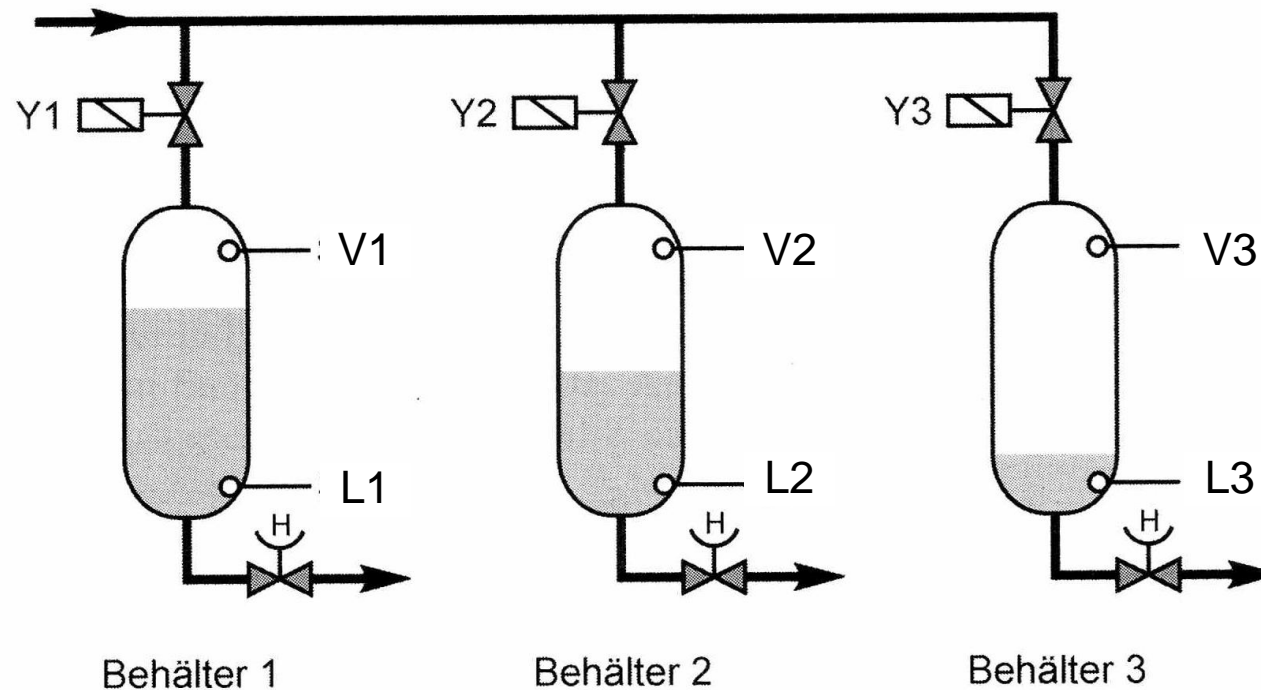


BEISPIELAUFGABE – 3 BEHÄLTER FÜLLEN

Entwerfen Sie ein Petrinetz, welche folgende Funktionalität abbildet:

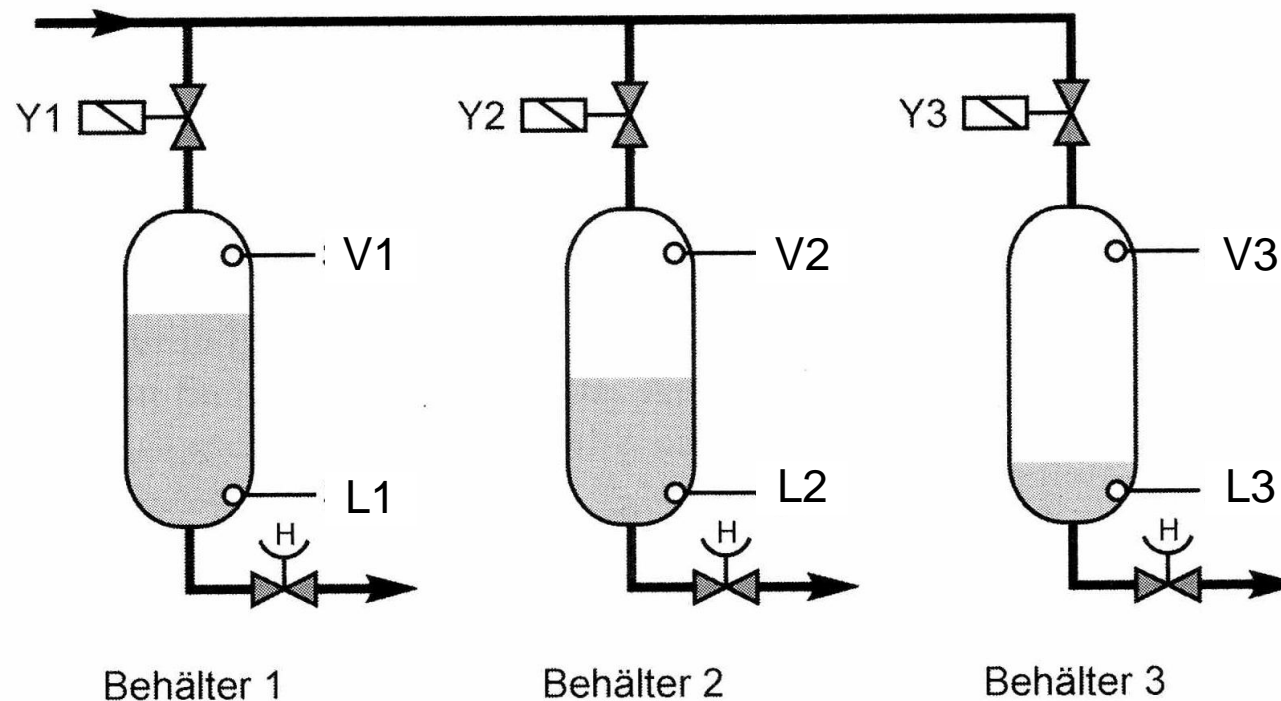
Melden die Sensoren L_i einen leeren Behälter, dann werden die Ventile Y_i geöffnet um die Behälter bis $V_i=1$ zu füllen. Es soll dabei zu einem Zeitpunkt nur ein Behälter gefüllt werden. Die Füllung erfolgt in Reihenfolge der Leermeldungen.



BEISPIELAUFGABE – 3 BEHÄLTER FÜLLEN - ZYKLISCH

Entwerfen Sie ein Petrinetz, welche folgende Funktionalität abbildet:

- Zyklische Überprüfung der Tanks (erst Tank1, Tank2, Tank3, dann wieder Tank1, ...).
- Falls nicht voll, wird jeweiliger Tank vollständig gefüllt



MATHEMATISCHE BESCHREIBUNG VON PETRINETZEN

Autonomes Petrinetz PN

Tupel $PN = (P, T, Pre, Post)$ mit

$P = \{P_1, P_2, \dots, P_n\}$ Menge der Plätze/Stellen

$T = \{T_1, T_2, \dots, T_m\}$ Menge der Transitionen

$Pre = \{ \dots \}$ Menge der Prekanten

$Post = \{ \dots \}$ Menge der Postkanten

Markierungsvektor m_k

$m_k^T = [m_k(P_1), m_k(P_2), \dots, m_k(P_n)]$

Netzmarkierung („1“= markiert, „0“=nicht markiert)

k = k -ter Schritt ($k=0$ „Initialisierung“)

Anfangsmarkiertes Petrinetz PN_0

Tupel $PN_0 = (P, T, Pre, Post, m_0)$

MATHEMATISCHE BESCHREIBUNG DER NETZDYNAMIK

Aktualisierung des Markierungsvektors $m_k \rightarrow m_{k+1}$

$$m_{k+1} = m_k + N \delta_k$$

mit

Schaltvektor δ_k :

In der i -ten Zeile genau dann eine „1“, wenn Transition T_i im k -ten Schritt schaltet, ansonsten eine „0“. Muss natürlich der Schaltregel genügen!

