

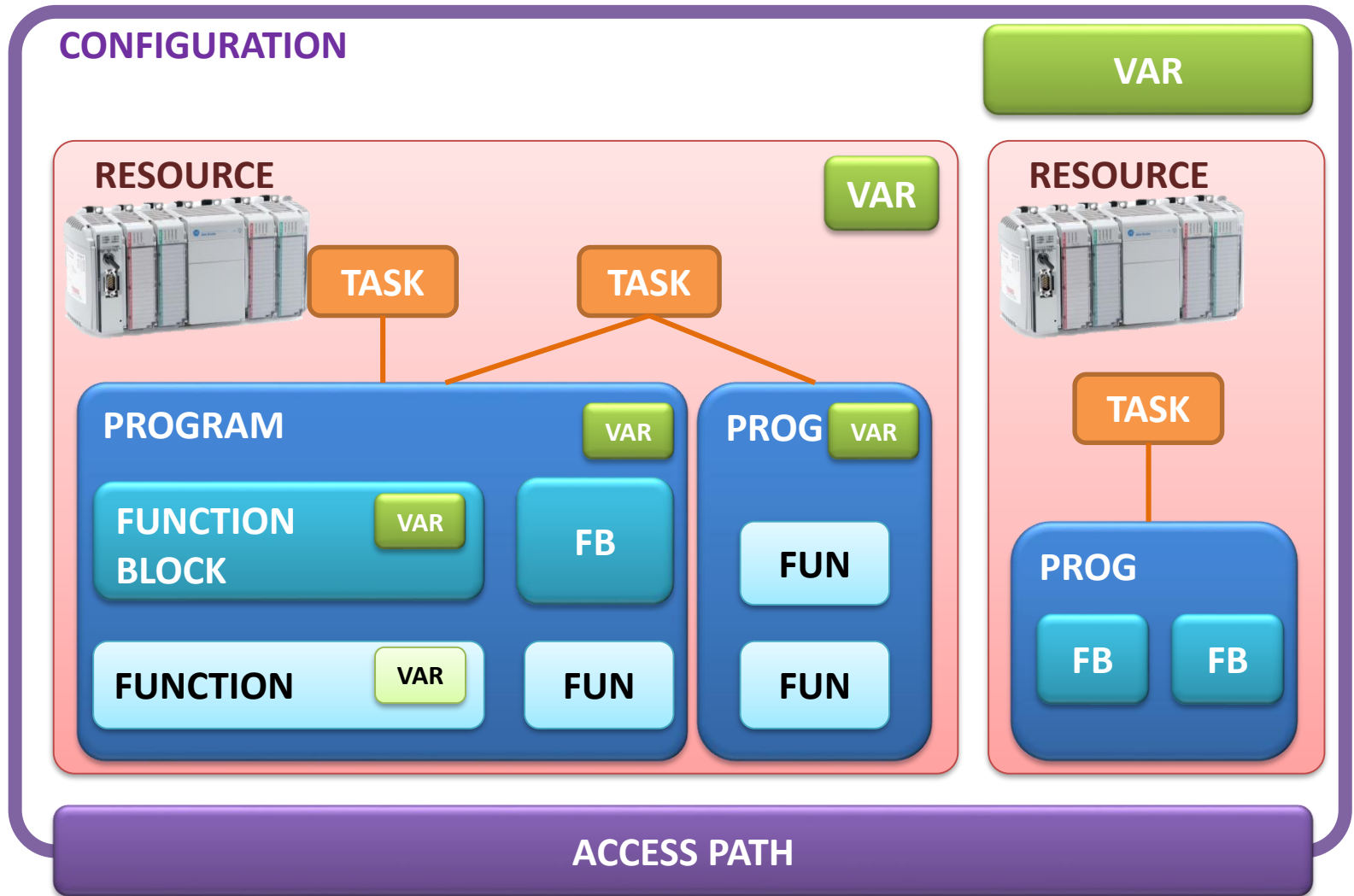
# Az IEC 61131-3 szabvány programozási nyelvei

## Sorrendi folyamatábra

Programozható irányítóberendezések  
és szenzorrendszerek

KOVÁCS Gábor  
gkovacs@iit.bme.hu

# Áttekintés



# Programszervezési egységek

## POU típus és név

### Deklarációs rész:

- Interfész változók
- Lokális változók
- Globális változók

### POU törzs: programkód

- Ladder Diagram (LD)
- Instruction List (IL)
- Function Block Diagram (FBD)
- Structured Text (ST)
- **Sequential Function Chart (SFC)**

**PROGRAM** prog\_name

PROGRAM ConveyorControl

**FUNCTION\_BLOCK** fb\_name

FUNCTION\_BLOCK Pusher

~~FUNCTION fun\_name : DataType~~  
~~FUNCTION IsReady : BOOL~~

# Sorrendi folyamatábra

## *(Sequential Function Chart, SFC)*

- Az SFC a programfolyamot írja le
- Tárolnia kell az aktuális állapotot: függvény nem valósítható meg SFC-ben
- A kimeneti változók érvényessége sokszor nehezen értelmezhető: egyes fejlesztői környezetek FB megvalósítását sem engedik SFC-ben

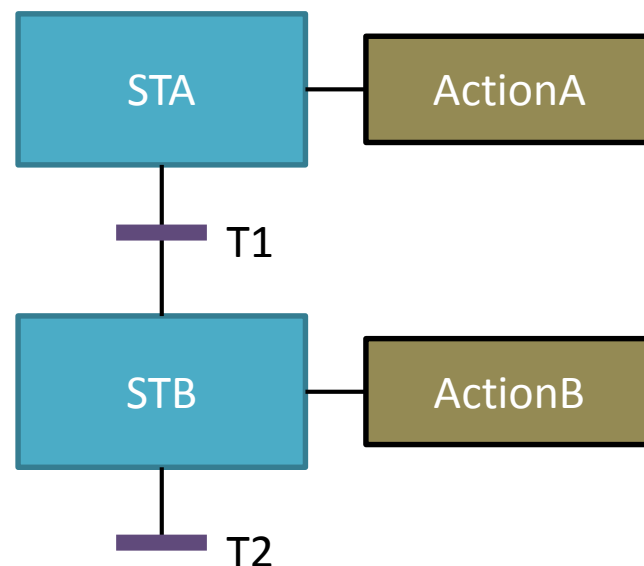
# Sorrendi folyamatábra

## *(Sequential Function Chart, SFC)*

- Cél: komplex programok kisebb részekre bontása és az azok közötti programfolyam leírása
- Eredet
  - Folyamatábra
  - Petri-hálók
  - Grafcet

# SFC elemek

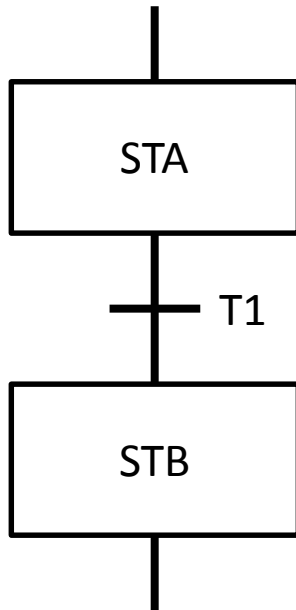
- SFC: páros gráf
  - Csomópontok:
    - Lépések (*step*)  $\approx$  állapotok
    - Átmenetek (*transition*) :  
logikai értékre kiértékelődő  
kifejezések
  - Élek
- Lépésekhez rendelt akciók  
(*action*)



# Lépés – Átmenet szekvenciák

- Egyszerű szekvencia
- Divergens utak (elágazás)
  - Szekvencia-hurok
  - Szekvencia átugrása
- Párhuzamos végrehajtás

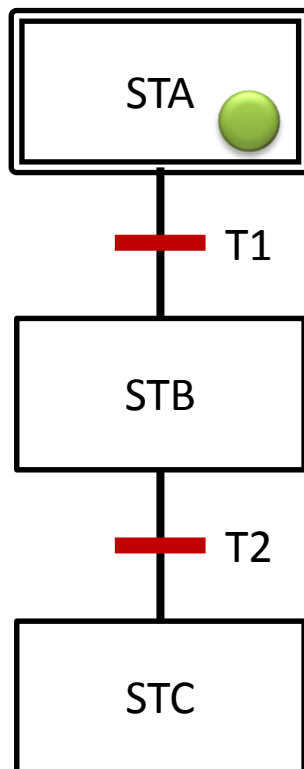
# Egyszerű szekvencia



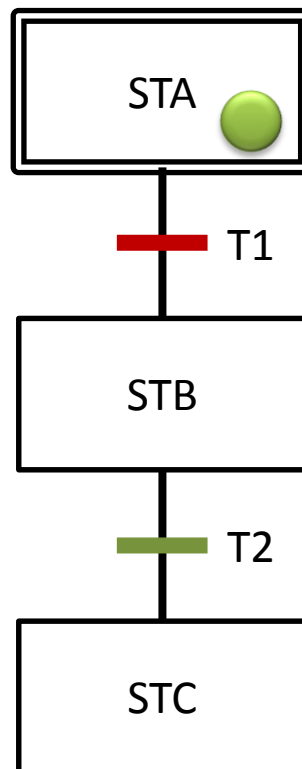
- Amikor T1 igazra értékelődik ki
  - STA deaktiválódik
  - STB aktiválódik



