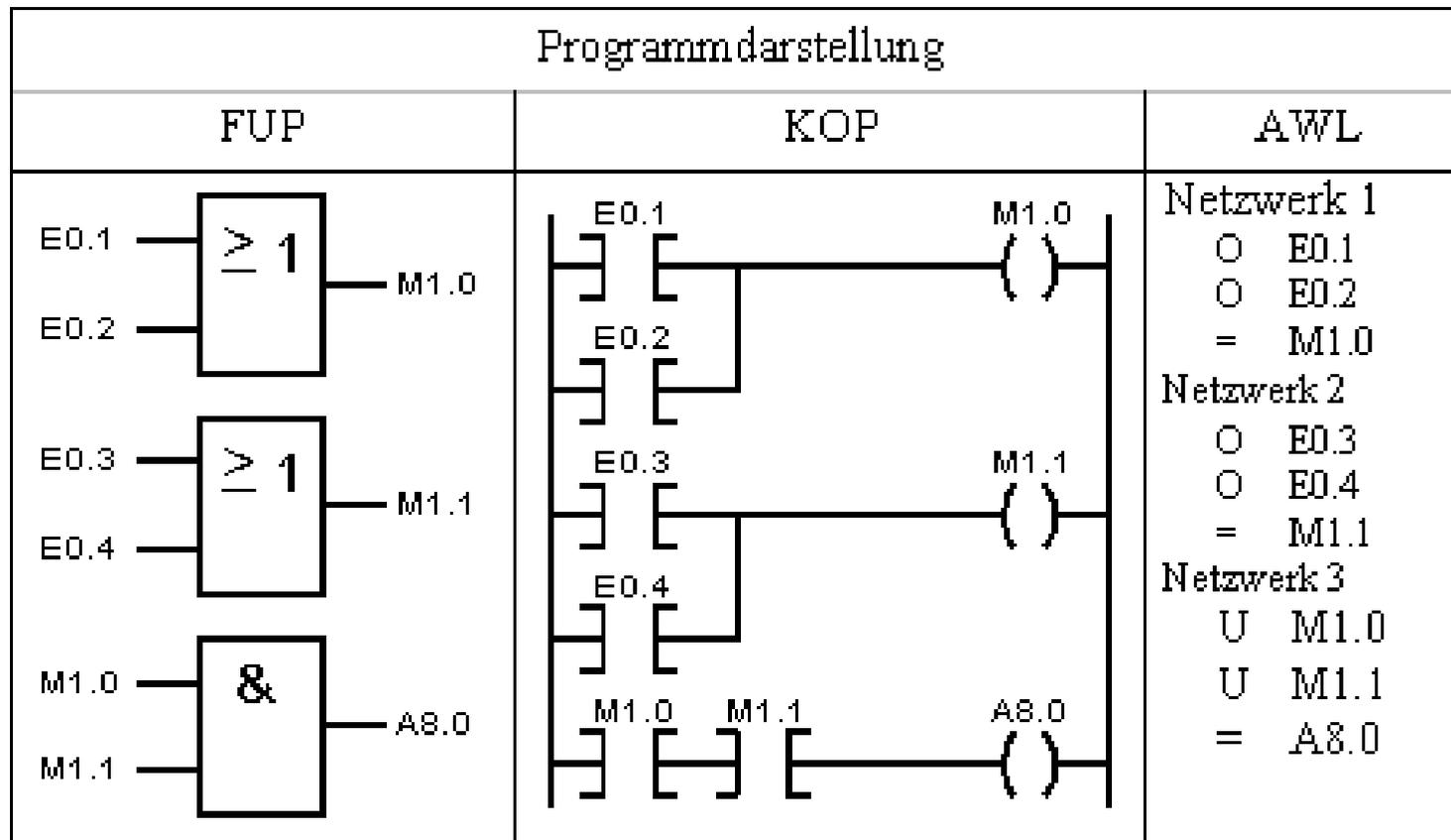
## Funktionsbeschreibung

Bei der ODER-vor-UND-Verknüpfung können je nach Automatisierungsgerät unterschiedliche Programmierungen vorgenommen werden. Die einfachste, an allen Automatisierungsgeräten anwendbare, aber an Speicherplätzen und Bearbeitungszeit aufwendigere Programmierung ist über Merker möglich. Dabei wird jede ODER-Verknüpfung mit einem Merker abgeschlossen. Anschließend werden die Merker nach UND verknüpft.

