

Kapitel 3: SPS-Verknüpfungssteuerung

3.1 Boolesche Algebra

Sehr kurze Wiederholung
des Stoffes aus anderen
Lehrveranstaltungen.

3.2 Aufbau und Funktion einer SPS

3.3 Kontaktplan

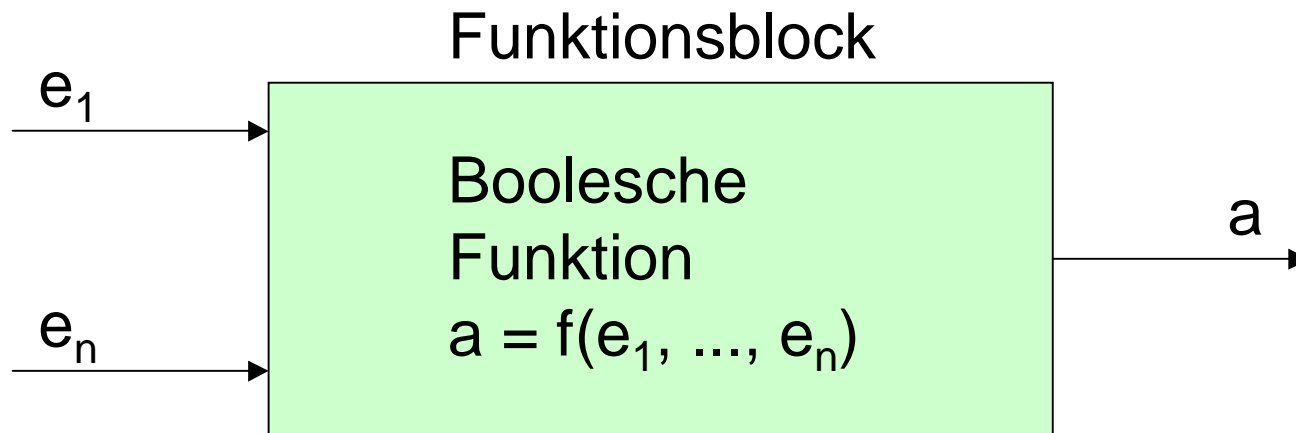
3.4 Funktionsbausteinsprache (FBS)

3.5 Speicher, Zähler und Zeitglieder

Boolesche Algebra

Auch Schaltalgebra: Beschreibung logischer Verknüpfungen zweiwertiger bzw. binärer Größen.

Bei einer Verknüpfungssteuerung bestimmen sich die Ausgangssignale ausschließlich aus den momentanen Werten der Eingangssignale. Eine Verknüpfungssteuerung hat demzufolge keine speichernden Eigenschaften



Boolesche Grundfunktionen

Konjunktion (AND)

$$e_1 \wedge e_2$$

Disjunktion (OR)

$$e_1 \vee e_2$$

Negation (NOT)

$$\neg e_1 \text{ bzw. } \overline{e_1}$$

Antivalenz (XOR)

$$e_1 \neq e_2$$

Äquivalenz (=)

$$e_1 = e_2$$

e_1	e_2	$e_1 \wedge e_2$	$e_1 \vee e_2$	$\neg e_1$	$e_1 \neq e_2$	$e_1 = e_2$
0	0	0	0	1	0	1
0	1	0	1	1	1	0
1	0	0	1	0	1	0
1	1	1	1	0	0	1

Insgesamt 16 verschiedene Kombinationen, z.B. NOR, NAND, Implikation usw.

Grundgesetze

1. Kommutativgesetz	$e_1 \wedge e_2 = e_2 \wedge e_1$
2. Distributivgesetz	$e_1 \wedge (e_2 \vee e_3) = (e_1 \wedge e_2) \vee (e_1 \wedge e_3)$
3. Assoziativgesetz	$e_1 \wedge (e_2 \wedge e_3) = (e_1 \wedge e_2) \wedge e_3$
4. Absorptionsgesetz	$e_1 \wedge (e_1 \vee e_2) = e_1$ $e_1 \vee (e_1 \wedge e_2) = e_1$
5. Idempotenzgesetz	$e_1 \wedge e_1 = e_1$
6. De Morgan Regel	$e_1 \wedge e_2 = \overline{\overline{e_2} \vee \overline{e_1}}$
7. Komplement	$e_1 \wedge \overline{e_1} = 0$ $e_1 \vee \overline{e_1} = 1$

Mit Hilfe der angegebenen Grundgesetze lassen sich boolesche Funktionen vereinfachen und an bestimmte Operatoren anpassen. Bei einer SPS ist dies aber nur noch bei der Programmierung in Kontaktplan relevant.

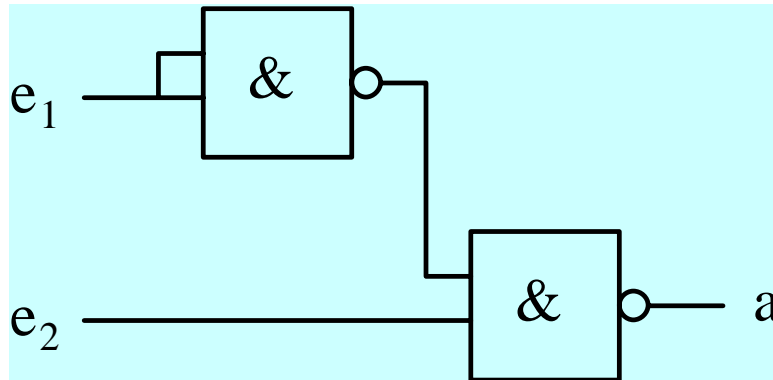
Aus der Wertetafel lassen sich die Booleschen Funktionen durch Bildung der disjunktiven und konjunktiven Normalformen (Min- und Maxterme) ableiten. Mit dem Karnaugh-Diagramm lassen sie sich anschließend vereinfachen.

Beispiel 1

Umformung für TTL-Technik (NAND Gatter)

$$a = e_1 \vee \overline{e_2} = \overline{\overline{e_1} \wedge e_2}$$

De Morgan Regel



Beispiel 2

Vereinfachung einer booleschen Funktion
(Speicherplatzreduktion und Beschleunigung)

$$\begin{aligned} a &= \overline{(\overline{e_1} \vee \overline{e_2})} \vee (\overline{e_1} \wedge e_2) && \text{De Morgan} \\ &= (e_1 \wedge e_2) \vee (\overline{e_1} \wedge e_2) && \text{Distributivgesetz} \\ &= (e_1 \vee \overline{e_1}) \wedge e_2 && \text{Komplement} \\ &= e_2 \end{aligned}$$

Weitere Literatur

Für Studierende OHNE Vorkenntnisse aus Lehrveranstaltungen, wie Digitaltechnik oder Technische Informatik, seien die folgenden Buchhinweise gegeben:

- Wellenreuther, Zastrow: Steuerungstechnik mit SPS.
Vieweg 1995
Teil I: Kapitel 3
Teil II: Kapitel 4
- Hütte: Die Grundlagen der Ingenieurwissenschaften.
Springer 1989
Teil I: Kapitel 14 (Binäre Steuerungstechnik)
Teil J: Kapitel 1 und 2 (Schaltnetze – Schaltwerke)

Kapitel 3: SPS-Verknüpfungssteuerung

3.1 Boolesche Algebra

3.2 Aufbau und Funktion einer SPS

3.3 Kontaktplan

3.4 Funktionsbausteinsprache (FBS)

3.5 Speicher, Zähler und Zeitglieder

SPS-Automatisierungsgerätes

Modularer Aufbau:

- Netzgerät (Stromversorgung)
- Verarbeitungseinheit (CPU)
- Digital I/O
- Analog I/O
- Feldbus-Schnittstelle
- Programmiergerät (heute: PC)