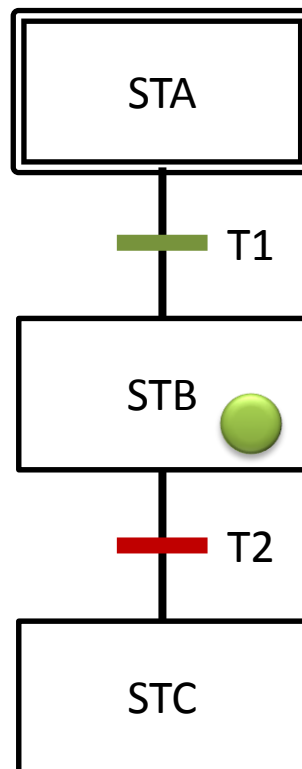
# Egyszerű szekvencia



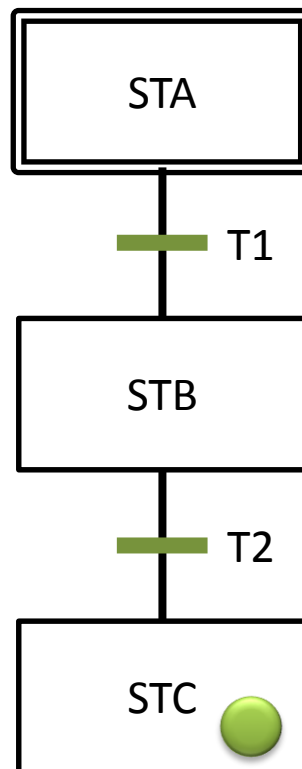
# Egyszerű szekvencia



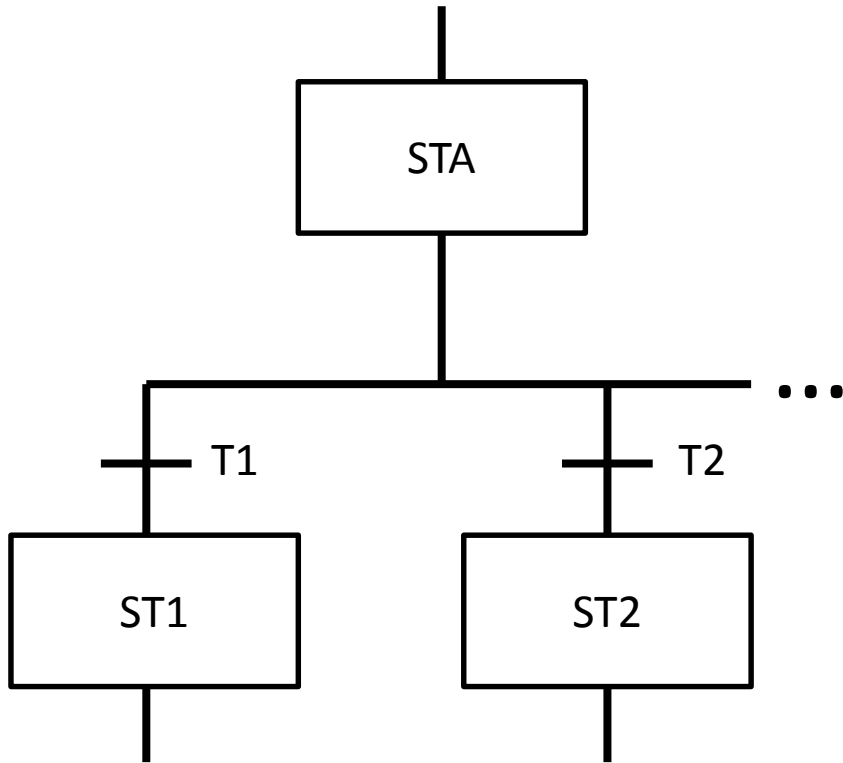
# Egyszerű szekvencia



# Egyszerű szekvencia



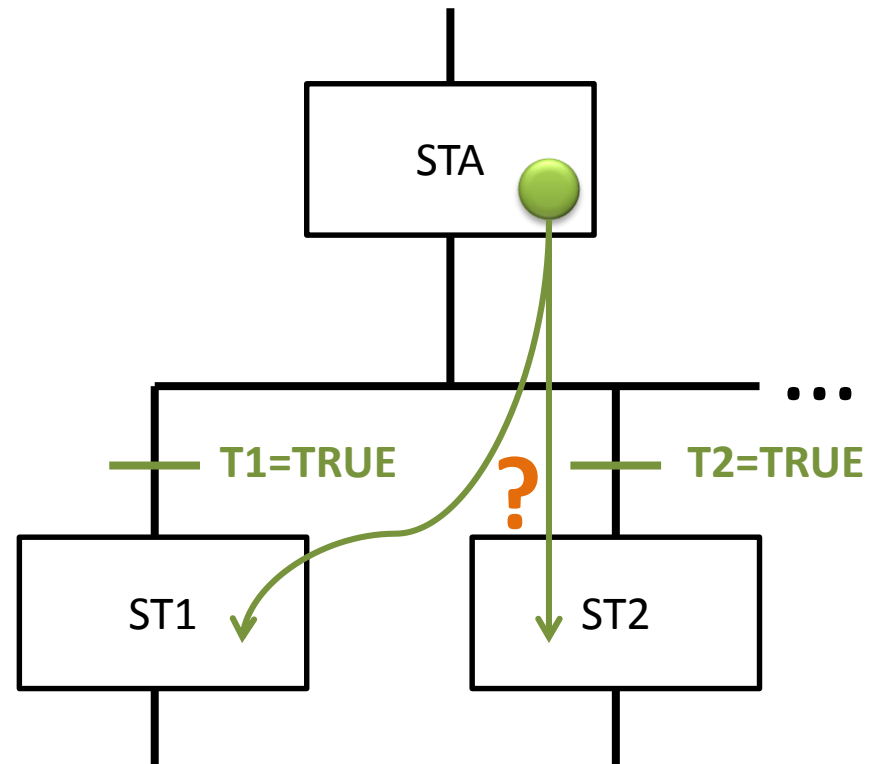
# Divergens utak



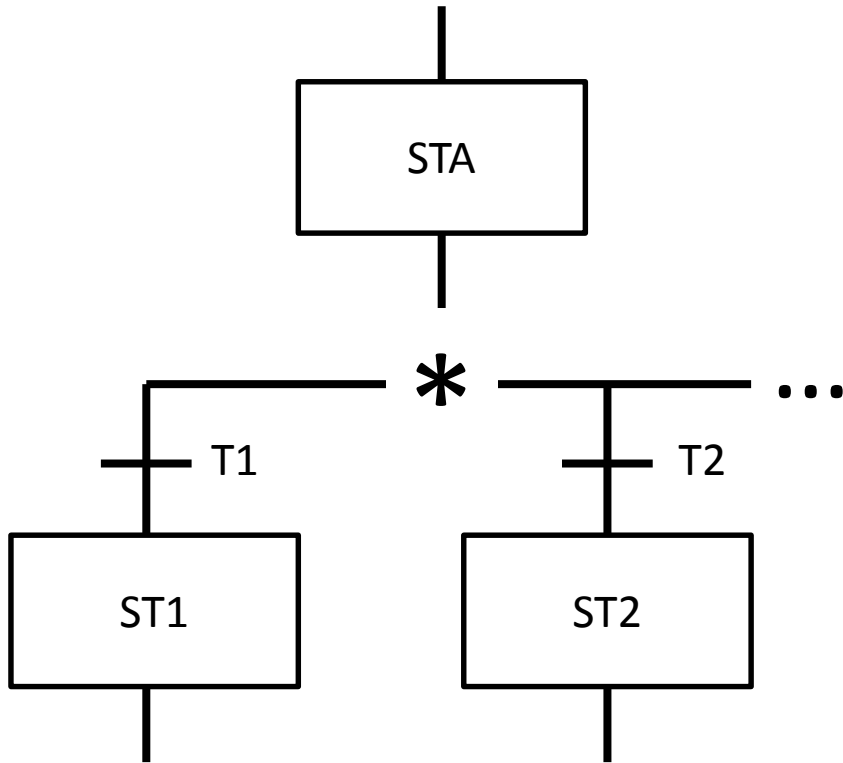
- Egy lépés után több átmenet is következik
- Az első igazra kiértékelődő  $T_i$  átmenet deaktiválja STA-t és aktiválja  $ST_i$ -t
- Az ágak közül csak egy lesz aktív

# Divergens utak átmeneteinek kiértékelése

- Szabvány szerint
  - Kiértékelés balról jobbra
  - Kiértékelés explicit prioritással
  - Kiértékelés felhasználói irányítással
- Általános gyakorlati megvalósítás:  
kiértékelés balról jobbra

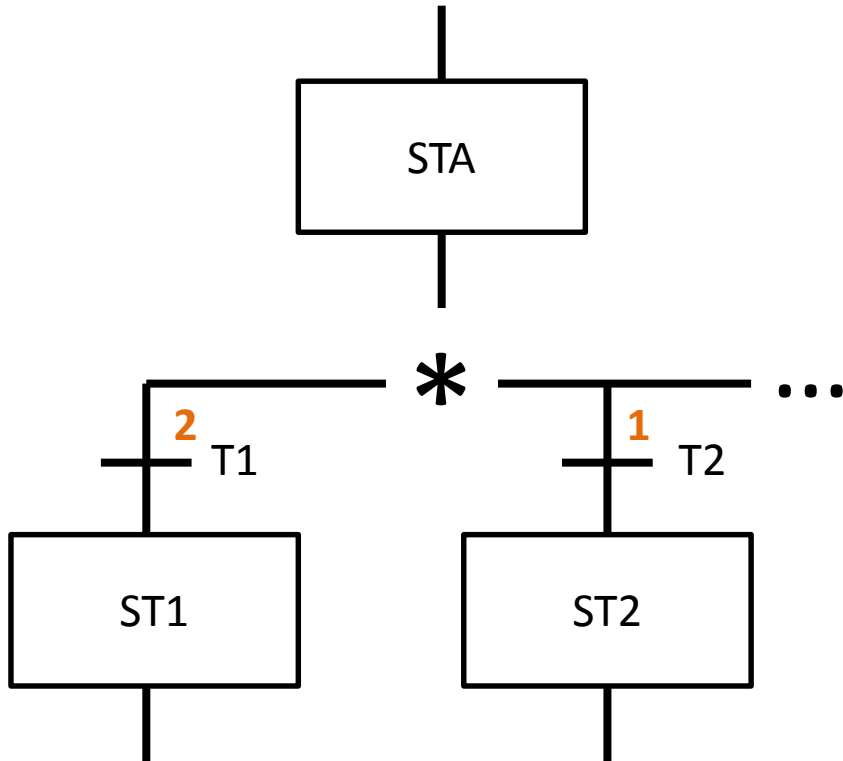


# Kiértékelés balról jobbra



- Amíg STA aktív, addig az átmeneteket folyamatosan kiértékeljük balról jobbra haladva
- Az első igazra kiértékelődő  $T_i$  átmenet deaktiválja STA-t és aktiválja  $ST_i$ -t

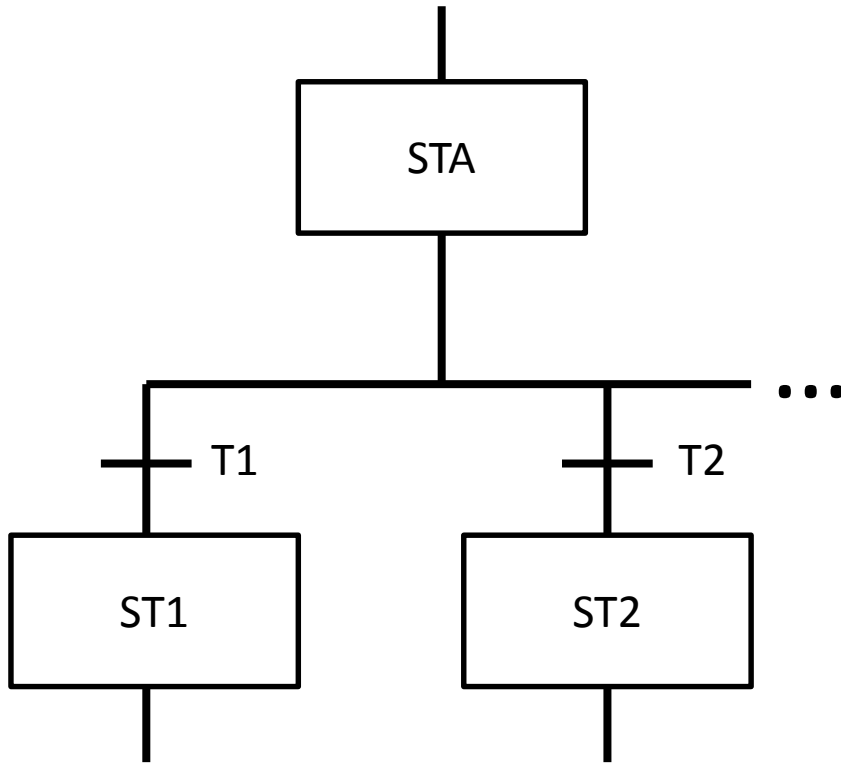
# Kiértékelés felhasználó által megadott prioritással



- Az átmeneteket a megadott sorrendben értékeljük ki
- Az alacsonyabb érték jelzi a magasabb prioritást



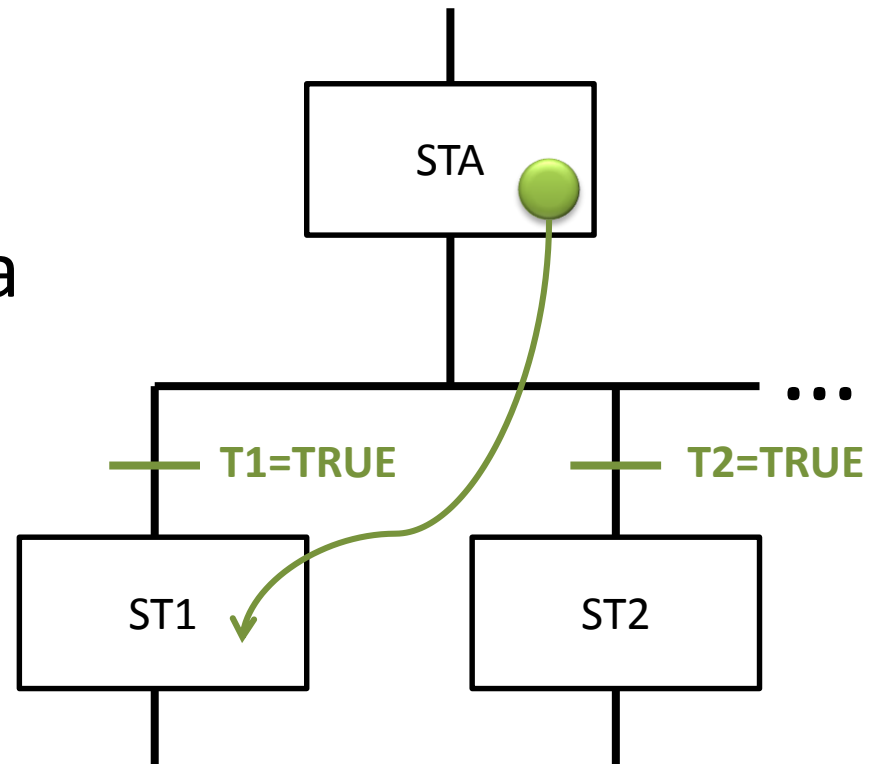
# Kiértékelés felhasználói irányítással



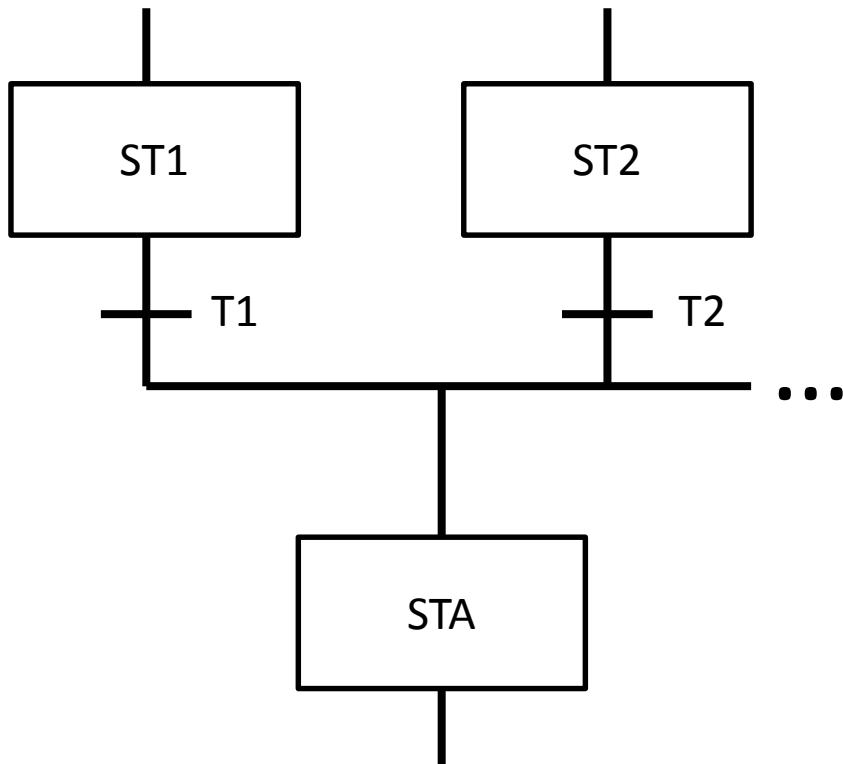
- Az átmenetek kiértékelésének sorrendje nem meghatározott
- A felhasználónak kell biztosítania, hogy a feltételek kölcsönösen kizáróak legyenek