Netzmatrix = Inzidenzmatrix $N = N^+ - N^-$:

für jede Stelle P_i gibt es eine Zeile

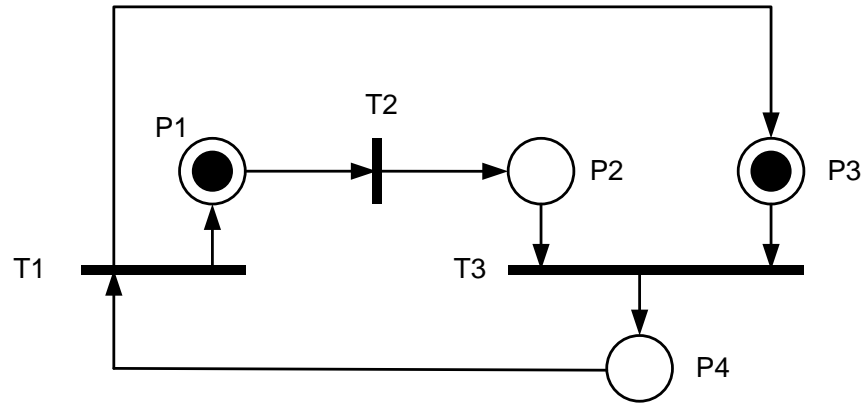
N^+ beschreibt Markenzunahme („Postkanten“)

N^- die Markenabnahme („Prekanten“)

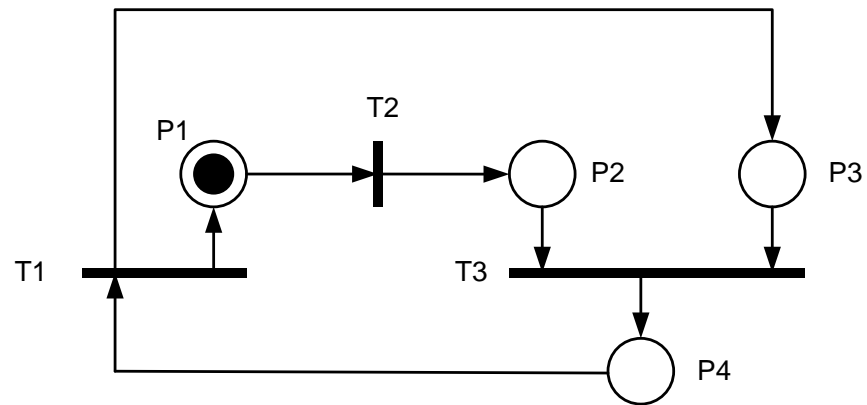
$$n_{i,j} = \begin{array}{ll} +1 & \text{wenn } (T_j, P_i) \text{ „Postkante“} \\ -1 & \text{wenn } (P_i, T_j) \text{ „Prekante“} \end{array}$$

MATHEMATISCHE BESCHREIBUNG - BEISPIEL

Fall a:



Fall b:



EIGENSCHAFTEN EINES PN: LEBENDIGKEIT, DEADLOCK

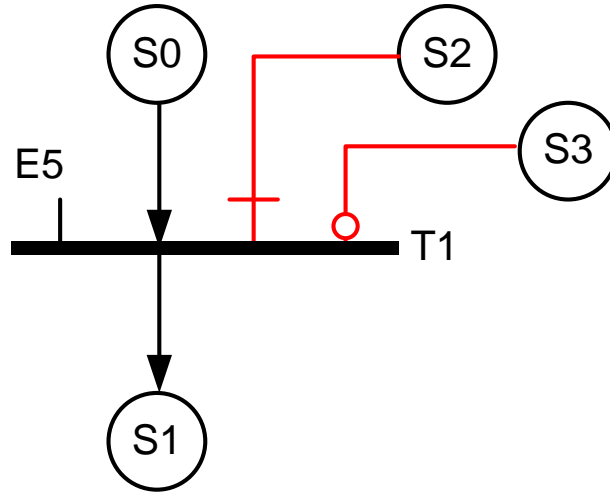
- **Tote Transition:**
wenn sie bei keiner Folgemarkierung der Anfangsmarkierung aktiviert ist
- **Deadlock = Verklemmung:**
Wenn der Markenfluss im PN zum Stillstand kommt, weil es für eine erreichte Markierung keine aktivierten Transitionen gibt.
- **Schwache Lebendigkeit:**
Wenn es zu jeder Markierung mindestens eine schaltfähige Transition gibt. Ein deadlockfreies PN ist mindestens schwach lebendig. Einzelne Transitionen dürfen „tot“ sein.
- **(Starke) Lebendigkeit:**
Jede Transition des PN kann durch eine geeignete Schaltfolge immer wieder aktiviert werden.