Versteht ein Steuergerät die Operation "U"(UND-Verknüpfung von Klammern ausgedrückt, wird ohne zusätzlichen Operanden programmiert), kann die ODER-vor-UND-Verknüpfung auch ohne Merker programmiert werden. Die Operation "U(" wird immer dann verwendet wenn ODER-Funktionen nach UND verknüpft werden. Mit der Operation "U(" ist festgelegt, dass die ODER-Funktionen vor den entsprechenden UND-Funktionen bearbeitet werden.

Siehe nächste Seite

# ODER-vor-UND mit Merker



# Negation von Klammerausdrücken

Programmdarstellung:		
FUP	AWL mit negierter Klammer	AWL mit der Operation NOT
	<pre> UN (   O E 0.0   O E 0.1 ) UN(   X E 1.0   X E 1.1 ) = A 8.0 </pre>	<pre> U (   O E 0.0   O E 0.1 ) NOT U(   U(     X E 1.0     X E 1.1   )   NOT ) = A 8.0 </pre>
<b>Funktionsbeschreibung</b>		

Neben Binäroperationen können Sie auch Klammerausdrücke negieren. Das bedeutet, dass die CPU das Ergebnis des Klammerausdrucks negiert weiterverarbeitet.

Eine zweite Möglichkeit des Negierens von Klammerausdrücken ist die Anweisung NOT.

Eine NOT-Operation vor der Klammer-zu-Anweisung negiert das Ergebnis des Klammerausdrucks vor der Weiterverknüpfung.

# Selbsthaltung

Für das Ausschalten des Ausgangs sind zwei Varianten möglich, je nachdem, ob das Einschalten oder Ausschalten vorrangig ist.

## Vorrangiges Ausschalten A

Mit Signal "1" am Eingang E 0.1 wird der Ausgang A 2.0 mit Selbsthaltung auf Signal "1" gelegt.

Mit Signal "0" am Eingang E 1.0 wird der Ausgang A 2.0 auf Signal "0" gelegt, auch dann wenn der Eingang E 1.1 Signal "1" führt.

## Vorrangiges Einschalten B

Mit Signal "1" am Eingang E 0.1 wird der Ausgang A 2.1 mit Selbsthaltung auf Signal "1" gelegt.

Mit Signal "0" am Eingang E 1.0 kann der Ausgang A 2.1 nur dann auf Signal "0" gelegt werden, wenn der Eingang E 1.1 Signal "0" führt.

An Stelle der Selbsthaltungeschaltung wird in der Praxis meist die RS-Funktion verwendet.

Siehe nächste Seite

# Selbsthaltung mit Merkern

Programmdarstellung		
FUP	KOP	AWL
		<pre> O   E 0.1 O   A 2.0 =   M1.0 *** U   M1.0 U   E 1.0 =   A 2.0 BE </pre>
		<pre> U   A 2.1 U   E 1.0 O   E 0.1 =   A 2.1 BE </pre>

**Auf Vorrang achten !!**

# RS -Speicherfunktion

## Funktionsbeschreibung

**Kurzzeitiger Signalzustand "1" am Setzeingang (S) setzt die Speicherfunktion, kurzzeitiger Signalzustand "1" am Rücksetzeingang führt zum Rücksetzen der Speicherfunktion. Signalzustand "0" an den Eingängen S und R verändert den vorher eingestellten Zustand nicht.**

**Wenn beide Eingänge R und S gleichzeitig mit Signal "1" belegt sind, kann vorrangig zurückgesetzt (Darstellung A) oder vorrangig gesetzt werden (Darstellung B).**