# Divergens utak - megjegyzés

- A legtöbb fejlesztői környezet csak a standard (balról jobbra haladó) kiértékelést támogatja
- Ebben az esetben a jelölésből kimarad a \*

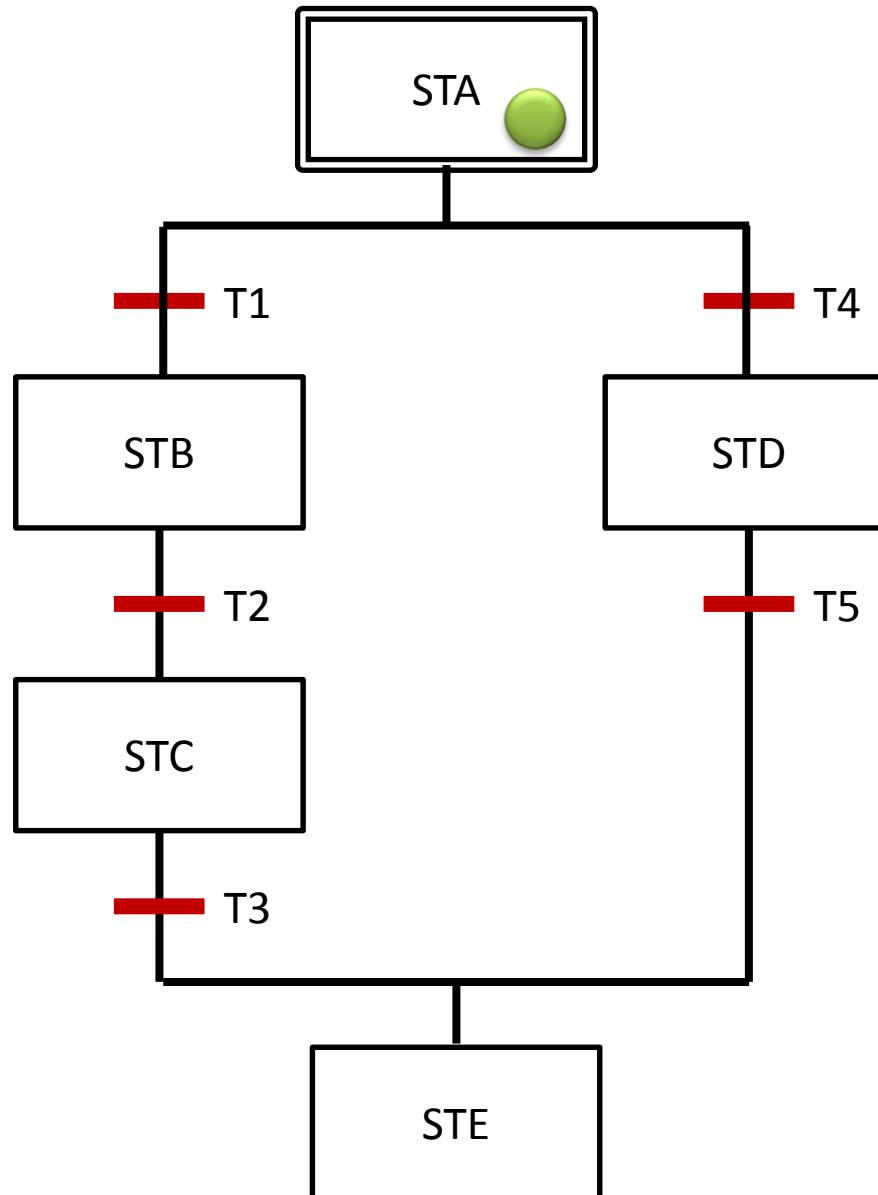


# Divergens utak találkozása

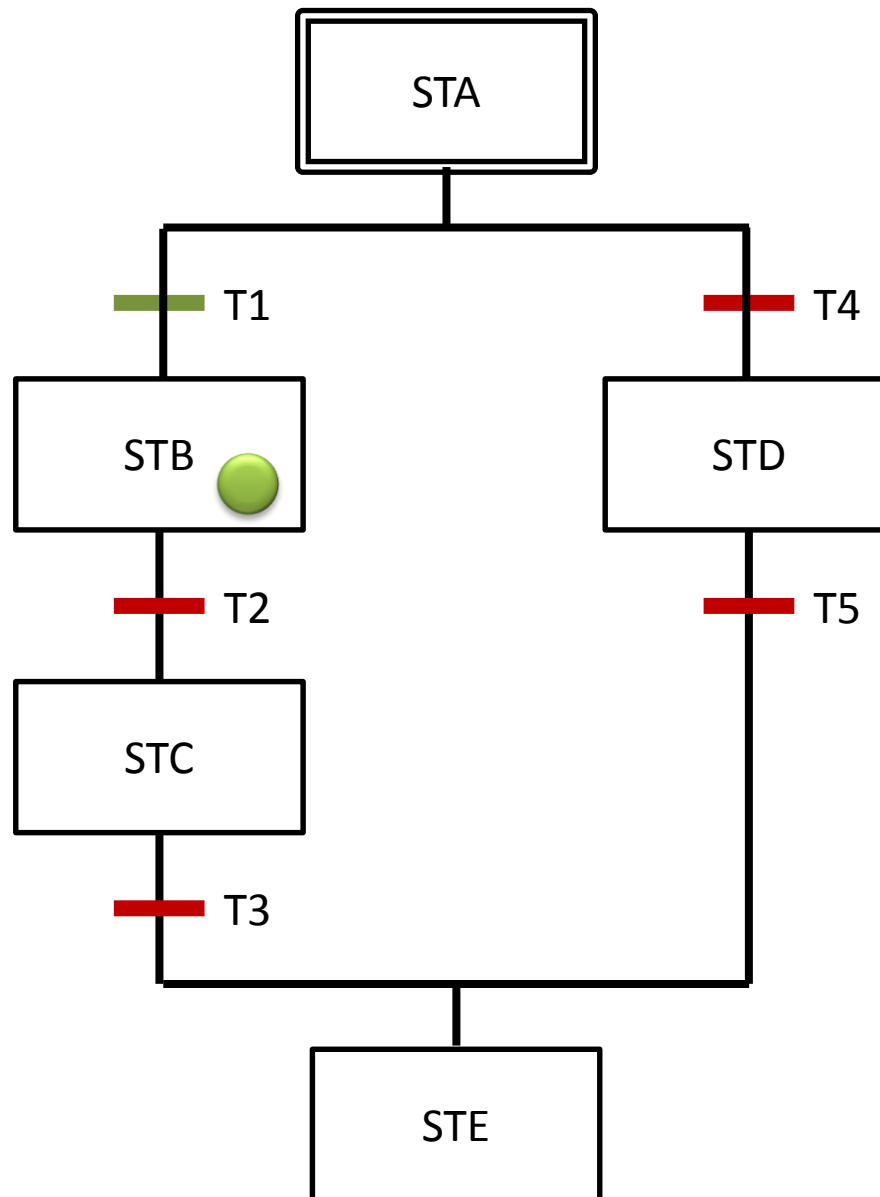


- STA-t akkor aktiváljuk, ha  $ST_i$  aktív és  $T_i$  igazra értékelődik ki
- Ugyanekkor  $ST_i$ -t deaktiváljuk