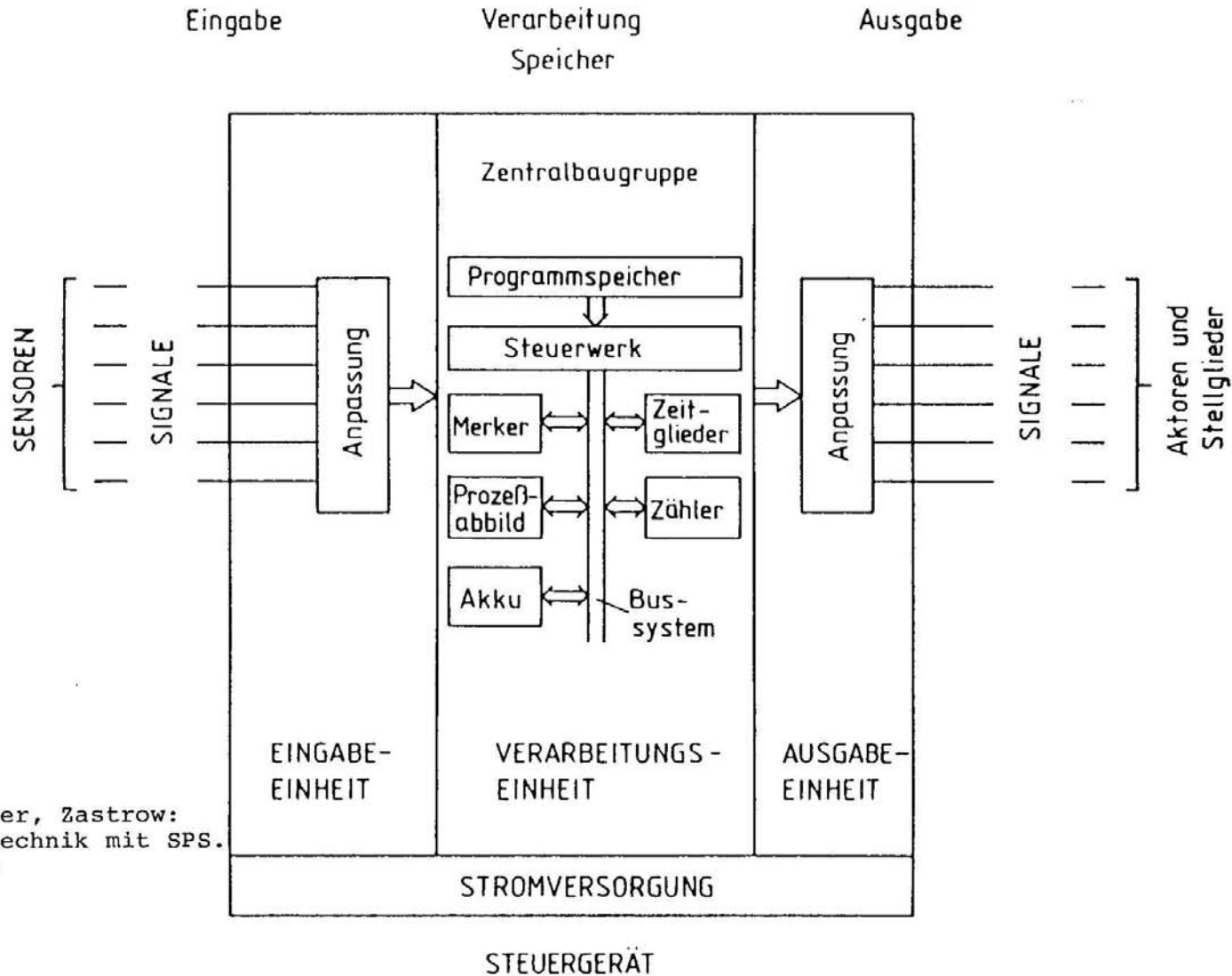
Standardisierte Betriebsarten:

- Neustart bei Initialisierung
- Warmstart bzw. Wiederanlauf bei Störung
- Automatikbetrieb
- Einzelschritt bzw. Tippbetrieb



CPU mit:
Steuer- und Rechenwerk,
Akku, Programmspeicher,
Merkern, Zeitgliedern,
Zählern, Prozessabbild
und Bussystem

Aufbau einer SPS



aus:
Wellenreuther, Zastrow:
Steuerungstechnik mit SPS.
Vieweg 1995

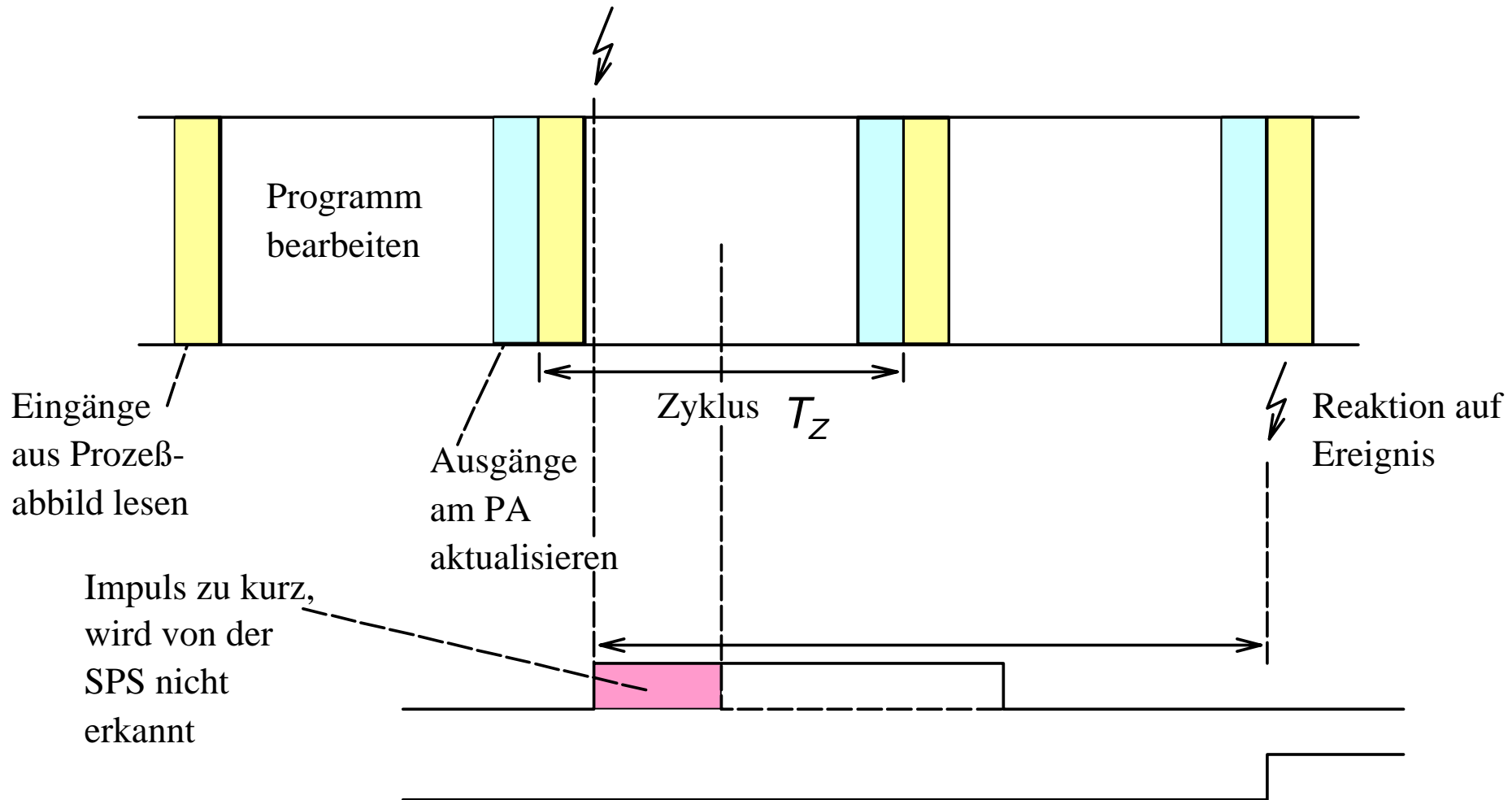


Prozessabbild

Speicherbereich zur Aufnahme der Signalwerte der Ein- und Ausgänge:

- Konstante Eingangssignale während eines Programmzyklus.
- Ausgabewerte werden zwischengespeichert, d.h. kurzzeitige Änderungen während des Zyklus haben nach außen keine Wirkung.
- maximale Reaktionszeit ist etwa doppelte Zyklusdauer T_Z
d.h. bei gegebener maximaler Reaktionszeit ergibt sich eine Begrenzung der Programmlänge (typischer Wert: 1000 Anweisungen/ms)
- minimale Dauer eines Eingangssignals ist etwa Zyklusdauer T_Z
- aufgrund von Programmverzweigungen ist die Zyklusdauer nicht konstant
 - ➔ Überwachung der Zyklusdauer
 - ➔ Zyklendurchlauf in regelmäßigen Abständen $T_A > T_{Zmax}$
- spezielle SPS Geräte erlauben den direkten Zugriff auf die Prozesssignale (ohne Prozessabbild)
 - ➔ langsamer

Maximale Reaktionszeit

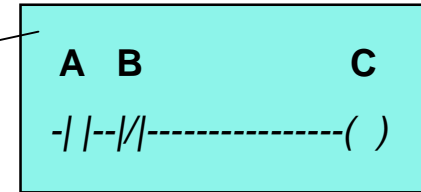


Programmiersprachen nach IEC 1131-3

KOP

Kontaktplan

(engl. Ladder Diagram, LD)



FBS

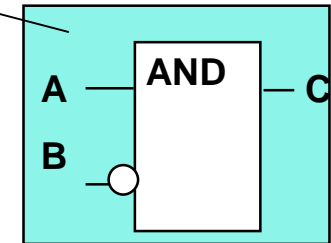
Funktionsbausteinsprache

(engl. Function Block Diagram, FBD)

ST

Strukturierter Text

(engl. Structured Text, ST)



AS

Ablaufsprache

(engl. Sequential Function Chart, SFC)

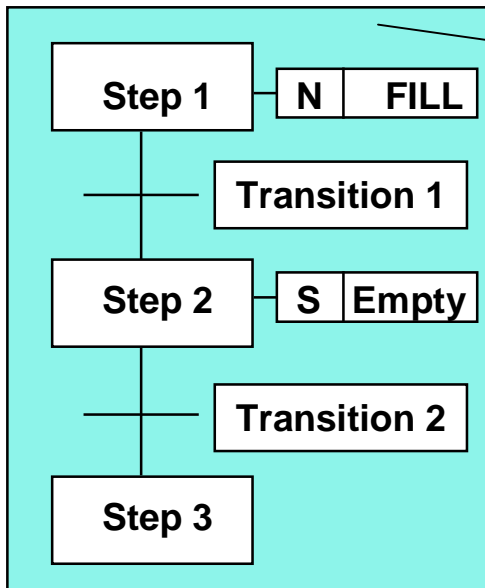
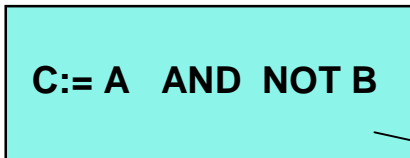
AWL

Anweisungsliste

(engl. Instruction List, IL)



Standardisierung und Zertifizierung durch PLCopen.