**Dieses vorrangige Rücksetzen oder Setzen muss bei der Programmierung berücksichtigt werden.**

**Die zuletzt programmierten Anweisungen werden vom Automatisierungsgerät mit Vorrang bearbeitet.**

**Im Beispiel A wird zunächst die Setzoperation ausgeführt; der Ausgang A 2.0 wird wieder zurückgesetzt und bleibt für den Rest der Programmbearbeitung zurückgesetzt.**

**Dieses kurzzeitige Setzen des Ausganges A 2.0 wird nur im Prozessabbild durchgeführt. Der Signalzustand auf der dazugehörigen Peripheriebaugruppe wird während der Programmbearbeitung nicht beeinflusst.**

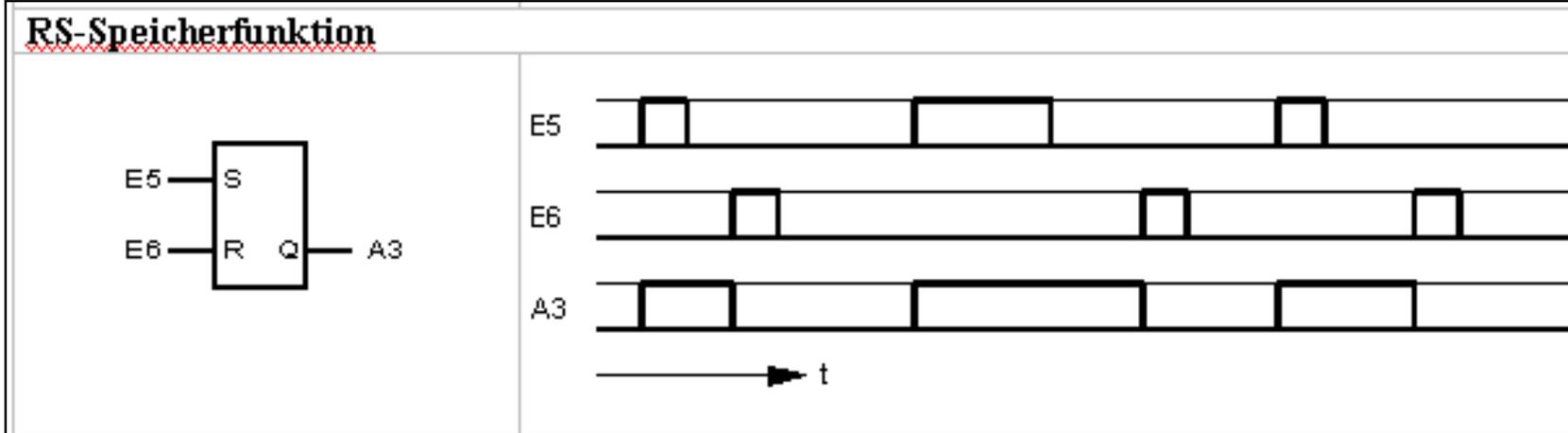
**Sinngemäß wird der Ausgang A 2.1 mit Vorrang gesetzt.**

**Bitte beachten: Da das Ausschalten durch Betätigung des Öffners am Eingang E 1.0 erfolgen soll, muss für den Eingang E 1.0 der Signalzustand "0" abgefragt werden.**

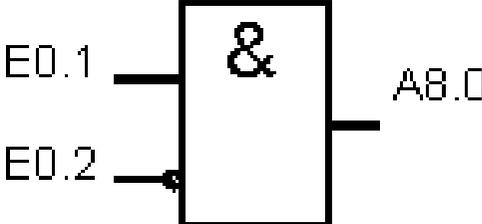
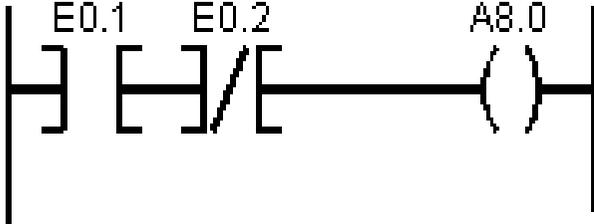
# RS -Speicherfunktion