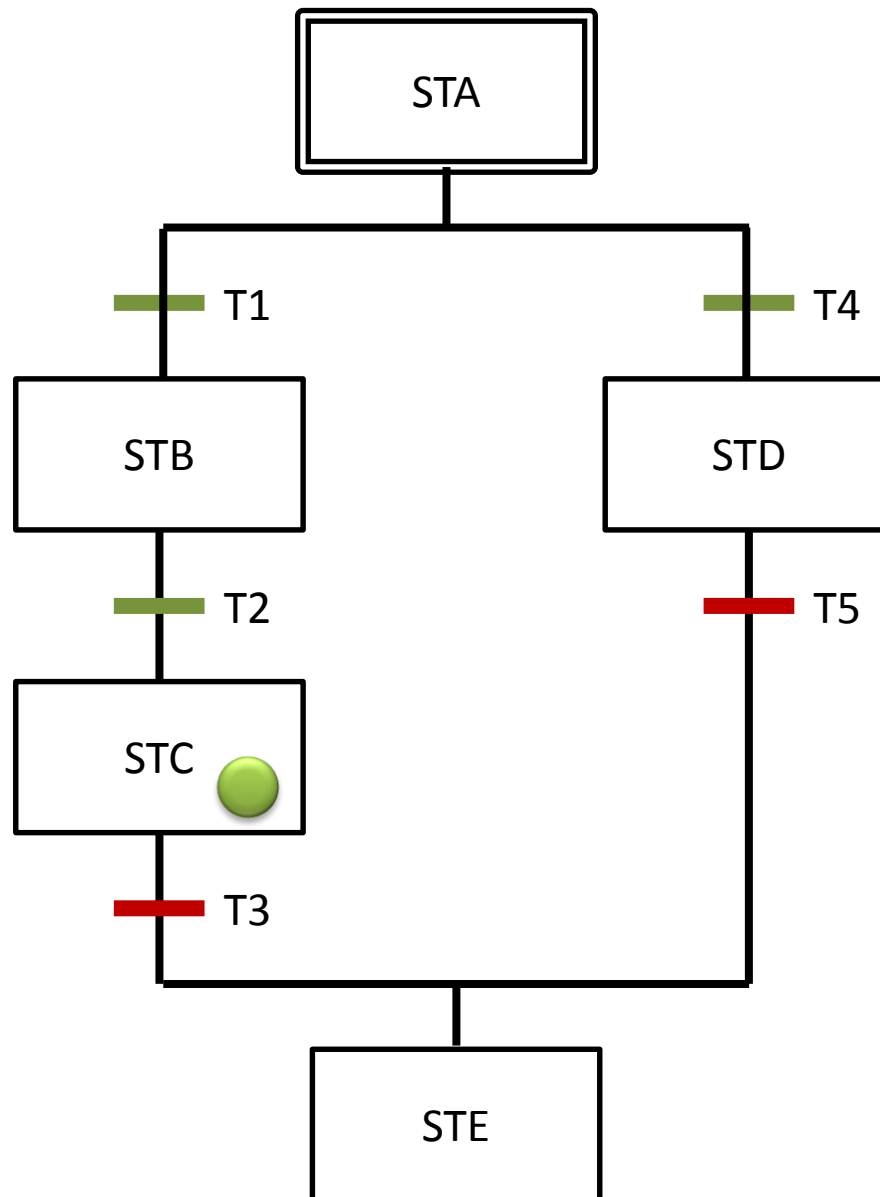
# Divergens utak



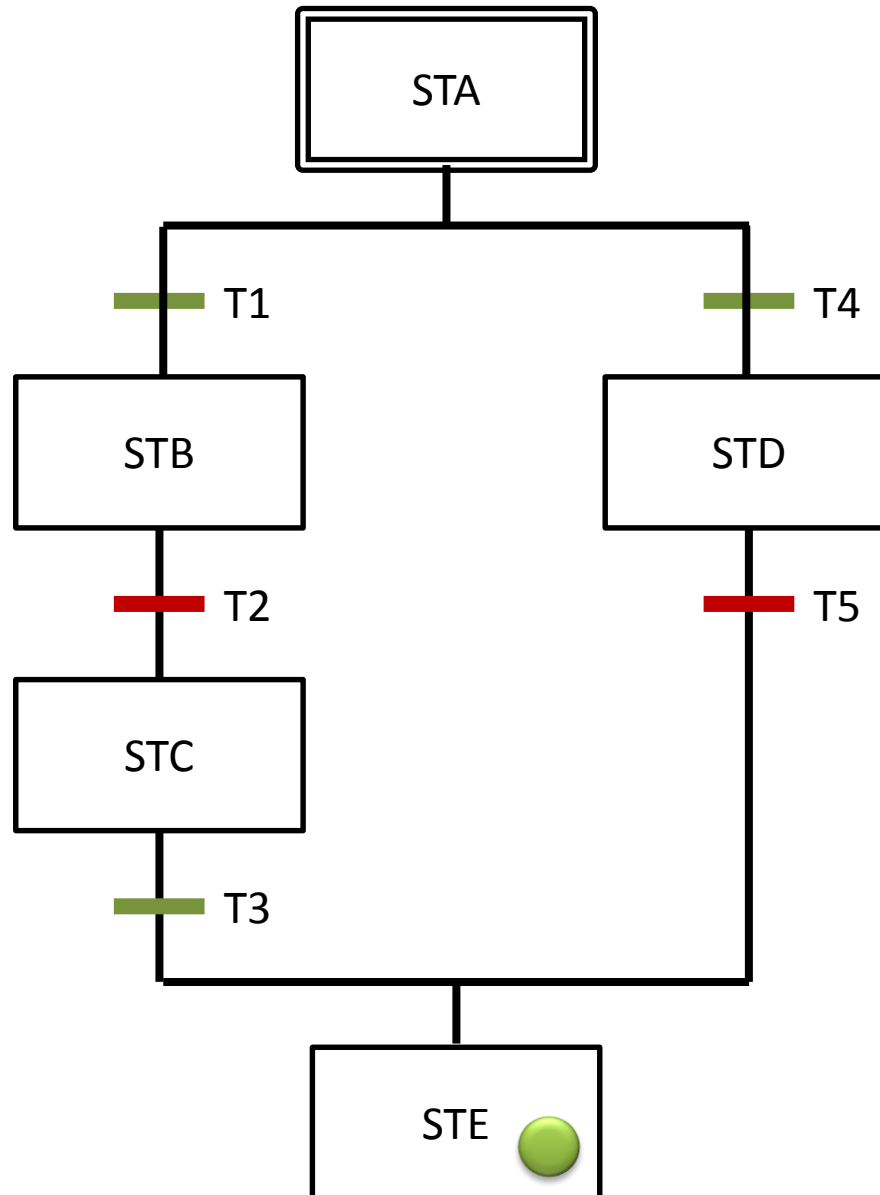
# Divergens utak



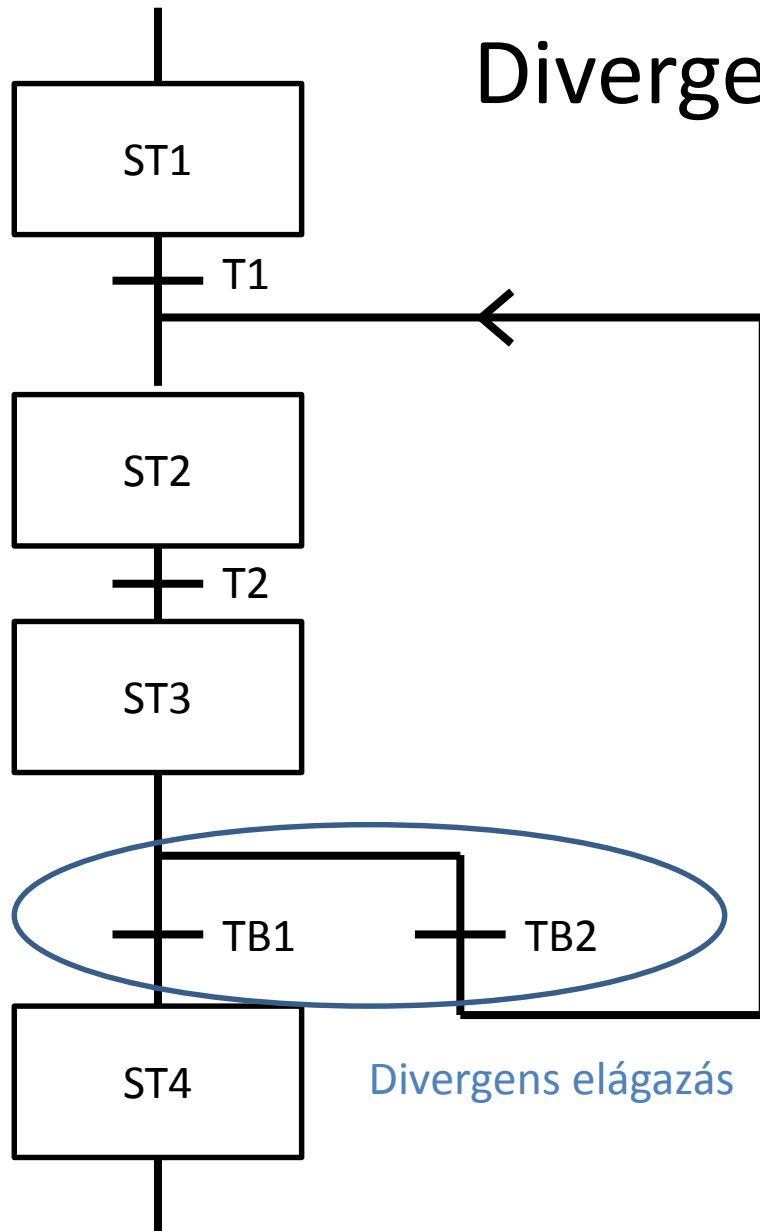
# Divergens utak



# Divergens utak



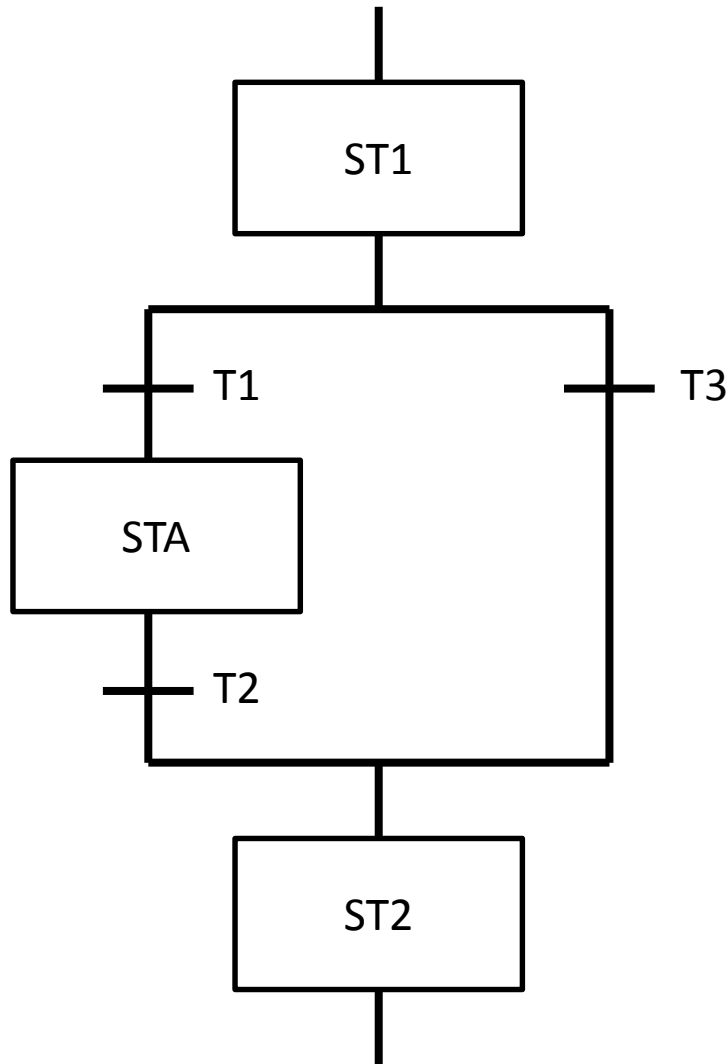
# Divergens utak: szekvencia-hurok



- Előd-lépéshez visszakanyarodó divergens út
- Tetszőleges prioritás-modell használható

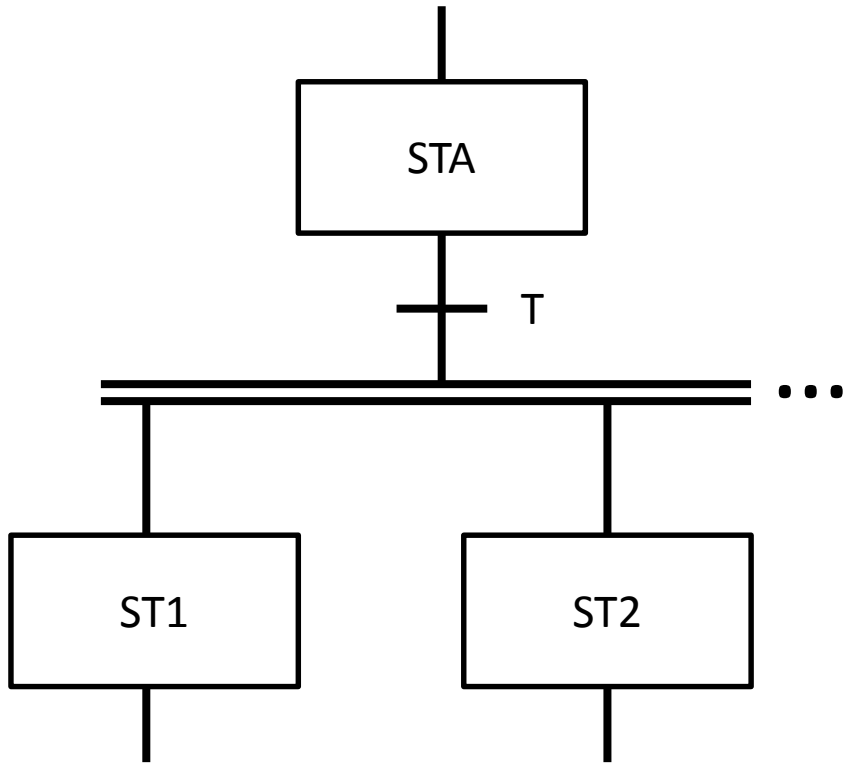


# Divergens utak – szekvencia átugrása



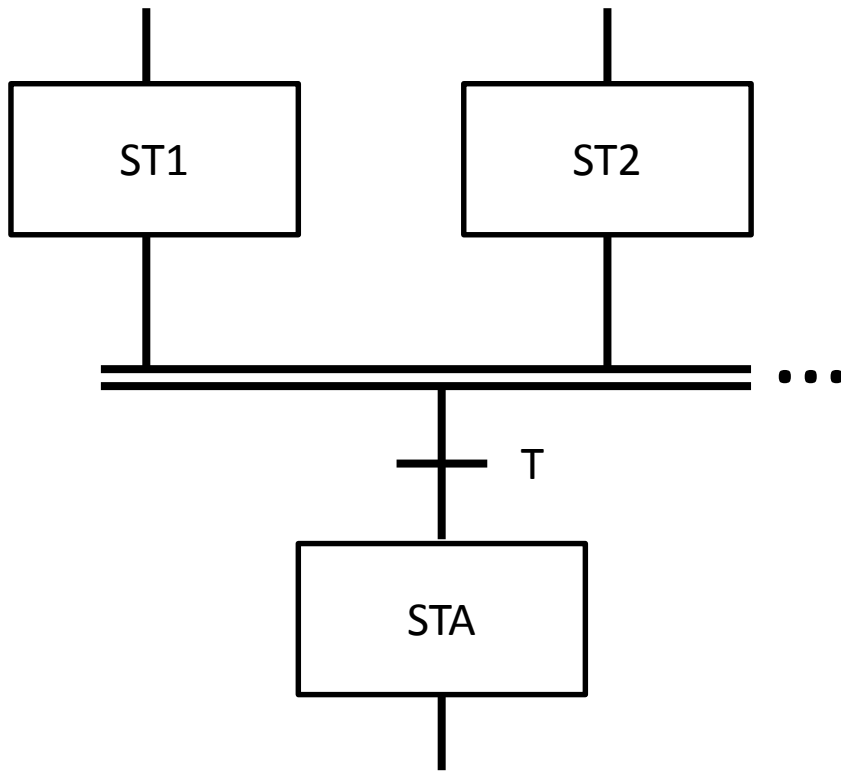
- Az STA lépést átugorjuk, ha T3 igazra értékelődik ki (miközben T1 hamis)