KAPITEL 6: PETRINETZE

- Warum Petrinetze für Steuerungsentwurf?
- Grundlagen Petrinetze
- Steuerungstechnisch Interpretierte Petrinetze (SIPN)
- Codierung von Petrinetzen
- Mathematische Beschreibung
- Einfache Erweiterungen von Petrinetzen

KOMMUNIKATIONSPLATZ – INHIBITORKANTE / TESTKANTE (1)

Stelle S2 besetzt /
Stelle S3 frei?



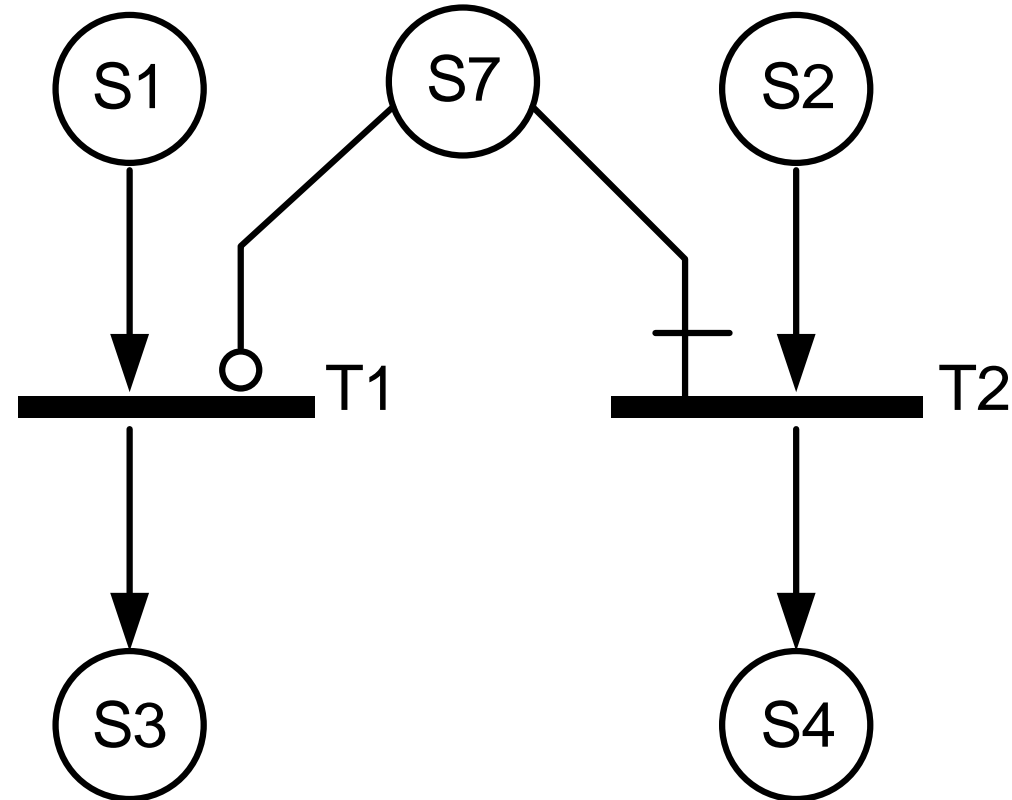
ST-Codierung:

```
IF (S0 AND NOT S1 AND E5 AND S2 AND NOT S3)
  THEN
    S0:=0;
    S1:=1;
  END_IF
```

- Stelle, die nur für eine Schaltbedingung ausgewertet (**abgefragt**) wird. Stelle markiert = „1“.
- Kante hat Querstrich (Testkante), bei Negation „Kreis“ (Negierte Kante/Inhibitorkante)!
- **Kein(!)** Zurücksetzen der Stelle nach Schalten der Transition = es werden **keine Marken von den Kommunikationsplätzen abgezogen!**

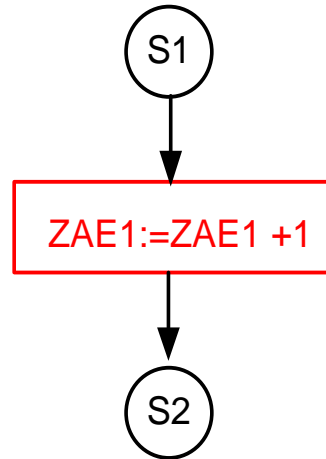
KOMMUNIKATIONSPLATZ – INHIBITORKANTE / TESTKANTE (2)

Was passiert hier?