Kapitel 3: SPS-Verknüpfungssteuerung

3.1 Boolesche Algebra

Engl.: Ladder Diagram

3.2 Aufbau und Funktion einer SPS

3.3 Kontaktplan

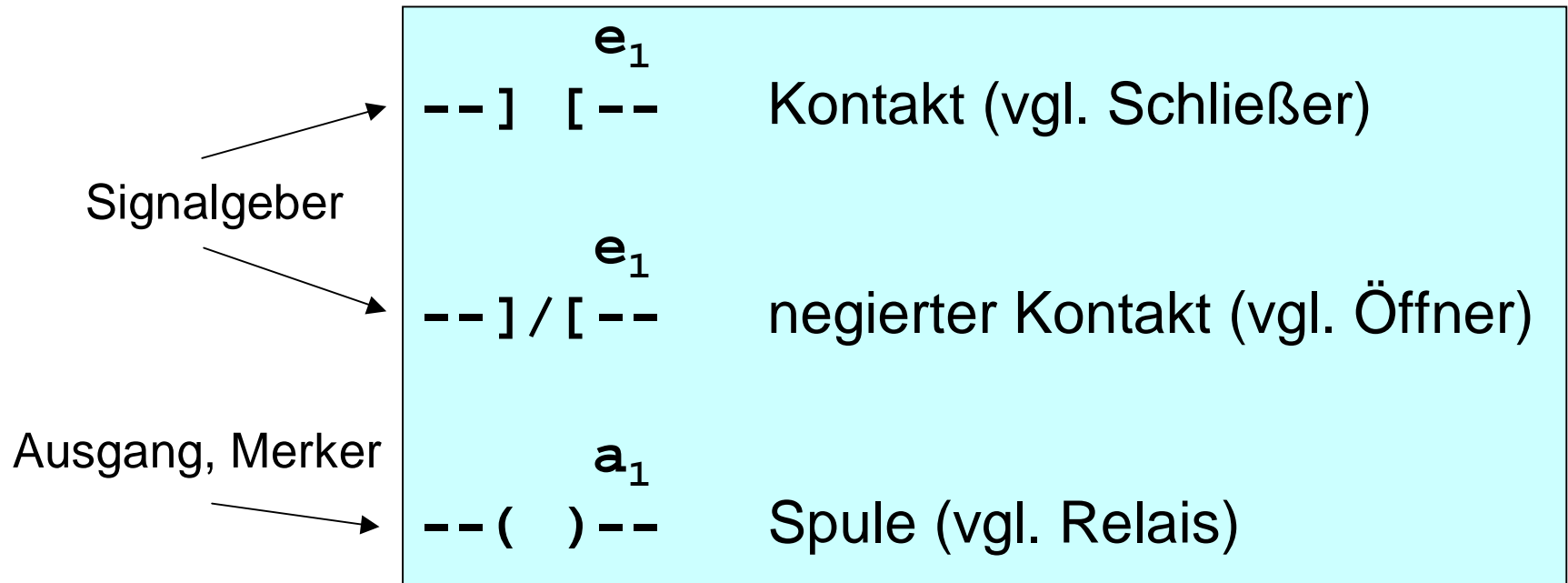


3.4 Funktionsbausteinsprache (FBS)

3.5 Speicher, Zähler und Zeitglieder

Kontaktplan (KOP) - Grundelemente

Historisch begründete, grafische Form der SPS Programmierung
(Verbindungsprogrammierte Relaissteuerung für einfache VKS)



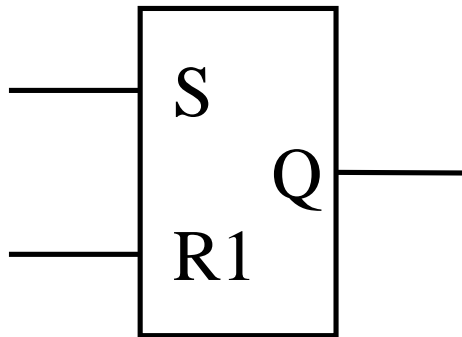
Der Wirkungszusammenhang wird durch den fiktiven Stromfluss von links nach rechts beschrieben. Bei mehreren Netzwerken erfolgt die Bearbeitung von oben nach unten.

- UND-Verknüpfung durch *Reihenschaltung*
- ODER-Verknüpfung durch *Parallelschaltung*

Weitere Elemente

Weitere Elemente der Kontaktplandarstellung:

- Sprünge, Sprungmarken und Rücksprünge
- Flankenerkennung
- Speicherelemente (Merker, Flip-Flop)
- Funktionen (kein Speicher, EIN Funktionswert)
- Funktionsbausteine (mit Speicher, mehrere Rückgabewerte)

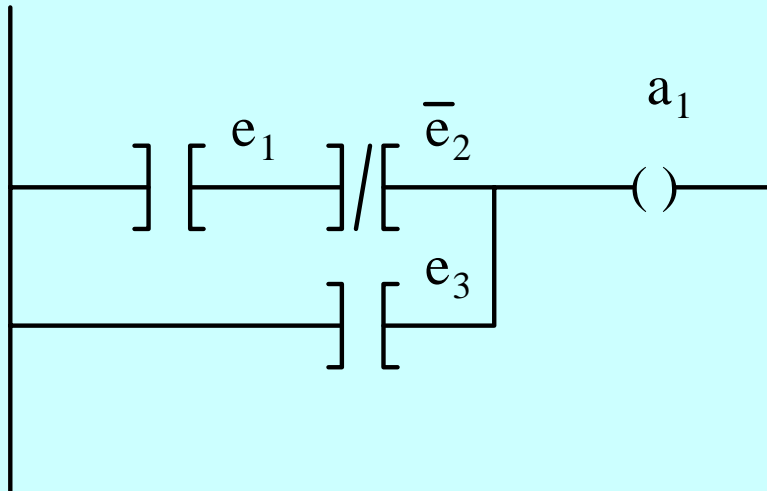


RS-Flip-Flop

- Priorität hat Setzen (R1)

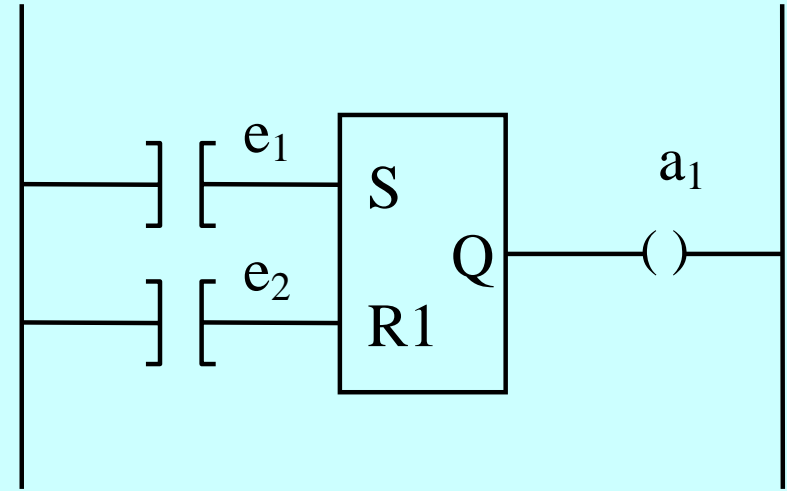
KOP - Beispiele

$$a_1 = e_3 \vee (e_1 \wedge \bar{e}_2)$$



Beispiel 1

$$a_1 = (a_1 \vee e_1) \wedge \bar{e}_2$$



Beispiel 2

Übersetzung KOP in AWL

Die Kontaktplandarstellung wird in ausführbaren Programmcode übersetzt.
Bei S5-Steuergeräten entspricht dies den AWL-Anweisungen:

Beispiel 1:	Beispiel 2:	mit RS-Glied	ohne RS-Glied
U E0.1		U E0.1	U E0.1
UN E0.2		S A0.1	O A0.1
O E0.3		U E0.2	UN E0.2
= A0.1		R A0.1	= A0.1

Nach IEC 61131-3 ergeben sich die folgenden AWL-Anweisungen:

Beispiel 1:	Beispiel 2:	mit RS-Glied	ohne RS-Glied
LD e1		LD e1	LD e1
ANDN e2		S a1	OR a1
OR e3		LD e2	ANDN e2
ST a1		R a1	ST a1