# Párhuzamos végrehajtás



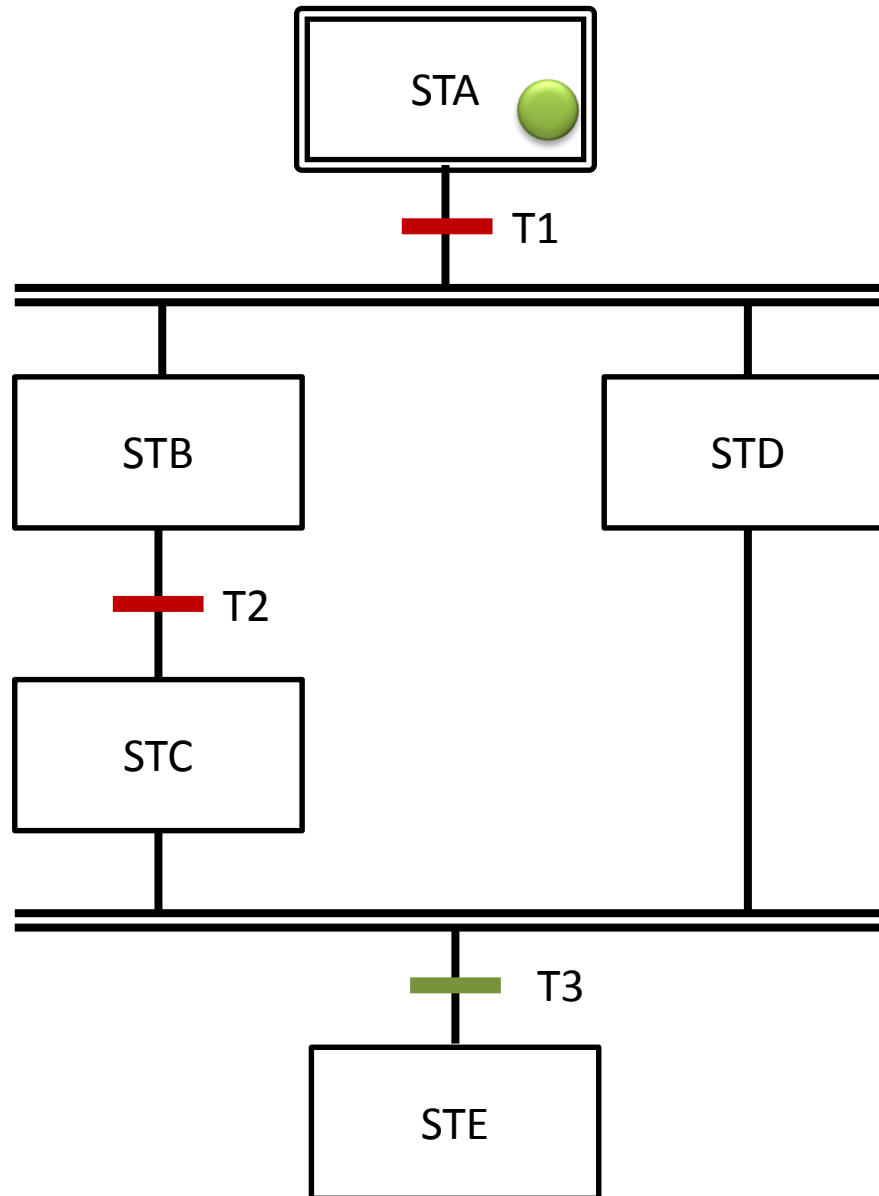
- Ha STA aktív és T igazra értékelődik ki, akkor STA-t deaktiváljuk és az összes  $ST_i$ -t aktiváljuk
- A token „osztódik”

# Párhuzamos szekvenciák találkozása

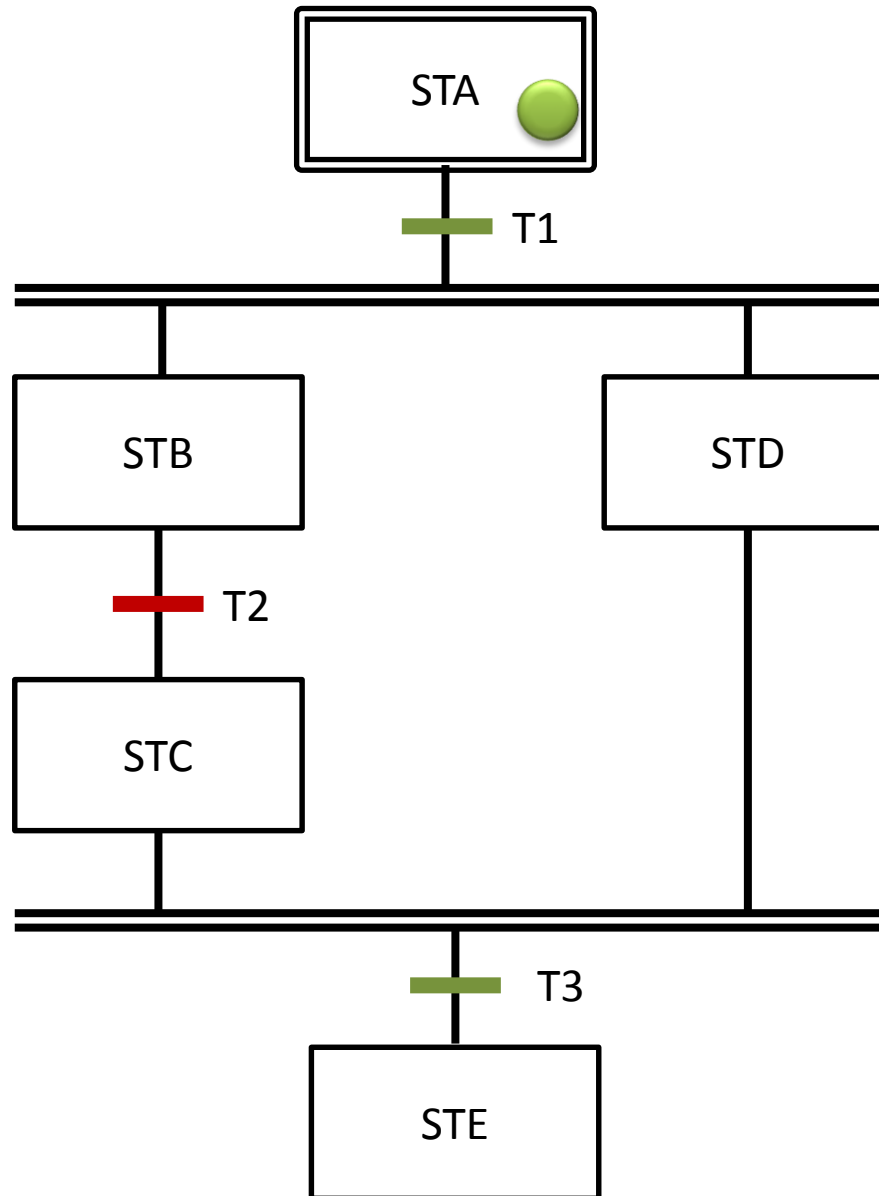


- STA-t akkor aktiváljuk, ha
  - T igazra értékelődik ki
- ÉS
- Az összes  $ST1...STn$  lépés aktív
- Ekkor az „osztódott” tokenek egyesülnek