Programmdarstellung		
FUP	KOP	AWL
<p>Diagram A: FUP representation of an RS flip-flop. Inputs E0.1 and E0.2 are connected to an OR gate (labeled <math>\geq 1</math>). The output of the OR gate is connected to the R input. Input E0.2 is also connected to the S input. The Q output is labeled A2.0.</p>	<p>Diagram A: KOP representation of an RS flip-flop. E0.2 is connected to the S coil. E0.1 is connected to the R coil. The output Q is connected to coil A.</p>	<pre> O E0.2 S A2.0 ON E1.0 O E0.1 R A2.0 NOP 0 BE                     </pre>
<p>Diagram B: FUP representation of an RS flip-flop. Inputs E0.1 and E0.2 are connected to an OR gate (labeled <math>\geq 1</math>). The output of the OR gate is connected to the R input. Input E0.2 is also connected to the S input. The Q output is labeled A2.1.</p>	<p>Diagram B: KOP representation of an RS flip-flop. E1.0 is connected to the R coil. E0.1 is connected to the R coil. E0.2 is connected to the S coil. The output Q is connected to coil B.</p>	<pre> ON E1.0 O E0.1 R A2.1 O E0.2 S A2.1 NOP 0 BE                     </pre>

# Zeitablauf-Diagramm



## Abfrage auf Signalzustand "0"

Programmdarstellung		
FUP	KOP	AWL
		<pre> U  E0.1 UN E0.2 =  A8.0                     </pre>

### Funktionsbeschreibung

Bei speicherprogrammierten Automatisierungsgeräten können nur Schalter mit einem Kontakt (Öffner oder Schließer) verwendet werden. Hier besteht die Möglichkeit einen Operanden auf den Signalzustand "1" aber auch auf den Signalzustand "0" abzufragen.

Nach DIN 57 113/VDE 0113, Pkt. 8.8 soll das Stillsetzen einer Maschine möglichst durch Abschalten der Spannung eingeleitet werden. Diese Art des Stillsetzens ist sicher, weil sie auch bei Erdschluss, Drahtbruch oder Spannungsausfall im Geberstromkreis funktioniert.

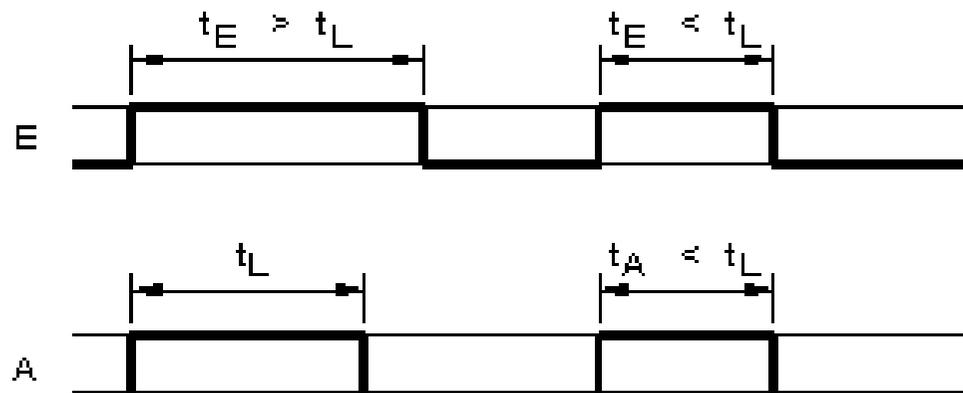
Die aus diesem Grund bei AUS-Tastern und Grenztastern üblichen Öffner sollten daher keinesfalls durch Schließer mit Abfrage auf den Signalzustand "0" ersetzt werden.