Kapitel 3: SPS-Verknüpfungssteuerung

3.1 Boolesche Algebra

3.2 Aufbau und Funktion einer SPS

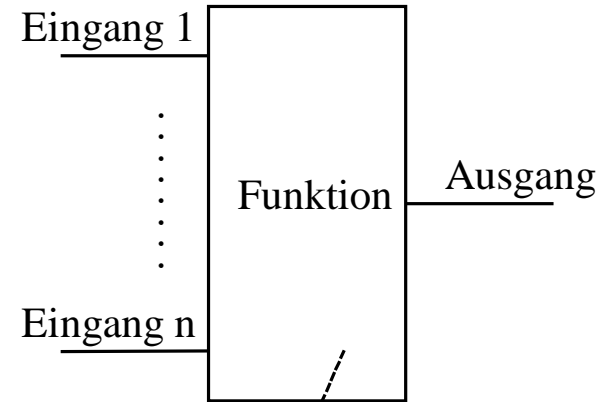
3.3 Kontaktplan

3.4 Funktionsbausteinsprache (FBS)

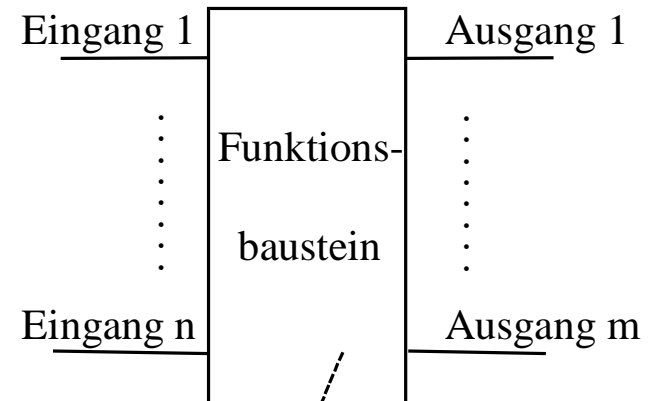
3.5 Speicher, Zähler und Zeitglieder

Funktionsbausteinsprache (FBS)

- Graphische Programmiersprache für Verknüpfungssteuerungen in Form einzelner Netzwerke. Die Wirkungsrichtung (Auswertereihenfolge) entsteht durch Verkettung der Funktionen und Funktionsbausteine, wobei links immer die Eingänge und rechts immer die Ausgänge gezeichnet werden.
- Ein- und Ausgänge können miteinander beliebig verbunden werden. Insbesondere sind Rückkopplungen möglich.
- Auch hier sind Sprünge, Marken und Rücksprünge möglich (unübersichtlich).



kein Speicher



mit Speicher



invertierter Eingang



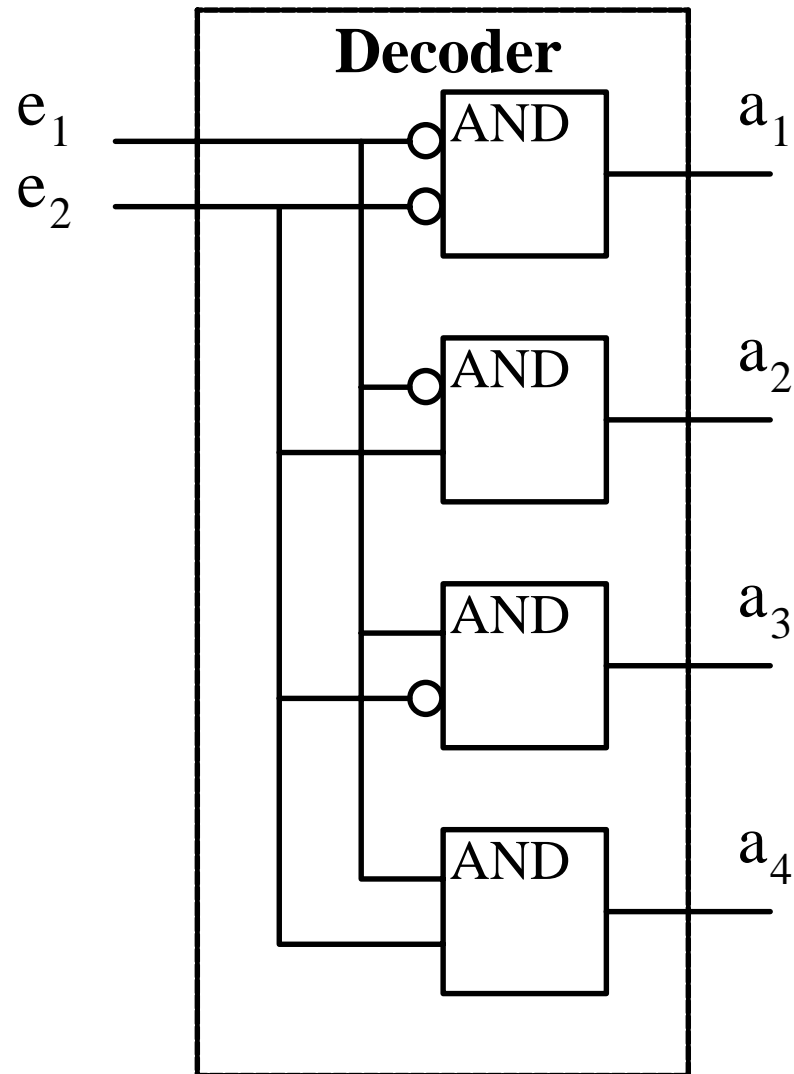
invertierter Ausgang

Beispiel: Decoder

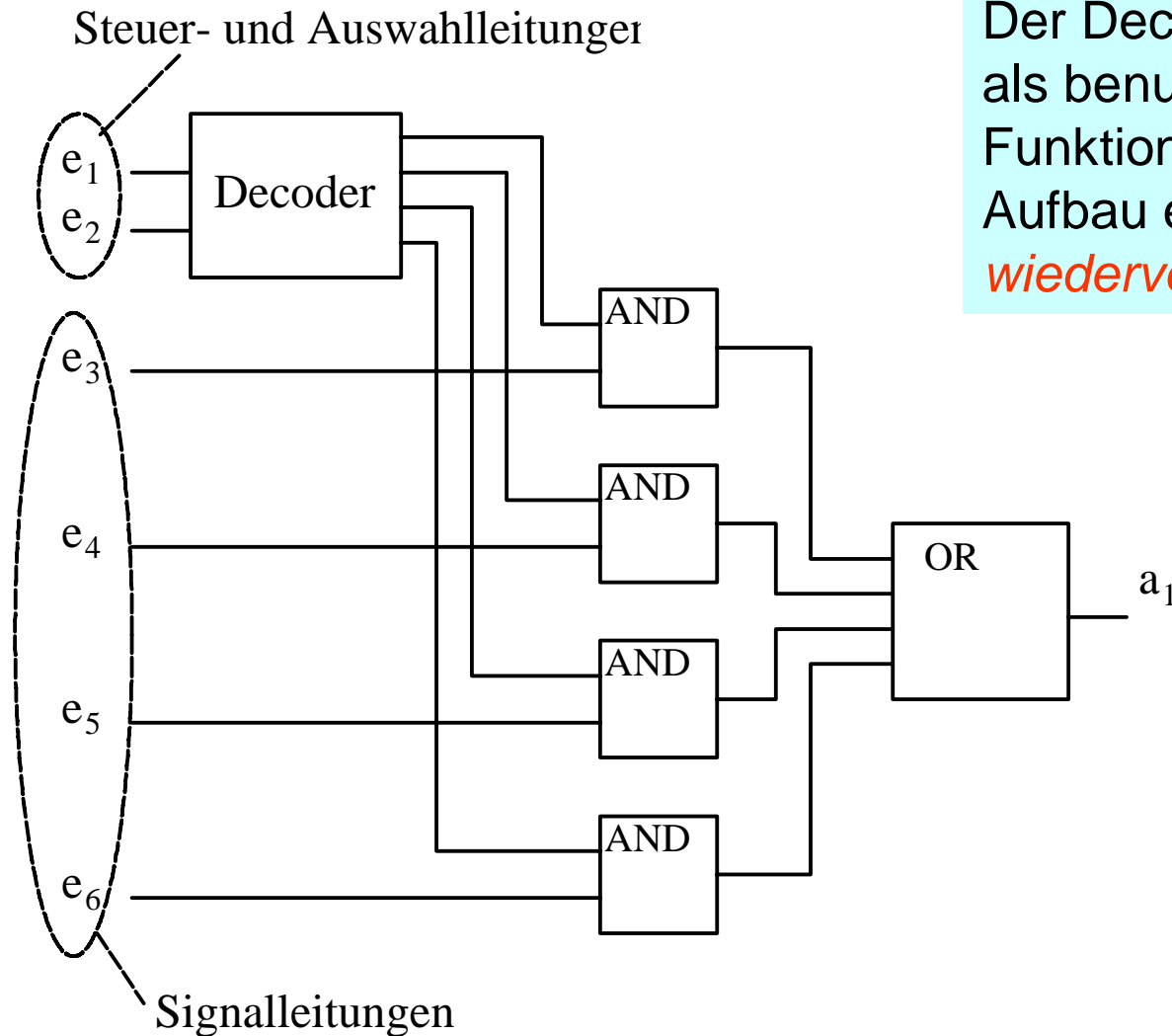
Wertetafel

e_1	e_2	a_1	a_2	a_3	a_4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

FBS Netzwerk Decoder



FBS Netzwerk Multiplexer



Der Decoder selbst kann als benutzerdefinierter Funktionsbaustein zum Aufbau eines Multiplexers *wiederverwendet* werden.