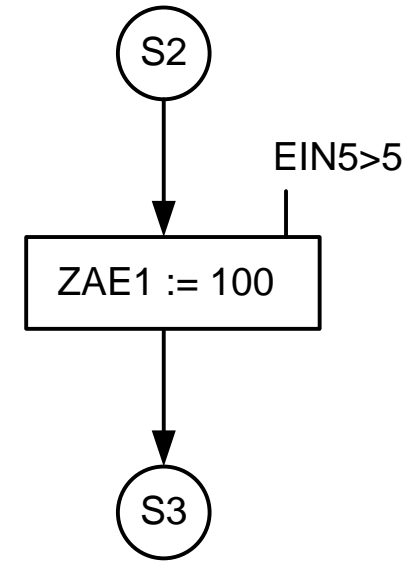
ANWEISUNGSTRANSITION

Zählerstand in ZAE1 erhöhen:



ST Codierung:

```
IF (S1 AND NOT S2)
  THEN
    S1:=0; S2:=1;
    ZAE1:=ZAE1+1;
  END_IF;
```



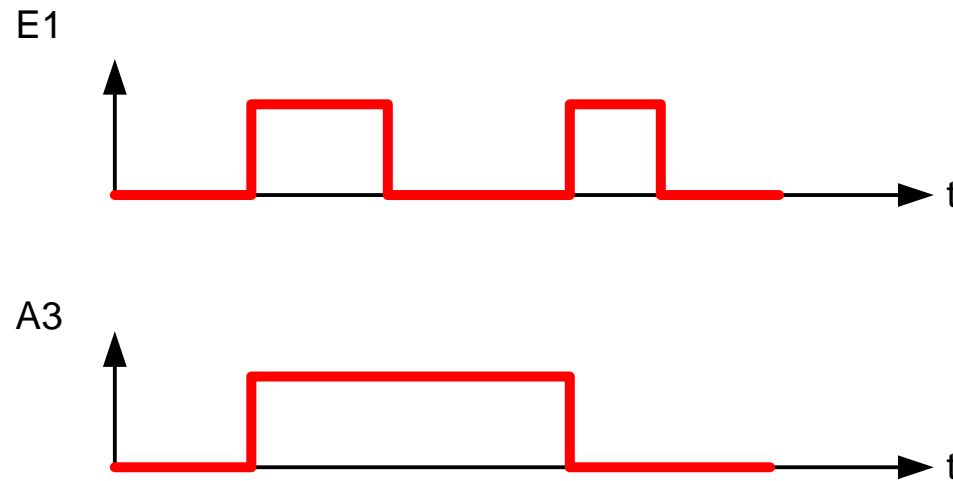
ST Codierung:

```
IF (S2 AND NOT S3 AND EIN5 > 5)
  THEN
    S2:=0; S3:=1;
    ZAE1:=100;
  END_IF;
```

Anweisungsblock wird nur **ein einziges Mal** im Moment des Schaltens abgearbeitet

BEISPIEL T-KIPPGLIED

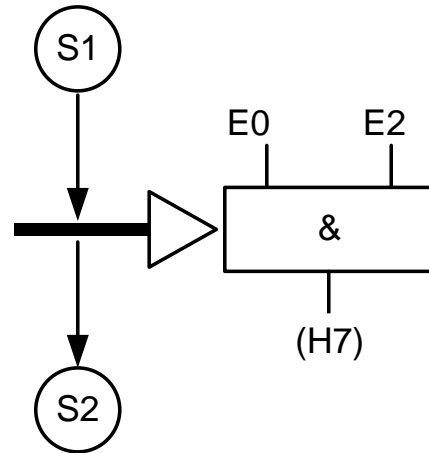
Gegeben ist folgende Zeit-Funktion:



- Entwerfen Sie ein Petrinetz, dass die Funktionalität abbildet?