# Zeit als Impuls

## Funktionsbeschreibung

Mit Signal "1" am Eingang wird der Ladebefehl für den programmierten Zeitwert  $L$  S5T#10S, und der Startbefehl für die Impulszeit SI T1 ausgeführt. Der Ausgang führt Signal "1". Die Dauer des Impulses entspricht der programmierten Laufzeit. Der Impuls wird vor Ablauf der Laufzeit beendet, wenn das Eingangssignal "0" wird oder wenn die Operation Rücksetzen mit Signal "1" ansteht.

Die digitalen Abfragen an den Ausgängen DUAL (=Dualzahl) bzw. DEZ (Dezimalzahl) liefern den aktuellen Zeitwert können mit LADE- und TRANSFEROPERATIONEN weiter verarbeitet werden.



Siehe nächste Seite

# Zeit als Impuls

Programmdarstellung		
FUP	KOP	AWL
		<pre> U  E0.1 L S5T#10S SI  T1 NOP 0 NOP 0 NOP 0 U  T1 =  A2.0 </pre>

# Zeit als verlängerter Impuls

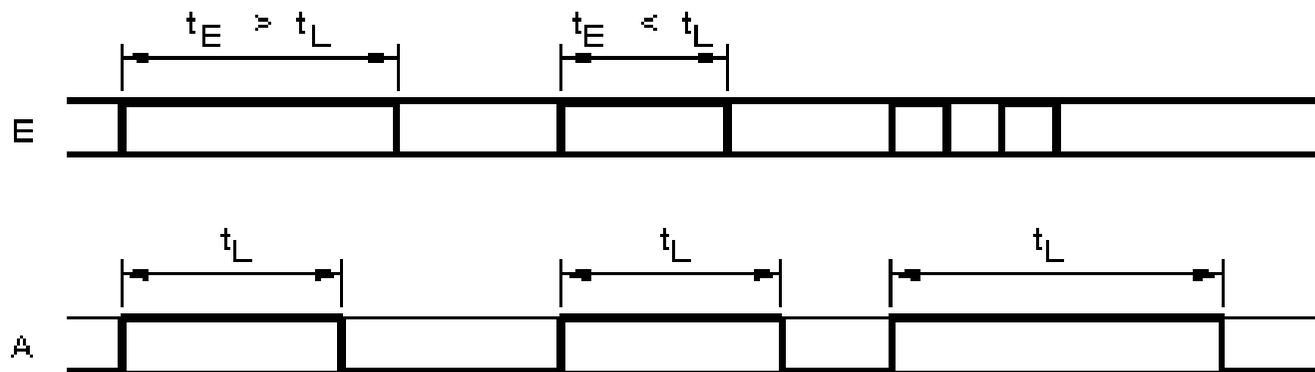
## Funktionsbeschreibung

Mit Signal "1" am Eingang wird der Ladebefehl für den programmierten Zeitwert L S5T#10S und der Startbefehl für die verlängerte Impulszeit SV T2 ausgeführt. Der Ausgang führt Signal "1".

Die Dauer des Impulses entspricht der programmierten Laufzeit.

Der Impuls wird vor Ablauf der Laufzeit beendet, wenn die Operation Rücksetzen mit Signal "1" ansteht.

Die digitalen Abfragen an den Ausgängen DUAL (= Dualzahl) bzw. DEZ (= Dezimalzahl) liefern den aktuellen Zeitwert (Istwert). Diese Zeitwerte können mit LADE- und TRANSFEROPERATIONEN weiter verarbeitet werden.