Kapitel 3: SPS-Verknüpfungssteuerung

3.1 Boolesche Algebra

3.2 Aufbau und Funktion einer SPS

3.3 Kontaktplan

3.4 Funktionsbausteinsprache (FBS)

3.5 Speicher, Zähler und Zeitglieder

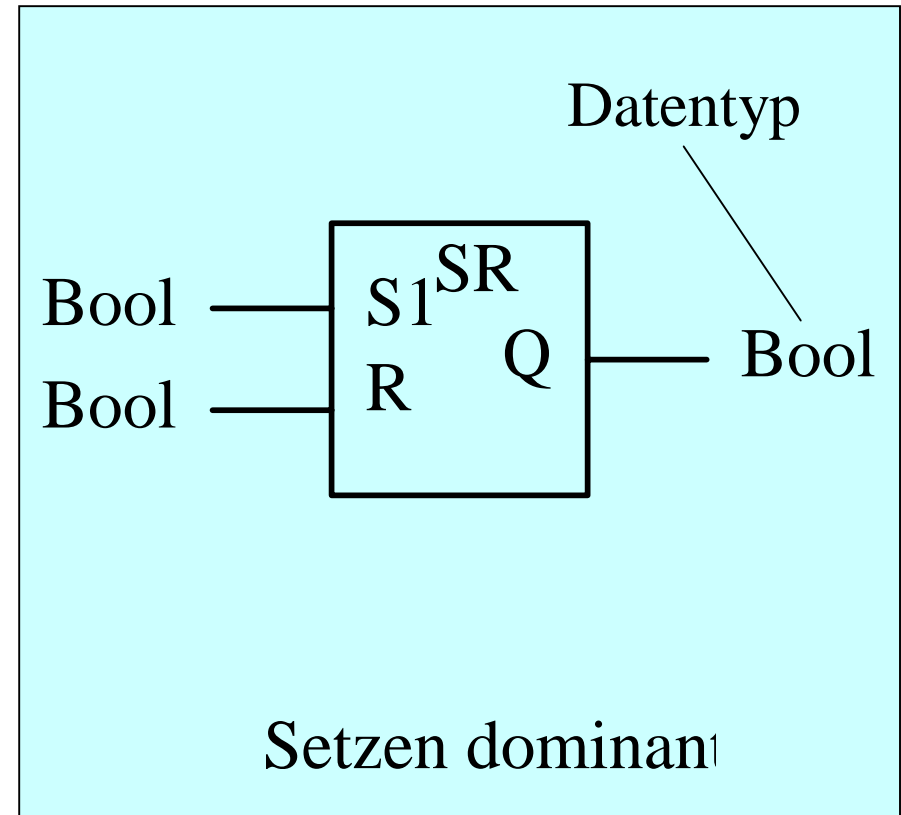
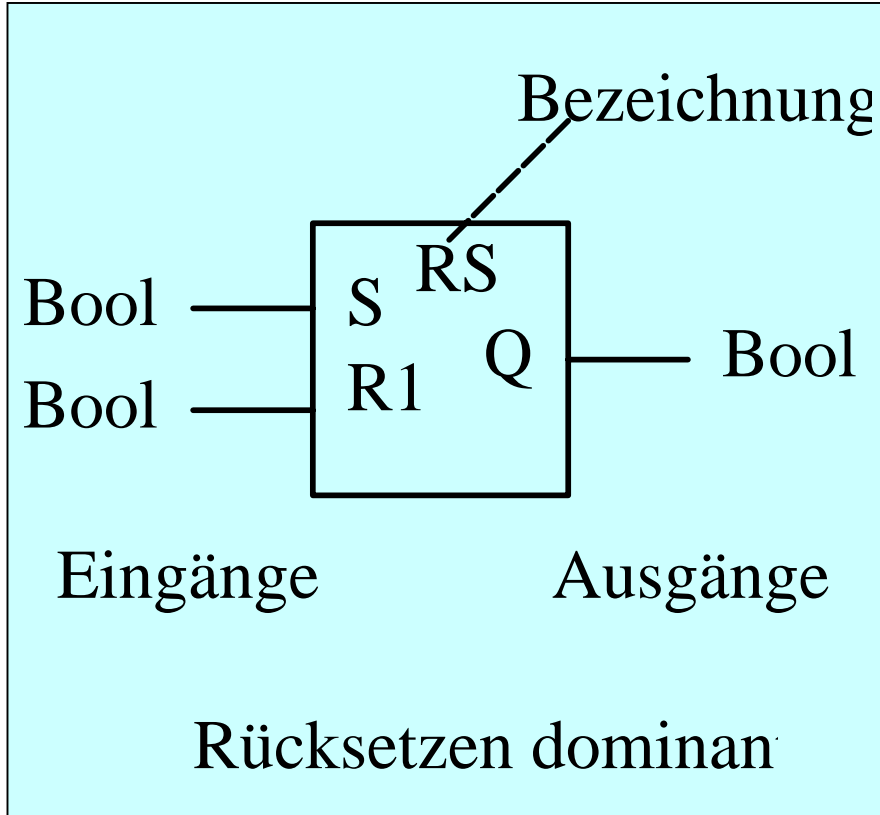
Speicher, Zähler und Zeitglieder

In vielen Steuerungsaufgaben sind häufig Teilprobleme der folgenden Art zu lösen:

- ein Taster soll zum Ein-, Aus- oder Umschalten verwendet werden;
→ Speicherglieder Flankenerkennung
- die Anzahl der Werkstücke in einer Pufferstrecke soll kontrolliert werden
→ Auf- und Abwärtszähler
- das Einschalten zweier Maschinen soll zeitversetzt erfolgen
→ Ein- und Ausschaltverzögerung

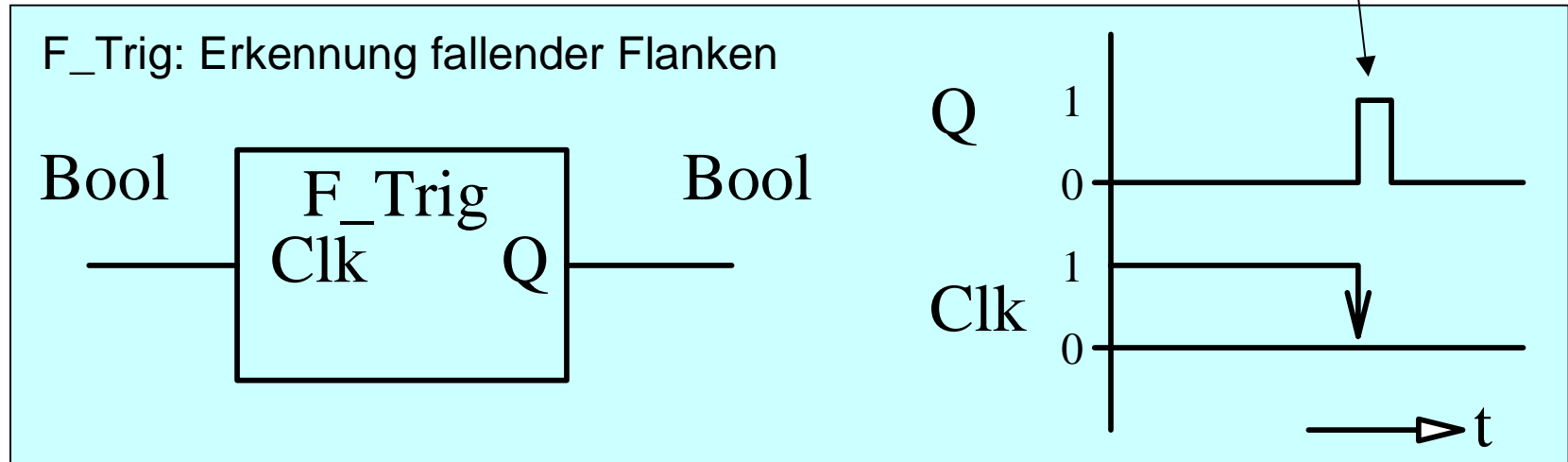
Zur Realisierung sind in der IEC 61131-3 Standardfunktionsbausteine vorgesehen. Die Standard-FBs können in Programmiersystemen aus vorbereiteten Bibliotheken ausgewählt werden.