TRIGGERTRANSITION

Synchronisiert einen Booleschen Ausdruck auswerten:



ST-Codierung:

```
IF (S1 AND NOT S2)
  THEN
    S1:=0; S2:=1;
    H7 := E0 AND E2;
  END_IF;
```

Nur **im Moment des Schaltens** der Transition wird der Vorgang rechts vom Pfeil **genau einmal** ausgelöst

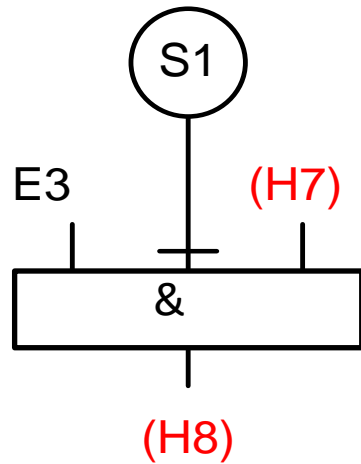
Synchronisation zwischen Transition und getriggertem Netzelement

Anwendung: Auswertung Boolescher Ausdruck // Funktionsblock. Funktion aufrufen /

Zähler erhöhen bzw. dekrementieren

HILFSMERKER

Hilfsmerker abfragen, zuweisen:



ST-Codierung:

```
H8 := S1 AND E3 AND H7;
```

Hilfsmerker können auch einen Ausgang ansteuern:

(H8) — A3

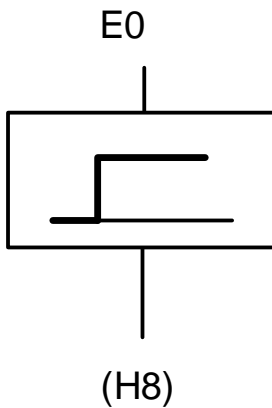
S2 — A5
A3

S7 — A3

- Hilfsmerker sind durch (...) gekennzeichnet
- Benutzung als Hilfsgröße im Steuerungsablauf, z.B. für Zwischenresultate, Berechnung boolescher Verknüpfungen, ...
- Keine Stelle im klassischen Petrinetzkonzept! **Kein Markenfluss!**

FLANKENDETEKTION

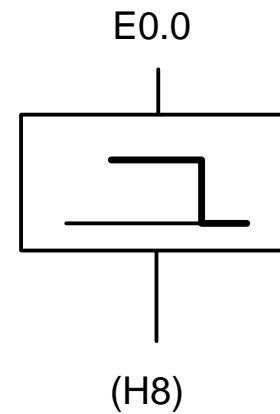
Pos. Flanke erkennen
und Hilfsmerker zuweisen:



ST Codierung:

```
H8 := E0.0 AND NOT HG;  
HG := E0.0;
```

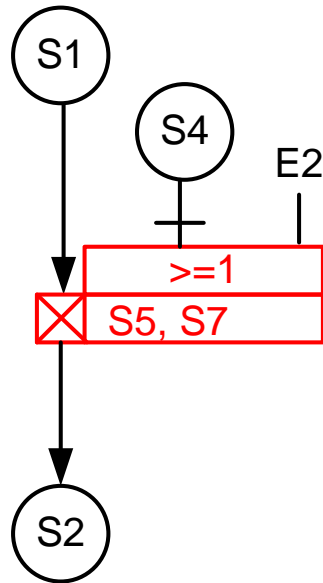
Neg. Flanke erkennen
und Hilfsmerker zuweisen:



```
H8 := NOT E0.0 AND HG;  
HG := E0.0;
```

LÖSCHTRANSITION

S5 und S7 löschen



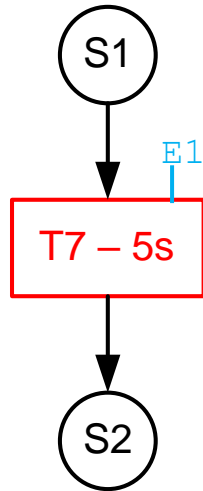
ST Codierung:

```
IF (S1 AND NOT S2 AND (S4 OR E2))  
  THEN  
    S1:=0; S2:=1;  
    S5:=0; S7:=0;  
  END_IF
```

- Spezielle Transition – gekennzeichnet durch ein „X“
- Zweck: Entfernen von Marken aus dem Petrinetz
- Beispiel: Komplettes Zurücksetzen eines Petrinetzes

ZEITTRANSITION

Stelle S1 soll 5s lang besetzt sein,
bevor weitergeschaltet wird:



ST Codierung:

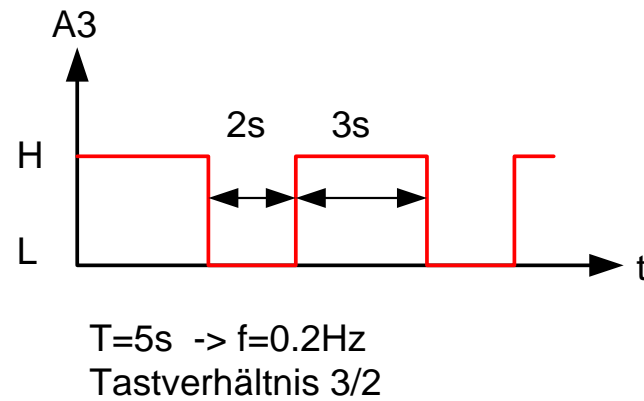
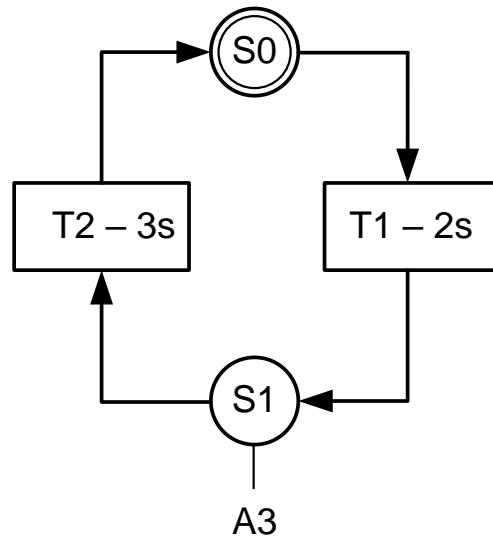
```
IF T7.Q
    THEN S1:=0; S2:=1;
END_IF
T7(IN:=S1 AND NOT S2 AND E1, PT:=T#5s);
```

T7 ist Instanz von TON!

- Die Schaltbedingung muss **während der gesamten eingestellten Zeit gültig** sein, damit die Transition durchschaltet (TON-Verhalten)
- Ist Schaltbedingung irgendwann während der eingestellten Zeit ungültig, läuft die Überwachungszeit der Transition neu an
- Beispiel: Pumpe muss 5 Sekunden laufen

ZEITTRANSITION – BEISPIEL „TAKTGENERATOR“

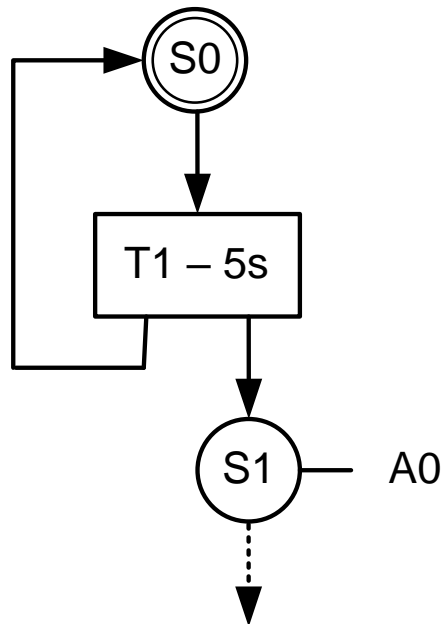
Taktgenerator mit Tastverhältnis 3:5



ZEITTRANSITION – BEISPIEL MARKENGENERATOR

Markengenerator:

„alle 5 s eine Marke ins PN geben“



ST Codierung:

```
IF T1.Q
    THEN S1:=1;
END_IF
T1(IN:=S0 AND NOT S1 AND NOT T1.Q,
    PT:=T#5s);
```

Achtung:

Nur bei einer Zeittransition
ist eine Rückführung erlaubt!

ZEITTRANSITION - BEISPIEL „ZYKLISCHER RESSOURCENTAUSCH“

Aufgabe:

- (1) Drei redundante Pumpen (Pumpe1, Pumpe2, Pumpe3) sollen über den Tag verteilt jeweils **4h** nacheinander die Pumpleistung für eine Anlage erbringen. Erstellen Sie dafür das Petrinetz unter **Benutzung eines Markengenerators!**
- (2) Erweitern Sie das Petrinetz aus Aufgabenteil a) derart, dass auch folgender Fall berücksichtigt ist:
„Falls die Pumpe i ($i=1,2$ oder 3) gestört ist ($STOER_i=1$) oder in Reparatur ($REP_i=1$) ist, wird die nächste funktionsfähige Pumpe benutzt.“