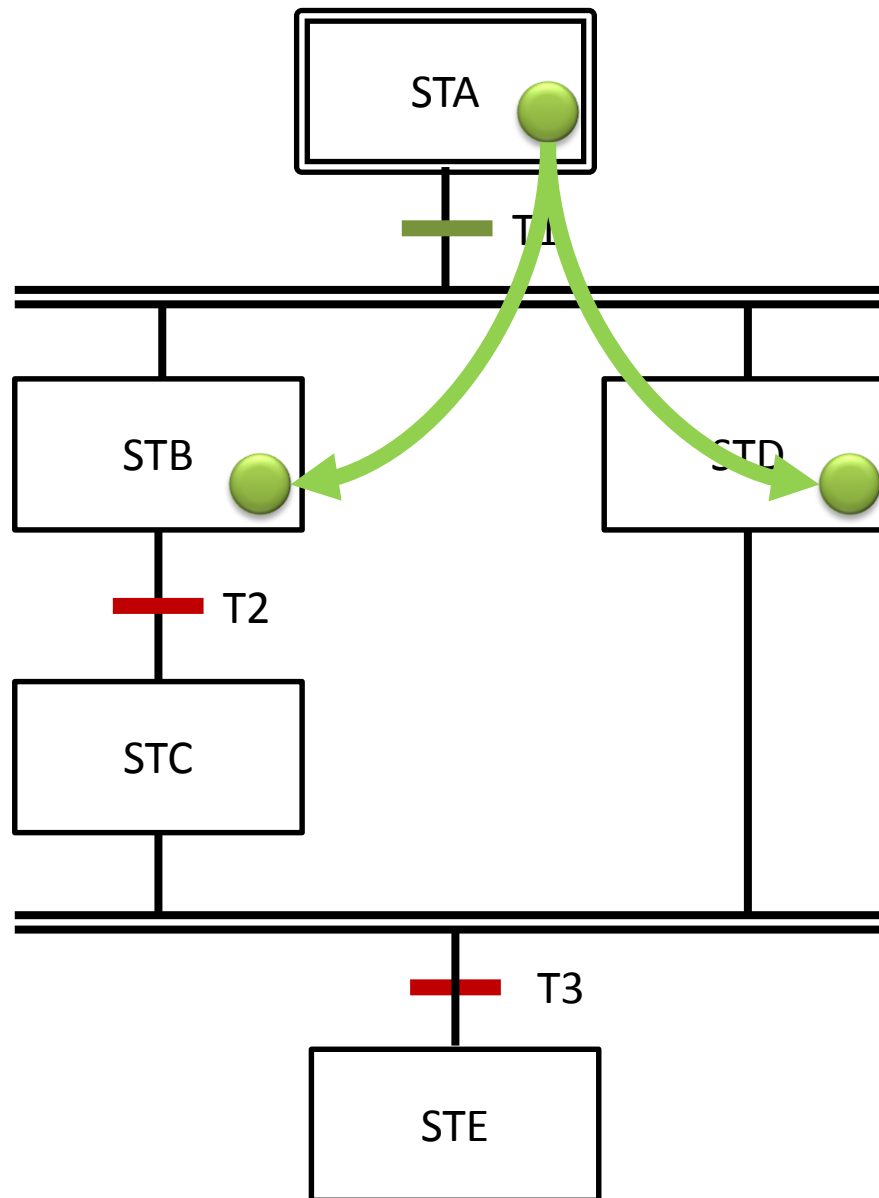
# Párhuzamos végrehajtás



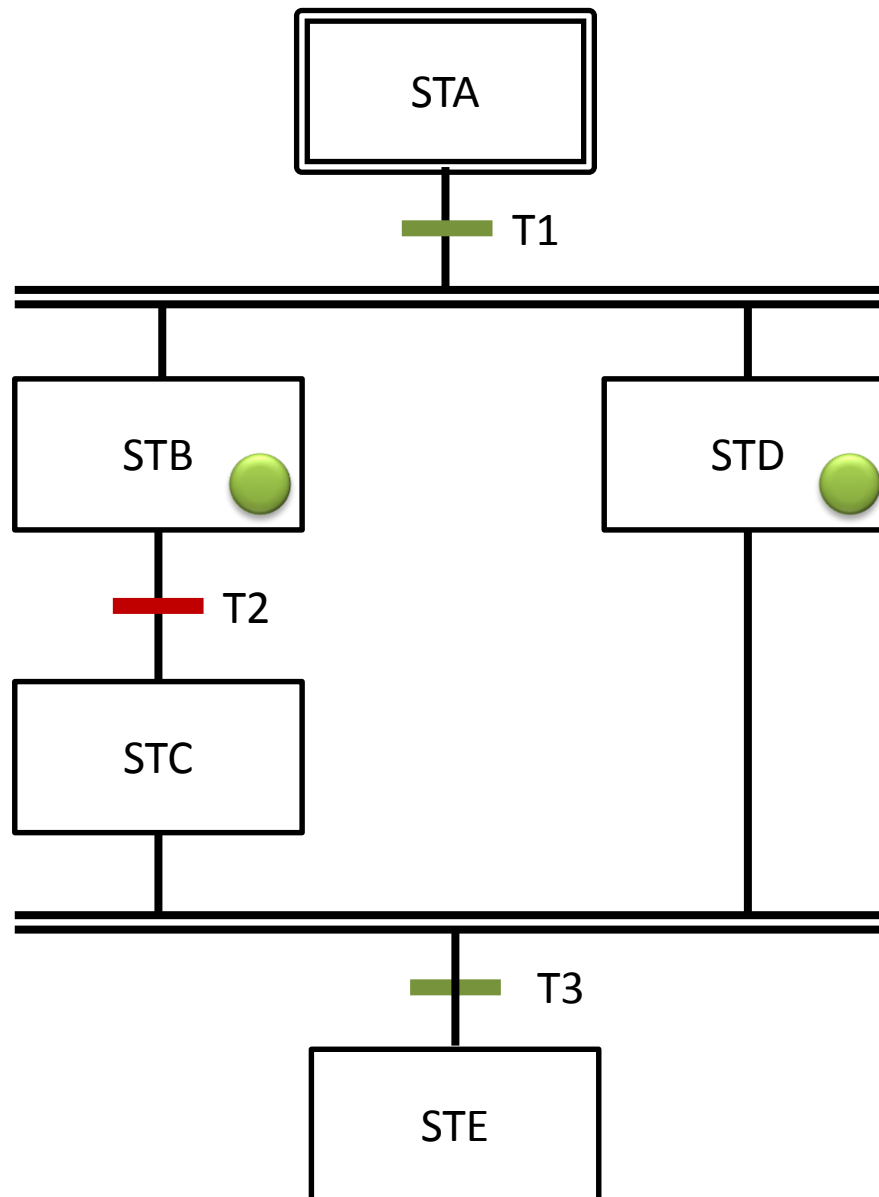
# Párhuzamos végrehajtás



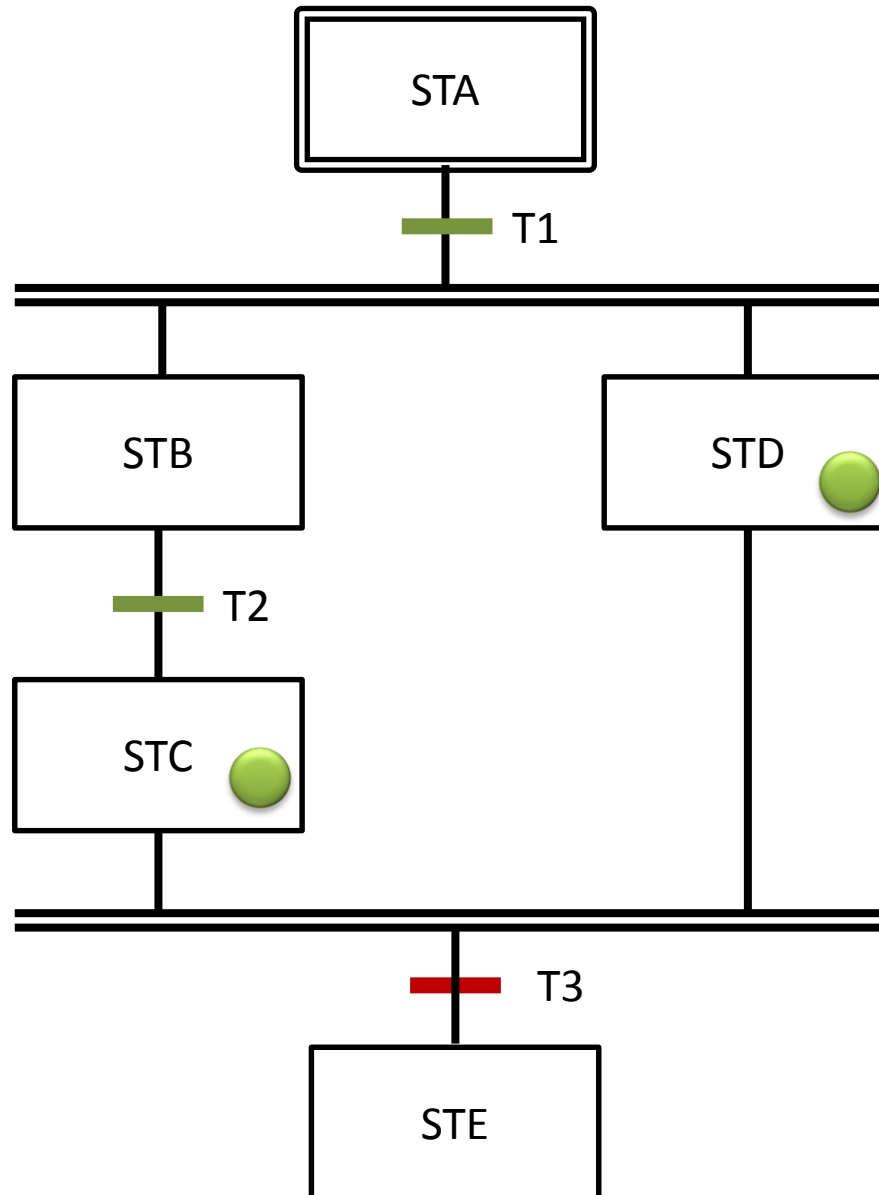
# Párhuzamos végrehajtás



# Párhuzamos végrehajtás

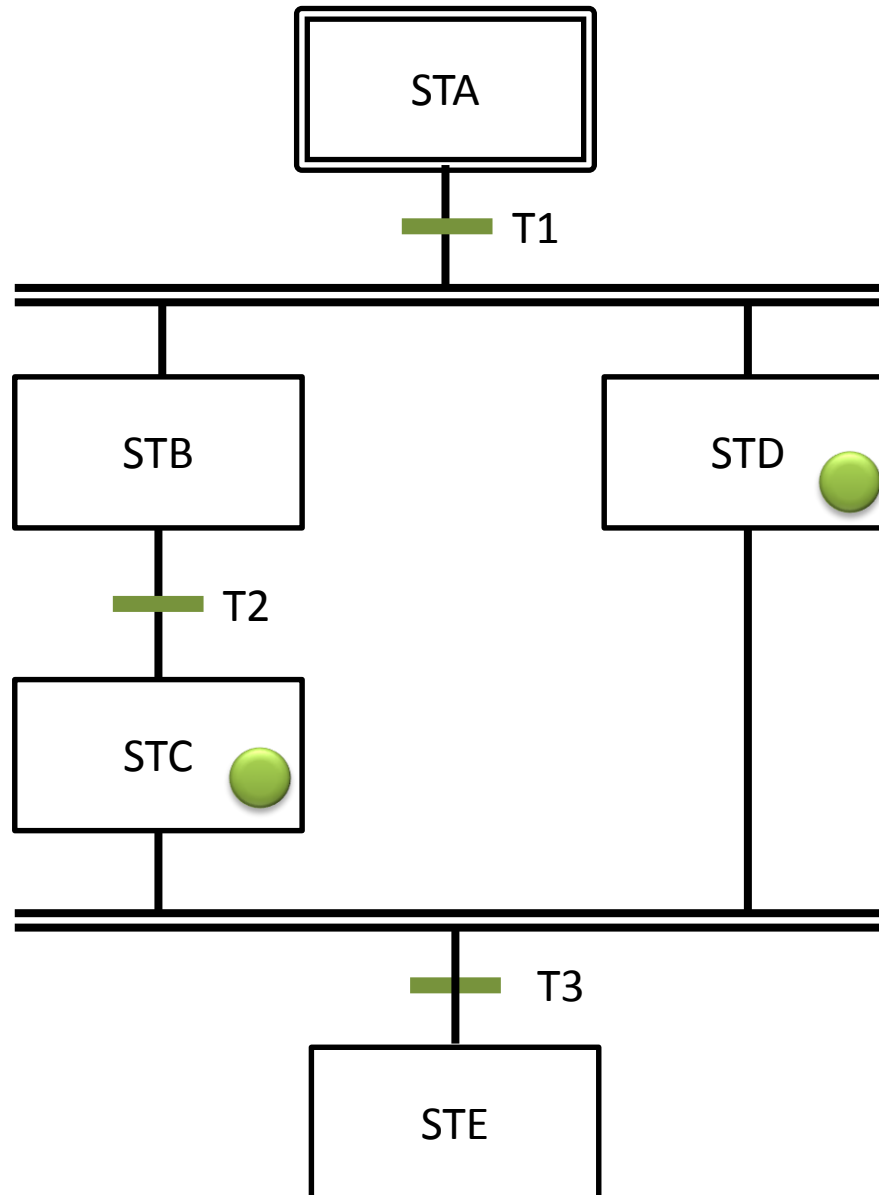


# Párhuzamos végrehajtás

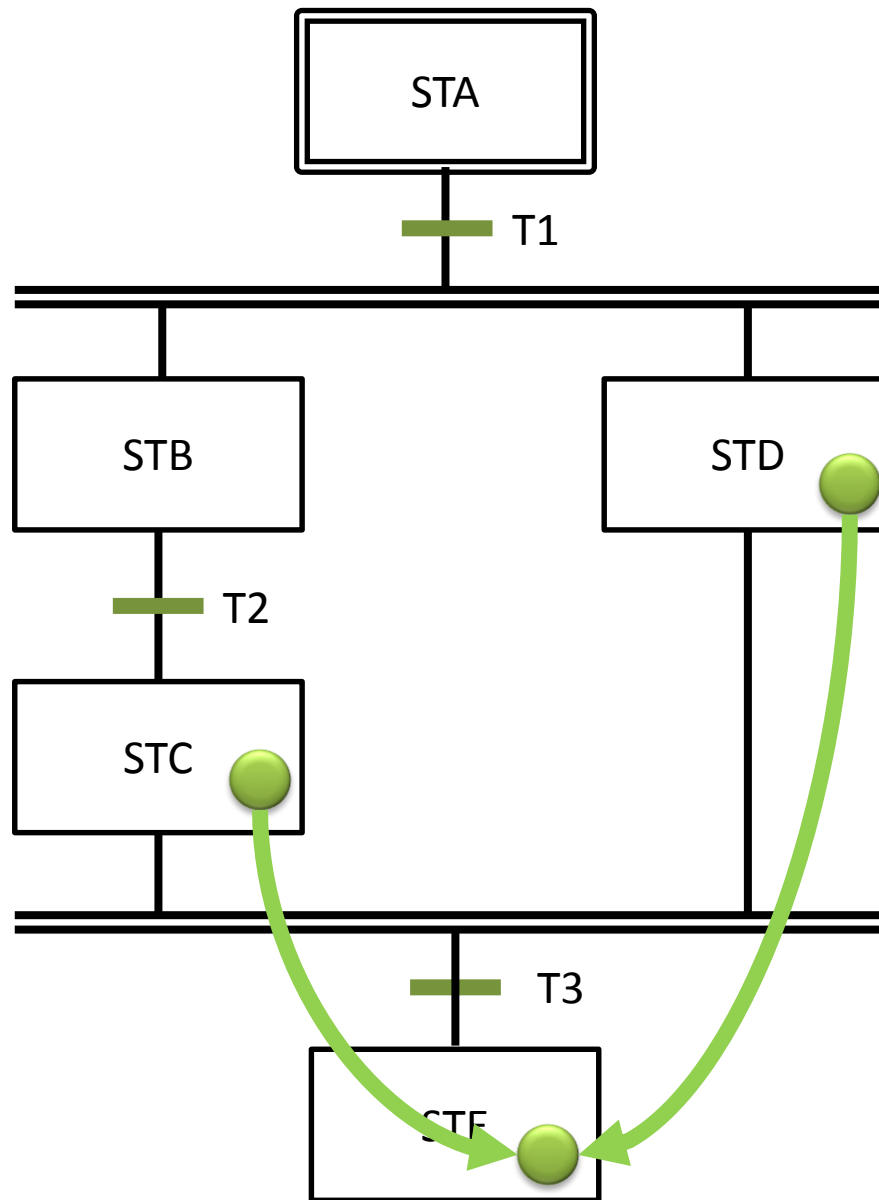




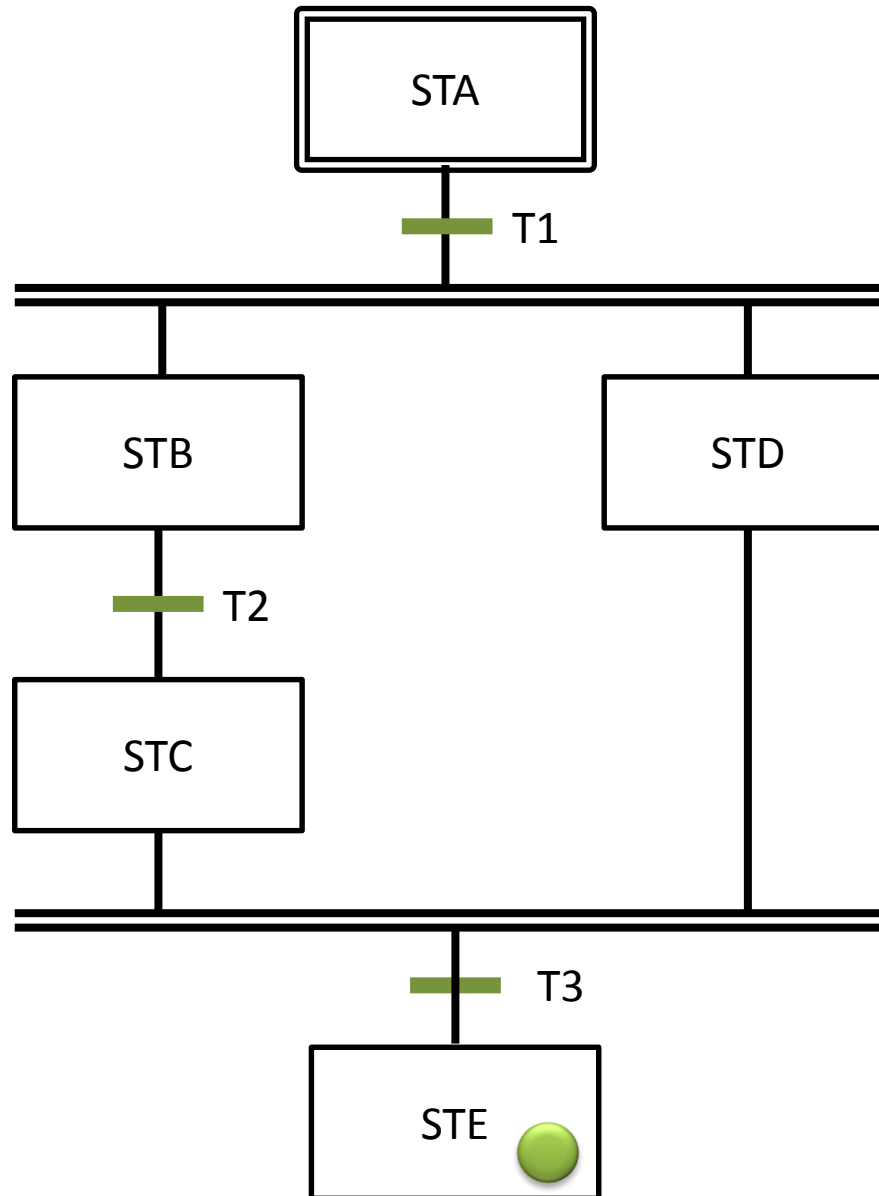
# Párhuzamos végrehajtás



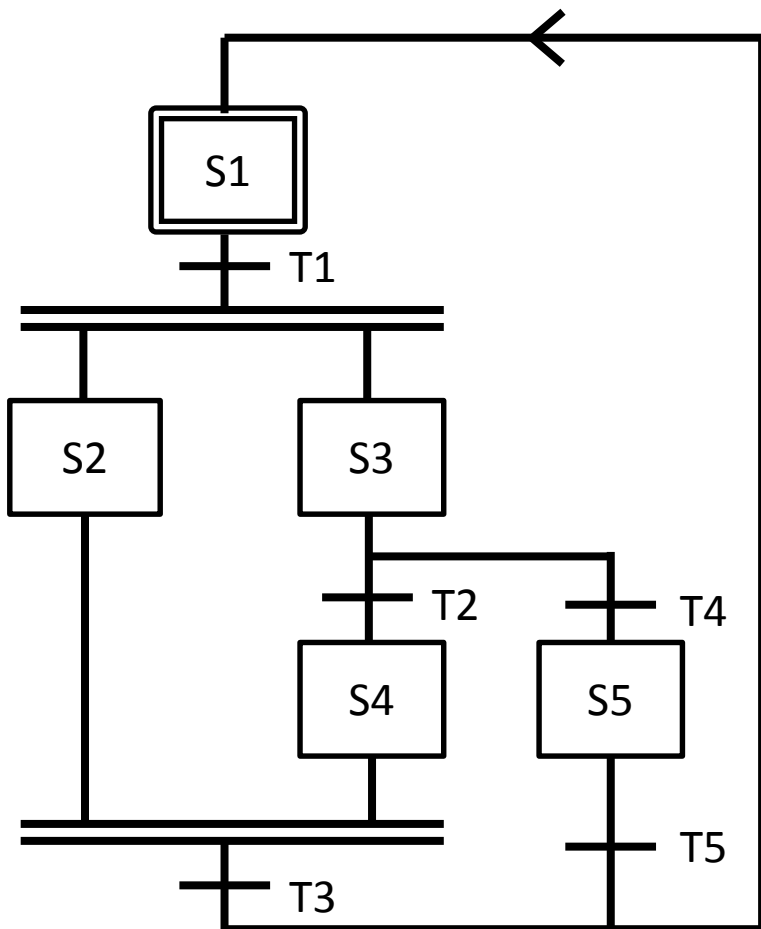
# Párhuzamos végrehajtás



# Párhuzamos végrehajtás

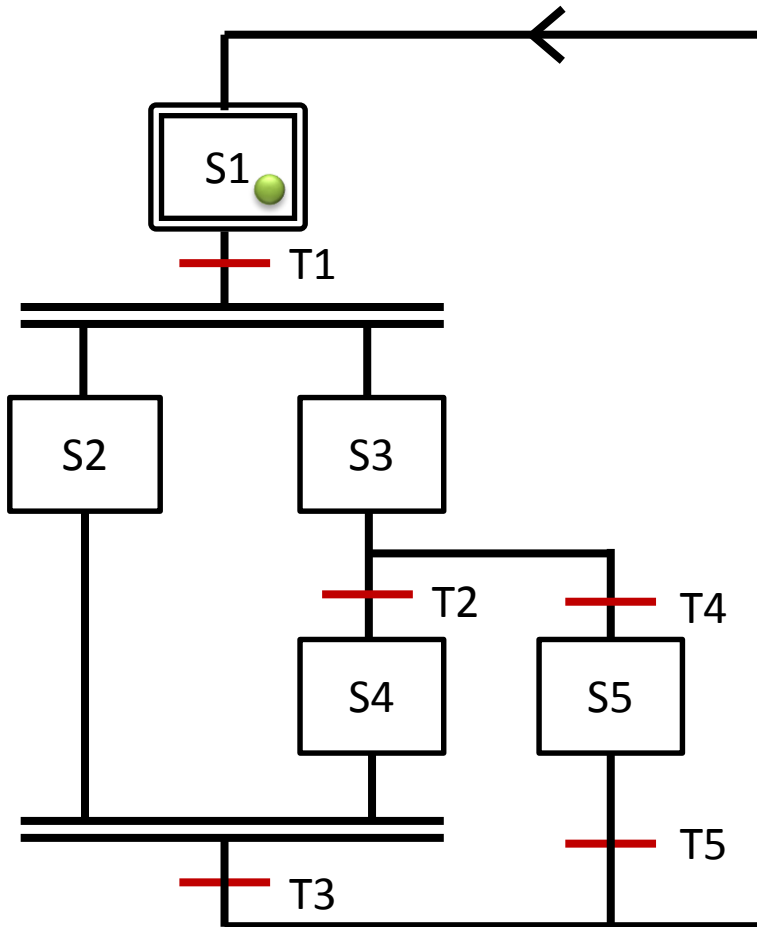