RS: Bistabiler Funktionsbaustein

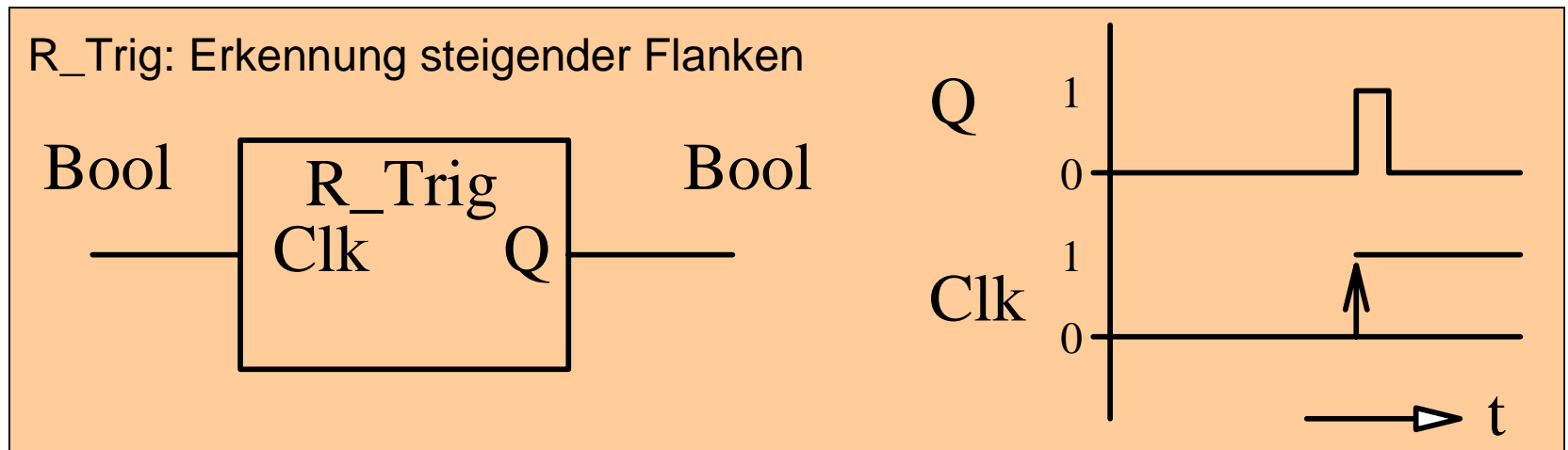


Flankenerkennung

Pulsdauer genau ein SPS-Zyklus

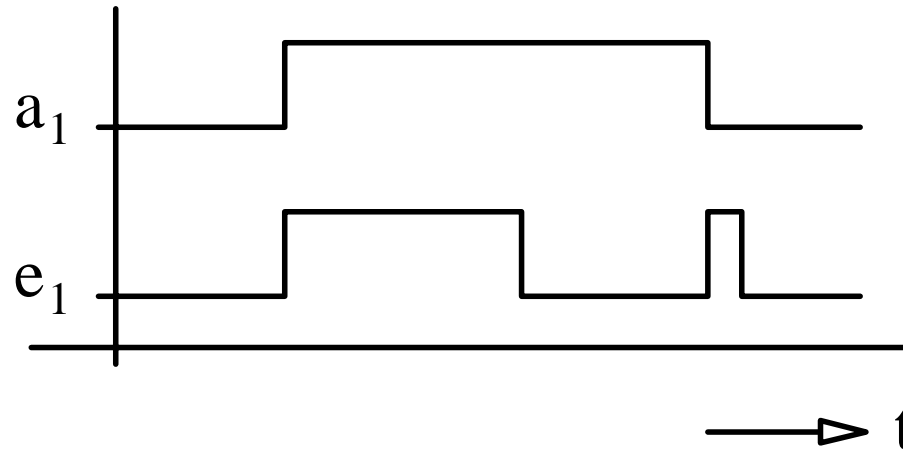


Nach einem Übergang 1->0 am Eingang bleibt am Ausgang bis zur nächsten Ausführung (T_z) eine 1; danach schaltet der Ausgang wieder auf 0.

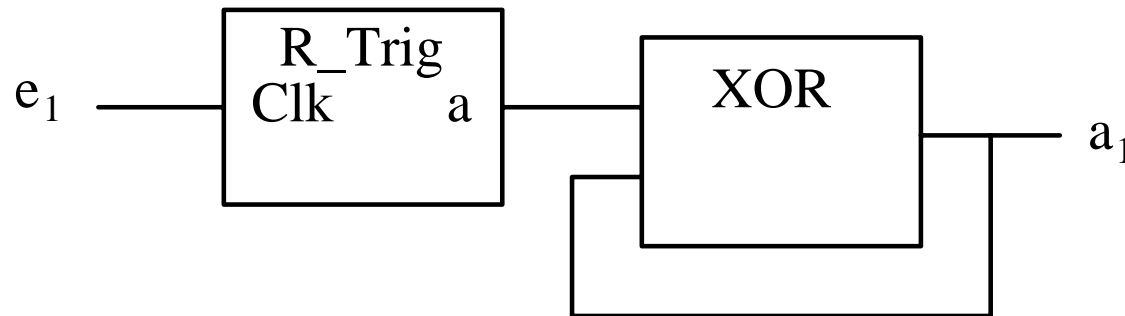


Beispiel

Ein- und Aussteuerung mit einem Taster



Lösung: Realisierung mit Standardfunktionen



Zeitgeberbausteine

Zeitgeberbaustein Einschaltverzögerung (TON)

IN Start Zeitgeber (steigende Flanke)

PT ... Voreinstellung Zeit

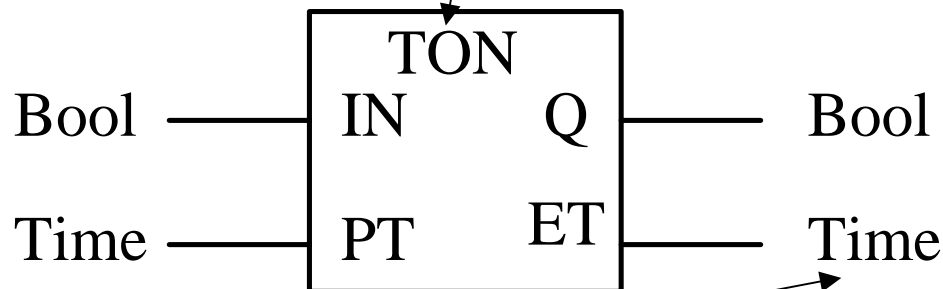
Q Ausgang

ET ... interne Zeit

TON: Einschaltverzögerung

TOF: Ausschaltverzögerung

TP: Pulsgeber



Angabe von Zeitdauern als Konstante:

T#14.7ms

TIME#3h_12m_4.3s

Einheiten:

d Tag

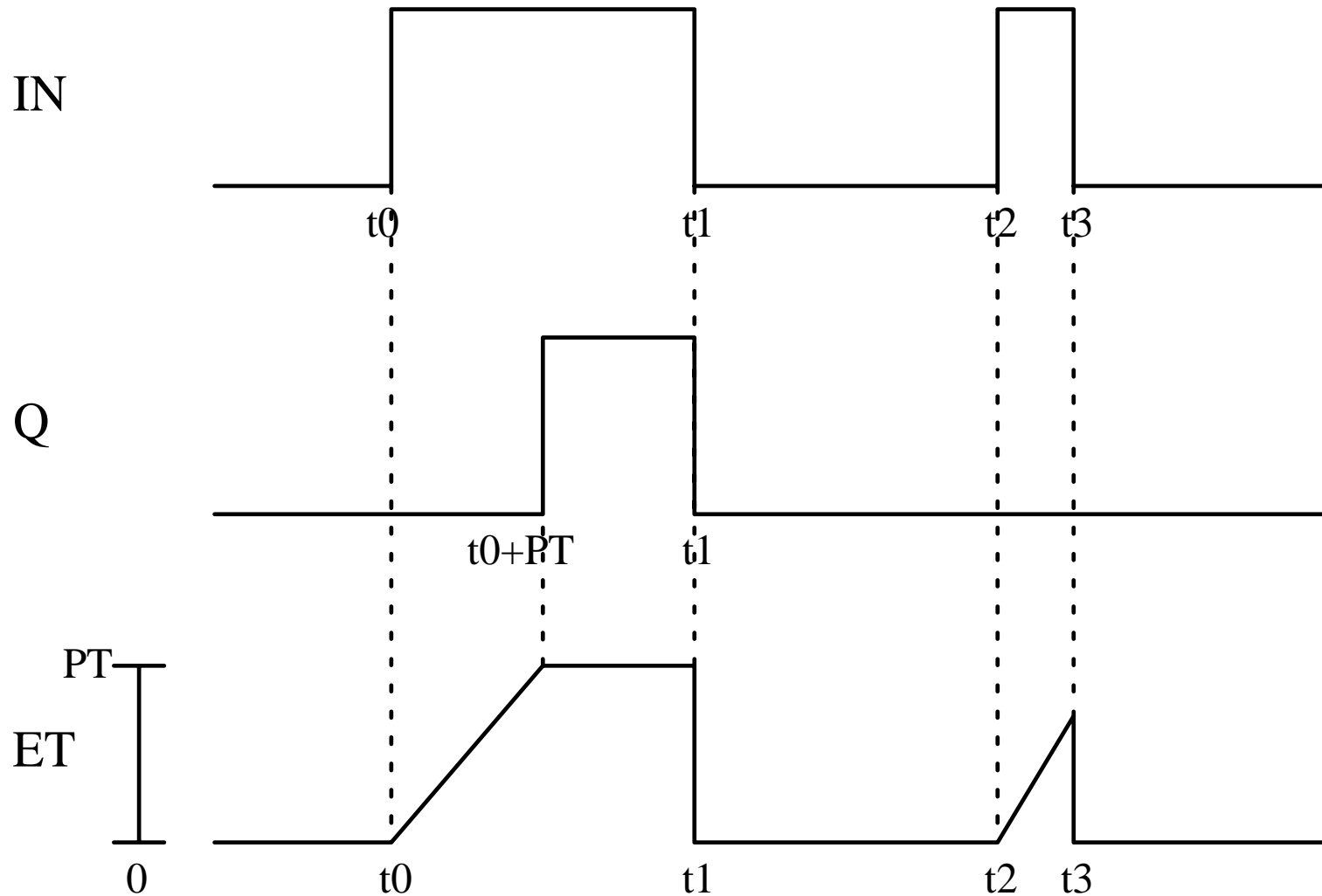
h Stunde

m Minute

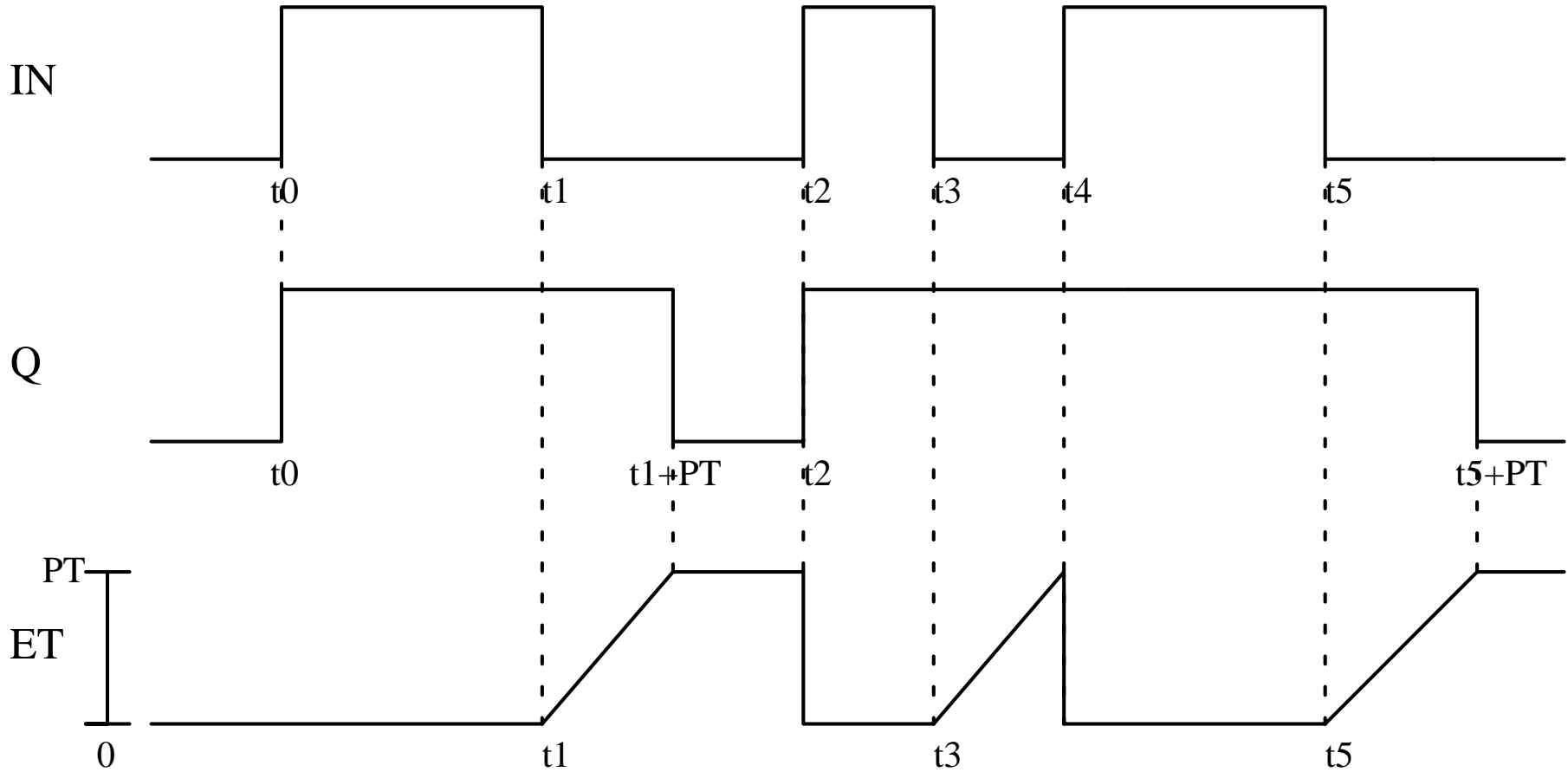
s Sekunde

ms .. Millisekunde

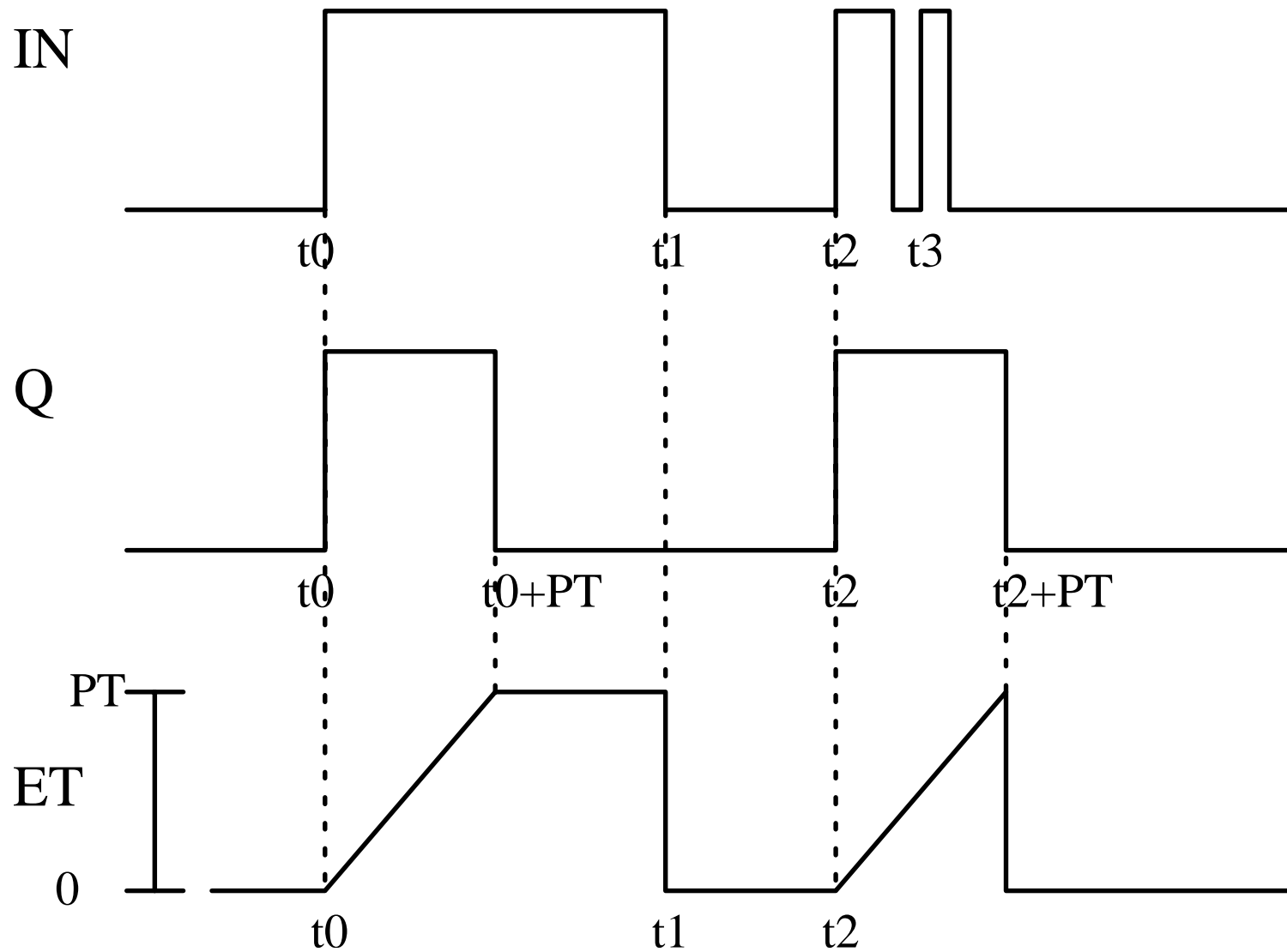
Zeitdiagramm TON



Zeitdiagramm TOF

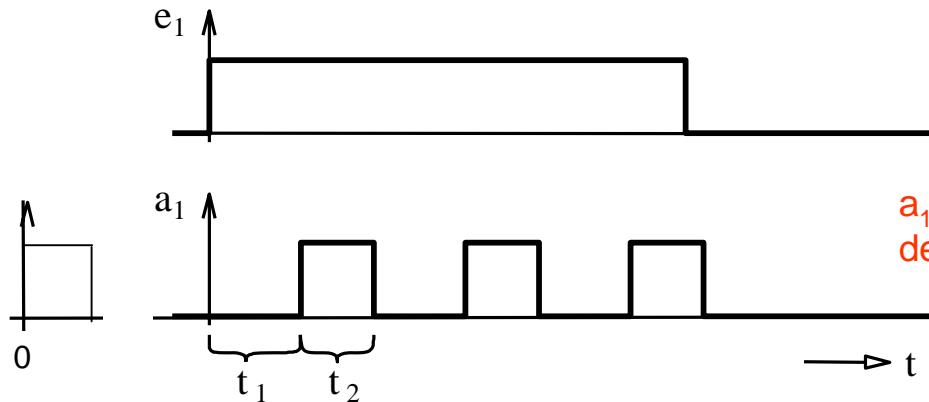


Zeitdiagramm TP



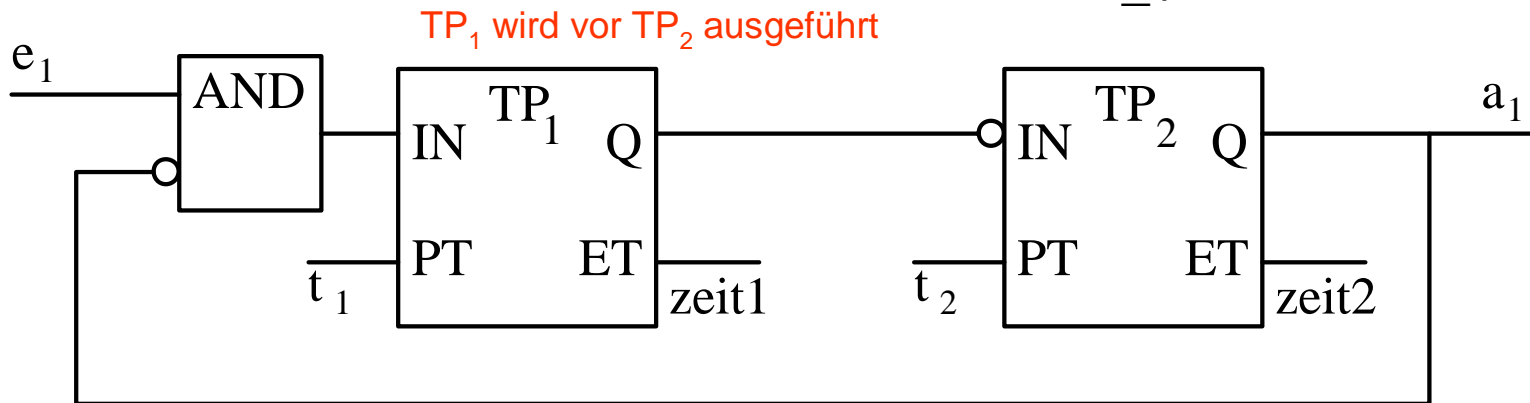
Beispiel: Blinklicht

Blinklichtsteuerung aus elementaren Bausteinen: Wenn das Eingangssignal e_1 auf True wechselt (Flanke) soll das Blinklicht im Wechsel für die Zeitdauer t_1 ausgeschaltet und für die Dauer t_2 eingeschaltet werden. Wenn e_1 wieder auf False steht, soll das Blinklicht wieder ausgeschaltet werden.



a_1 ist beim Einschalten für die Dauer t_2 des Gesamtsystems einmalig 1!

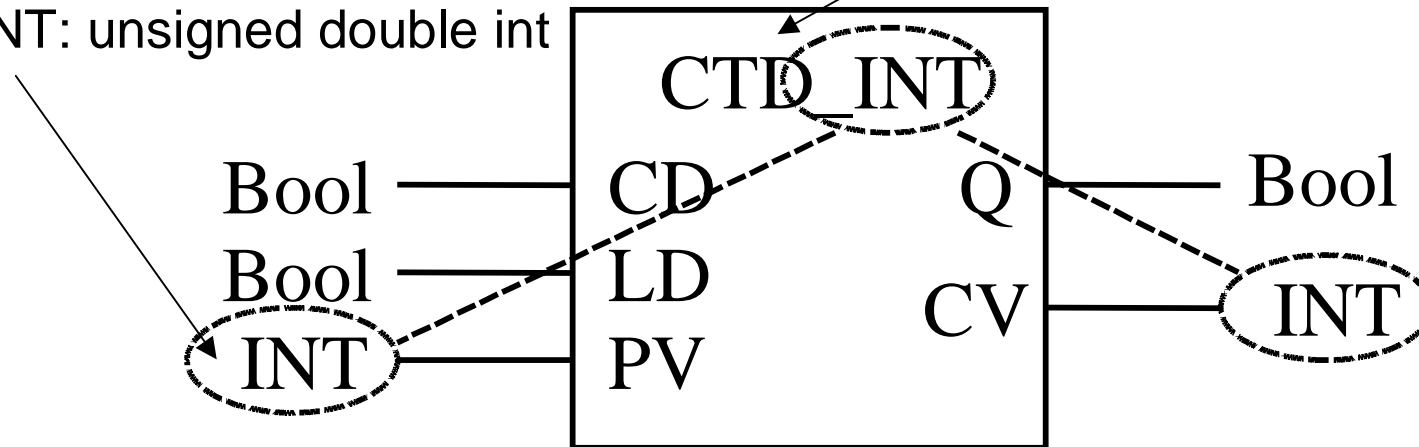
```
VAR zeit1, zeit2: TIME;
END_VAR
```



Zählerbausteine

oder:
 DINT: double int
 UINT: insigned int
 UDINT: unsigned double int

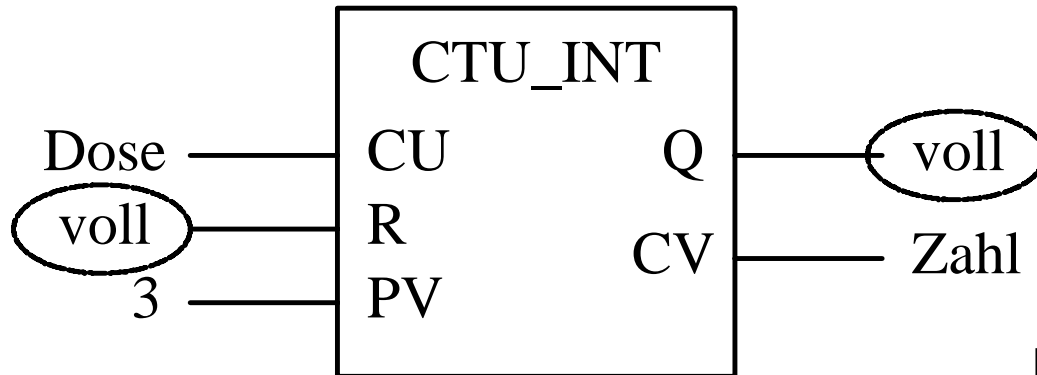
CTD: Abwärtszähler
 CTU: Aufwärtszähler
 CTUD: Auf-/Abwärtszähler