Siehe nächste Seite

# Zeit als verlängerter Impuls

Programmdarstellung		
FUP	KOP	AWL
<div style="text-align: center;">T2</div>	<div style="text-align: center;">T2</div>	<pre> U  E0.1 L S5T#10S SV T2 NOP 0 NOP 0 NOP 0 U  T2 =  A2.0                     </pre>

# Einschaltverzögerung

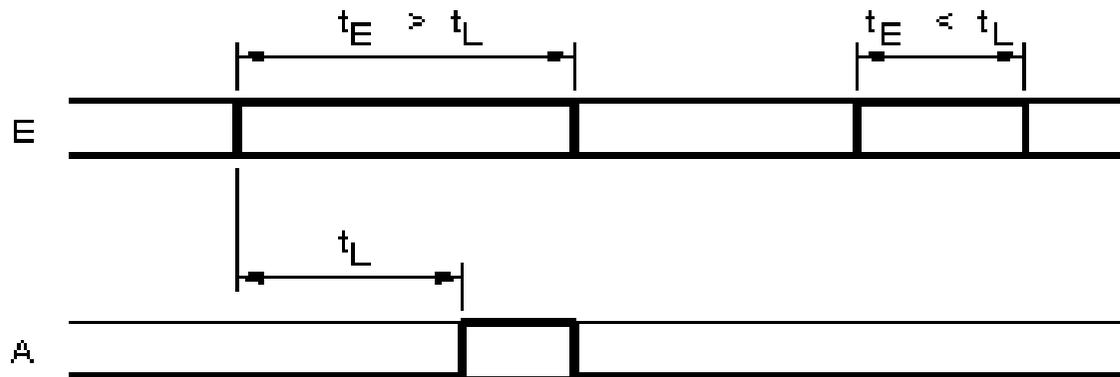
## Funktionsbeschreibung

Mit Signal "1" am Eingang wird der Ladebefehl für den programmierten Zeitwert  $L$   $S5T\#10S$  und der Startbefehl für die Verzögerungszeit  $SE T3$  ausgeführt. Der Ausgang führt erst dann Signal "1", wenn die eingestellte Laufzeit abgelaufen ist und der Eingang noch mit Signal "1" beschaltet ist.

Der Ausgang wird ausgeschaltet (Signal "0"), wenn das Eingangssignal "0" wird. Ist das Eingangssignal kürzer als die eingestellte Verzögerungszeit, bleibt der Ausgang ausgeschaltet (Signal "0").

Die digitalen Abfragen an den Ausgängen DUAL (= Dualzahl) bzw. DEZ (= Dezimalzahl) liefern den aktuellen Zeitwert (Istwert). Diese Zeitwerte können mit LADE- und TRANSFEROPERATIONEN weiter verarbeitet werden.

## Funktionsdiagramm



Siehe nächste Seite

# Einschaltverzögerung

## Programmdarstellung

FUP	KOP	AWL
		<pre> U   EO.1 L SST#10S SE  T3 NOP 0 NOP 0 NOP 0 U   T3 =   A2.0 </pre>

# Speichernde Einschaltverzögerung

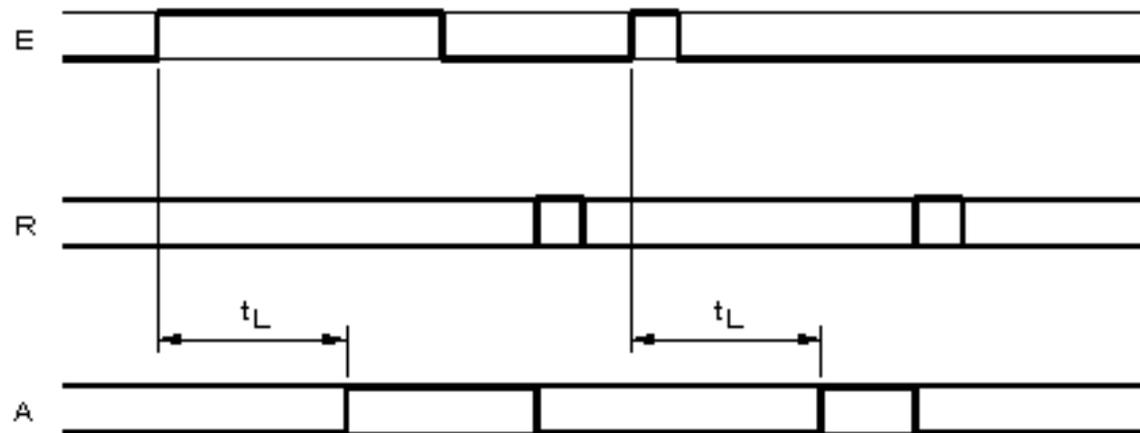
## Funktionsbeschreibung

Mit Signal "1" am Eingang E 0.1 wird der Ladebefehl für den programmierten Zeitwert, L S5T#10S, und der Startbefehl (ein kurzer Impuls genügt) für die speichernde Einschaltverzögerung ausgeführt.