# Nem biztonságos hálózatok



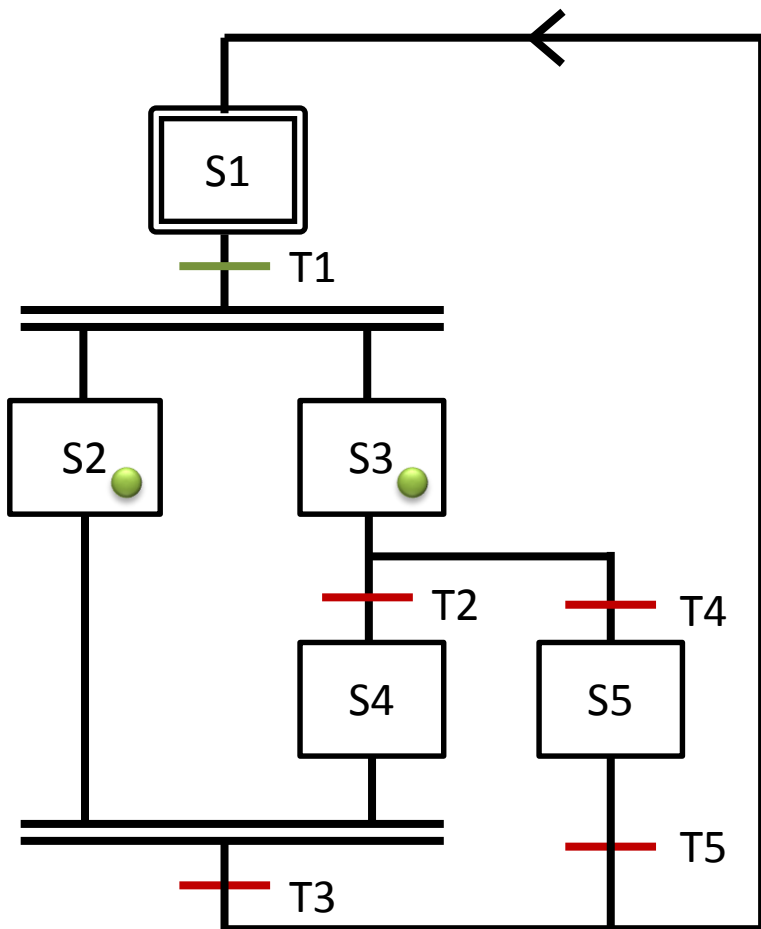
- Ha egy hálózatban az állapotok aktiválása nem kontrollált módon történik (lehetséges több token jelenléte párhuzamos szekvenciákon kívül), akkor a hálózat nem biztonságos (*unsafe*)
- A nem biztonságos hálózatok fordításkor hibát okoznak

# Nem biztonságos hálózatok



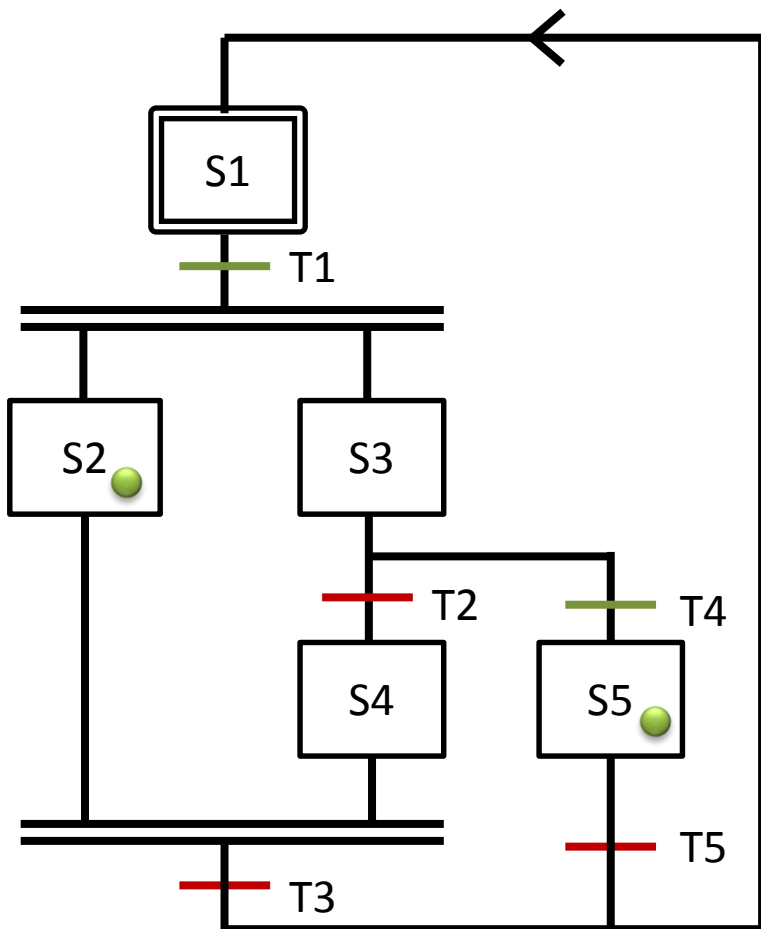
- Ha egy hálózatban az állapotok aktiválása nem kontrollált módon történik (lehetséges több token jelenléte párhuzamos szekvenciákon kívül), akkor a hálózat nem biztonságos (*unsafe*)
- A nem biztonságos hálózatok fordításkor hibát okoznak