Abwärtszähler:
 CD: Triggereingang
 LD: Lade Daten
 PV: Voreinstellungswert
 Q: Ausgang
 CV: Zählwert

auch:
 CU: Triggereingang
 R: Rücksetzen
 QU: $CV \geq PV$
 QD: $CV \leq 0$

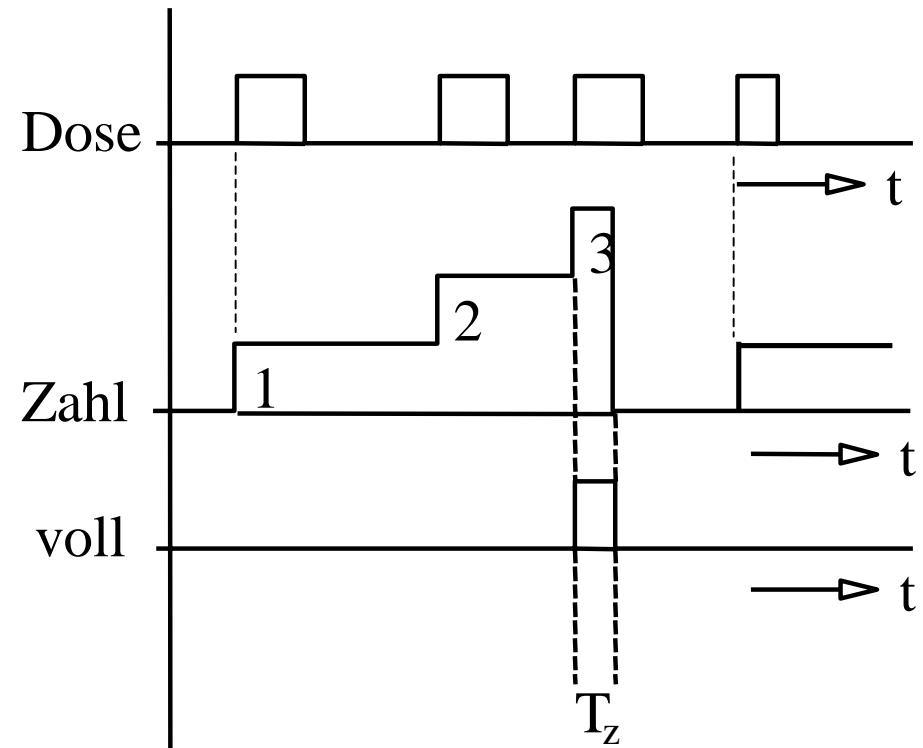
Aufwärtszähler



```

VAR    Dose,voll:BOOL;
        Zahl:INT;
END_VAR

```



Aufgabe

Gegeben ist die boolesche Funktion:

$$a = \overline{(e_1 \wedge e_2)} \vee e_3$$

Wie lautet die Realisierung dieser Funktion in:

- a) Funktionsbausteinsprache,
- b) Kontaktplan,
- c) Strukturiertem Text (nächstes Kapitel),
- d) Anweisungsliste (nächstes Kapitel)?

Zusammenfassung

Es wurde der Begriff der Verknüpfungssteuerung eingeführt. Dazu wurden zunächst elementare Begriffe der Booleschen Algebra wiederholt.

Verknüpfungssteuerungen werden vor allem von den Programmiersprachen Kontaktplan und Funktionsbausteinsprache unterstützt.

Die standardisierten Funktionsbausteine wurden vorgestellt:

- RS und SR
- F_TRIG und R_TRIG
- TON, TOFF und TP
- CTD, CTU und CTUD