Der Ausgang A 2.0 führt dann Signal "1", wenn die eingestellte Zeit abgelaufen ist. Der Ausgang A 2.0 wird ausgeschaltet (Signal "0"), wenn die Zeitfunktion mit Signal "0" am Eingang E 1.1 rückgesetzt wird.

Die digitalen Abfragen an den Ausgängen DUAL (= Dualzahl) bzw. DEZ (= Dezimalzahl) liefern den aktuellen Zeitwert (Istwert). Diese Zeitwerte können mit LADE- und TRANSFEROPERATIONEN weiter verarbeitet werden.

## Funktionsdiagramm



Siehe nächste Seite

# Speichernde Einschaltverzögerung

Programmdarstellung		
FUP	KOP	AWL
<p>Diagram of timer T4 in FUP notation. The timer is labeled T4. It has a set coil (S) with input E0.1, a time delay setting (TWW) of S5T#10S, and a reset coil (R) with input E1.1. The output (Q) is connected to A2.0. The timer is also labeled with S_DUAL and DEZ.</p>	<p>Diagram of timer T4 in KOP notation. The timer is labeled T4. It has a set coil (S) with input E0.1, a time delay setting (TWW) of S5T#10S, and a reset coil (R) with input E1.0. The output (Q) is connected to A2.0. The timer is also labeled with S_DUAL and DEZ.</p>	<pre> U  E0.1 L S5T#10S SS  T4 UN  E1.1 R  T4 NOP 0 NOP 0 U  T4 =  A2.0 </pre>

# Ausschaltverzögerung

## Funktionsbeschreibung

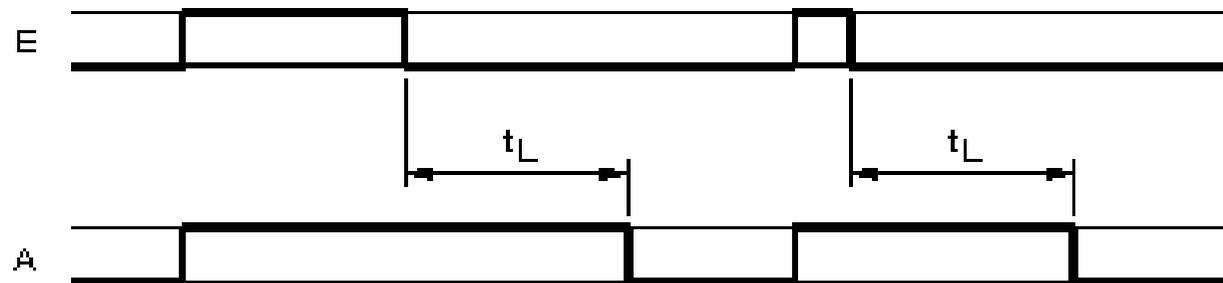
Mit Signal "1" am Eingang E 0.1 wird der Ausgang A 2.0 auf Signal "1" geschaltet. Wechselt das Signal am Eingang E 0.1 von Signal "1" auf "0" wird der Ladebefehl für den programmierten Zeitwert L S5T#10S und der Startbefehl für die eingestellte Laufzeit SA T5 ausgeführt. Der Ausgang A 2.0 bleibt auf Signal "1".

Erst nach Ablauf der eingestellten Zeit wird, um diese verzögert, der Ausgang auf Signal "0" geschaltet.

Der Ausgang A 2.0 führt immer dann Signal "1", wenn der Eingang E 0.1 Signal "1" führt oder die eingestellte Zeit läuft.

Die digitalen Abfragen an den Ausgängen DUAL (= Dualzahl) bzw. DEZ (= Dezimalzahl) liefern den aktuellen Zeitwert (Istwert). Diese Zeitwerte können mit LADE- und TRANSFEROPERATIONEN weiter verarbeitet werden.

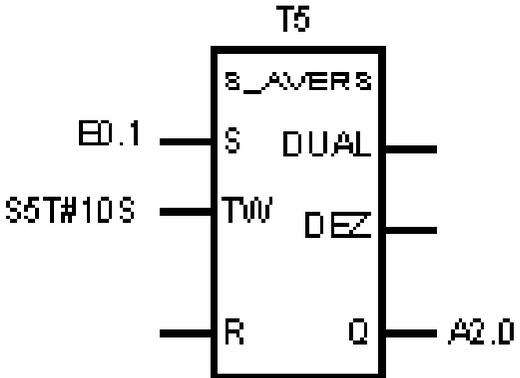
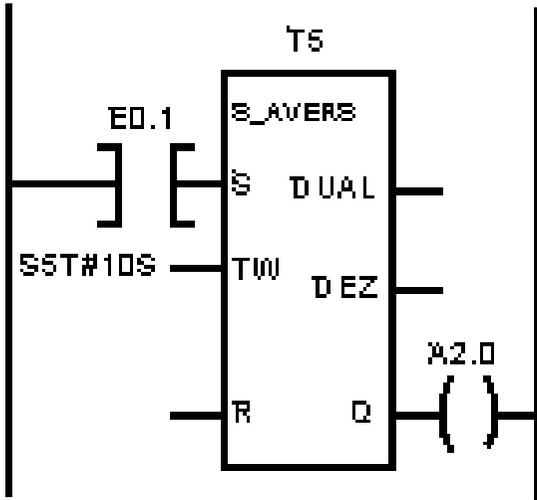
## Funktionsdiagramm



Siehe nächste Seite

# Ausschaltverzögerung

## Programmdarstellung

FUP	KOP	AWL
		<pre> U  E0.1 L S5T#10S SA T5 NOP 0 NOP 0 NOP 0 U  T5 =  A2.0 </pre>

# Darstellung von Bausteinen

	KOP	FBS	AWL
<b>FC</b>	<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: 80%;"> <p style="text-align: center;">FC</p> <p style="text-align: center;">E          EN</p> <hr style="width: 20%; margin: 5px auto;"/> <p style="text-align: center;">Par</p> <hr style="width: 20%; margin: 5px auto;"/> <p style="text-align: center;">Par      Par</p> </div>	<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: 80%;"> <p style="text-align: center;">FC</p> <p style="text-align: center;">E          Par</p> <hr style="width: 20%; margin: 5px auto;"/> <p style="text-align: center;">Par</p> <hr style="width: 20%; margin: 5px auto;"/> <p style="text-align: center;">Par      EN</p> </div>	Call FC1 Par1: ... Par2: ... Par3:
<b>FB</b>	<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: 80%;"> <p style="text-align: center;">DB1</p> <div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: 80%;"> <p style="text-align: center;">FC</p> <p style="text-align: center;">E          EN</p> <hr style="width: 20%; margin: 5px auto;"/> <p style="text-align: center;">Par</p> <hr style="width: 20%; margin: 5px auto;"/> <p style="text-align: center;">Par      Par</p> </div> </div>	<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: 80%;"> <p style="text-align: center;">DB</p> <div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: 80%;"> <p style="text-align: center;">FC</p> <p style="text-align: center;">E          Par</p> <hr style="width: 20%; margin: 5px auto;"/> <p style="text-align: center;">Par</p> <hr style="width: 20%; margin: 5px auto;"/> <p style="text-align: center;">Par      EN</p> </div> </div>	Call FC1, DB1 Par1: ... Par2:

**das war's**  
**Viel Erfolg wünscht**  
**U. Ohm**