# Nem biztonságos hálózatok



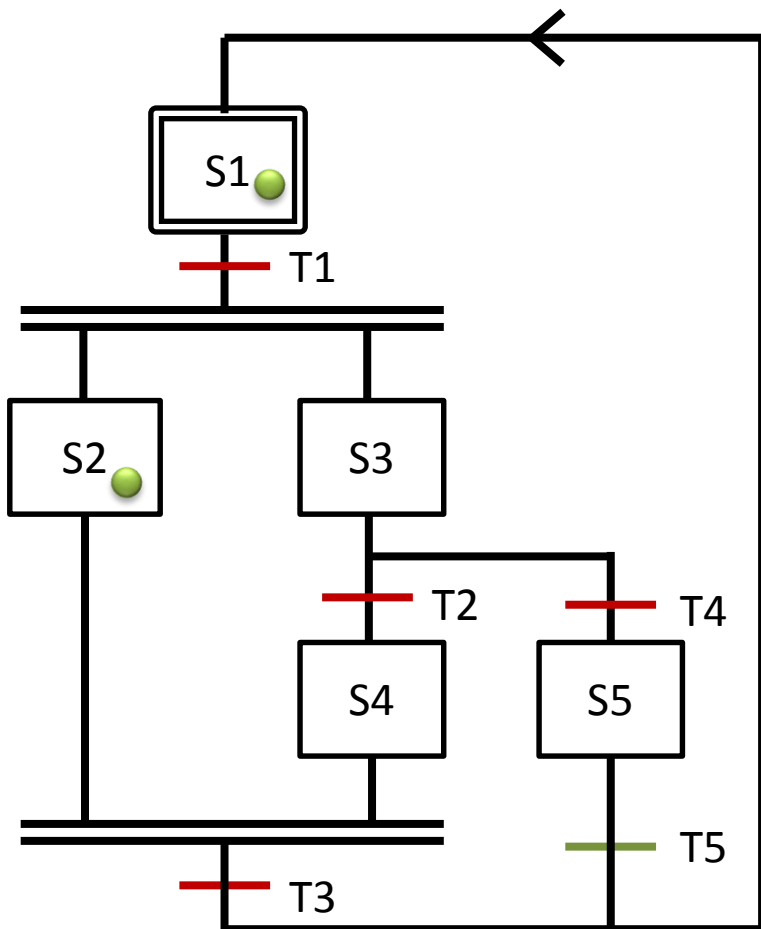
- Ha egy hálózatban az állapotok aktiválása nem kontrollált módon történik (lehetséges több token jelenléte párhuzamos szekvenciákon kívül), akkor a hálózat nem biztonságos (*unsafe*)
- A nem biztonságos hálózatok fordításkor hibát okoznak

# Nem biztonságos hálózatok



- Ha egy hálózatban az állapotok aktiválása nem kontrollált módon történik (lehetséges több token jelenléte párhuzamos szekvenciákon kívül), akkor a hálózat nem biztonságos (*unsafe*)
- A nem biztonságos hálózatok fordításkor hibát okoznak

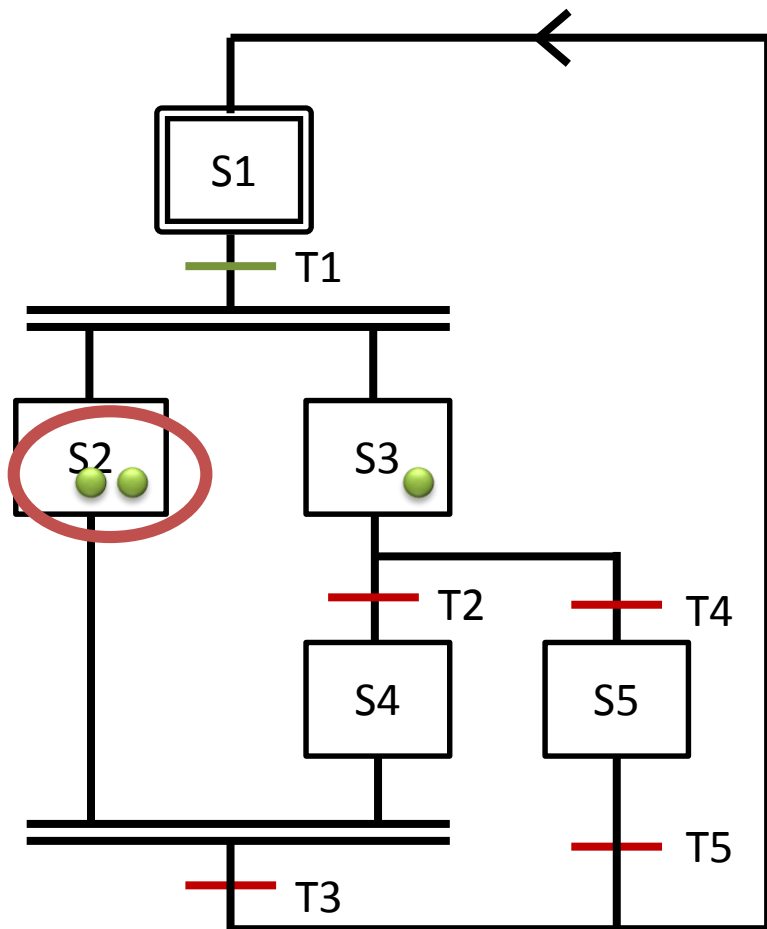
# Nem biztonságos hálózatok



- Ha egy hálózatban az állapotok aktiválása nem kontrollált módon történik (lehetséges több token jelenléte párhuzamos szekvenciákon kívül), akkor a hálózat nem biztonságos (*unsafe*)
- A nem biztonságos hálózatok fordításkor hibát okoznak

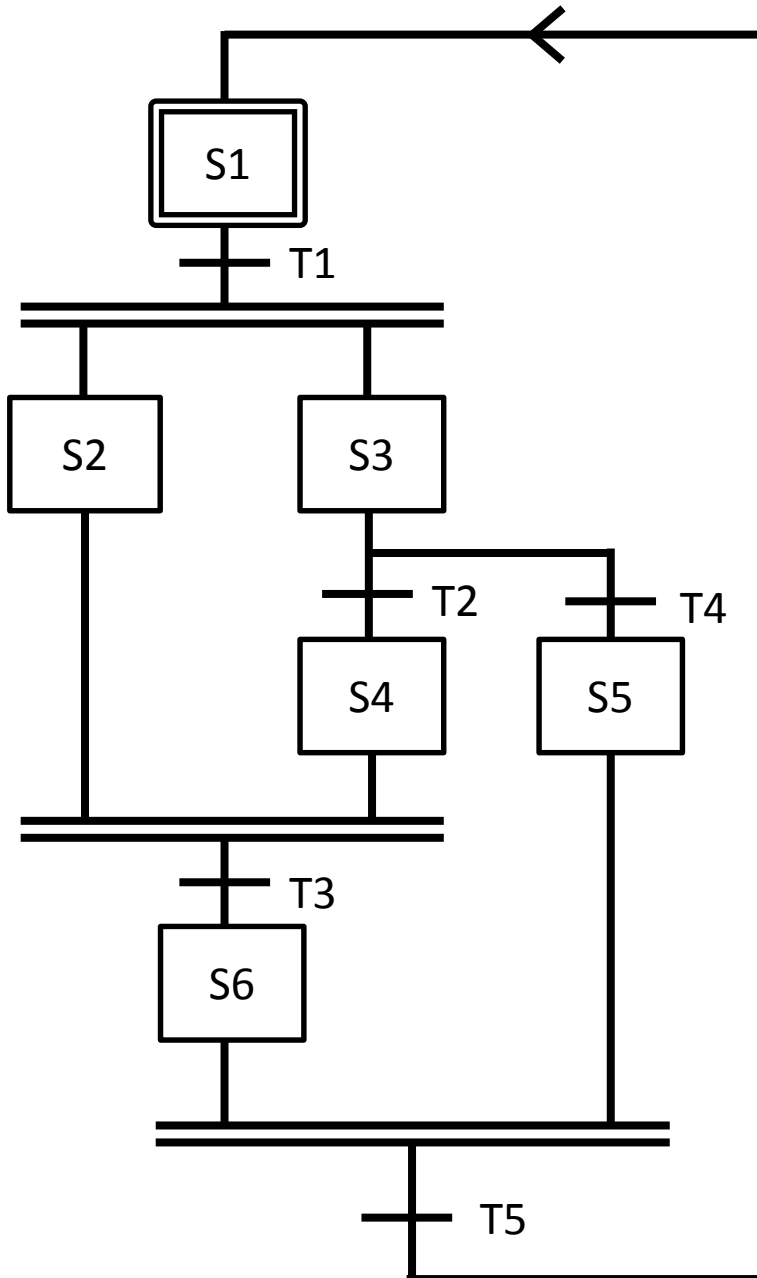


# Nem biztonságos hálózatok



- Ha egy hálózatban az állapotok aktiválása nem kontrollált módon történik (lehetséges több token jelenléte párhuzamos szekvenciákon kívül), akkor a hálózat nem biztonságos (*unsafe*)
- A nem biztonságos hálózatok fordításkor hibát okoznak

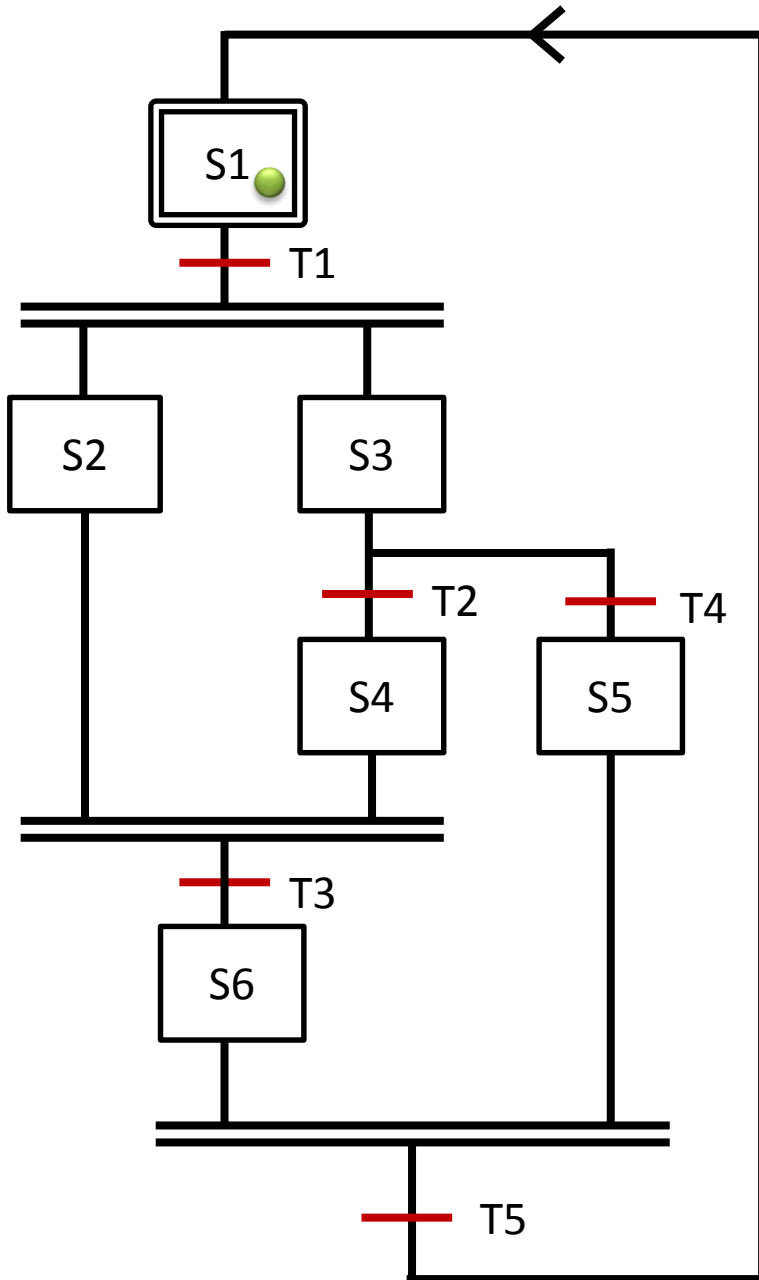
# Nem elérhető hálózatok



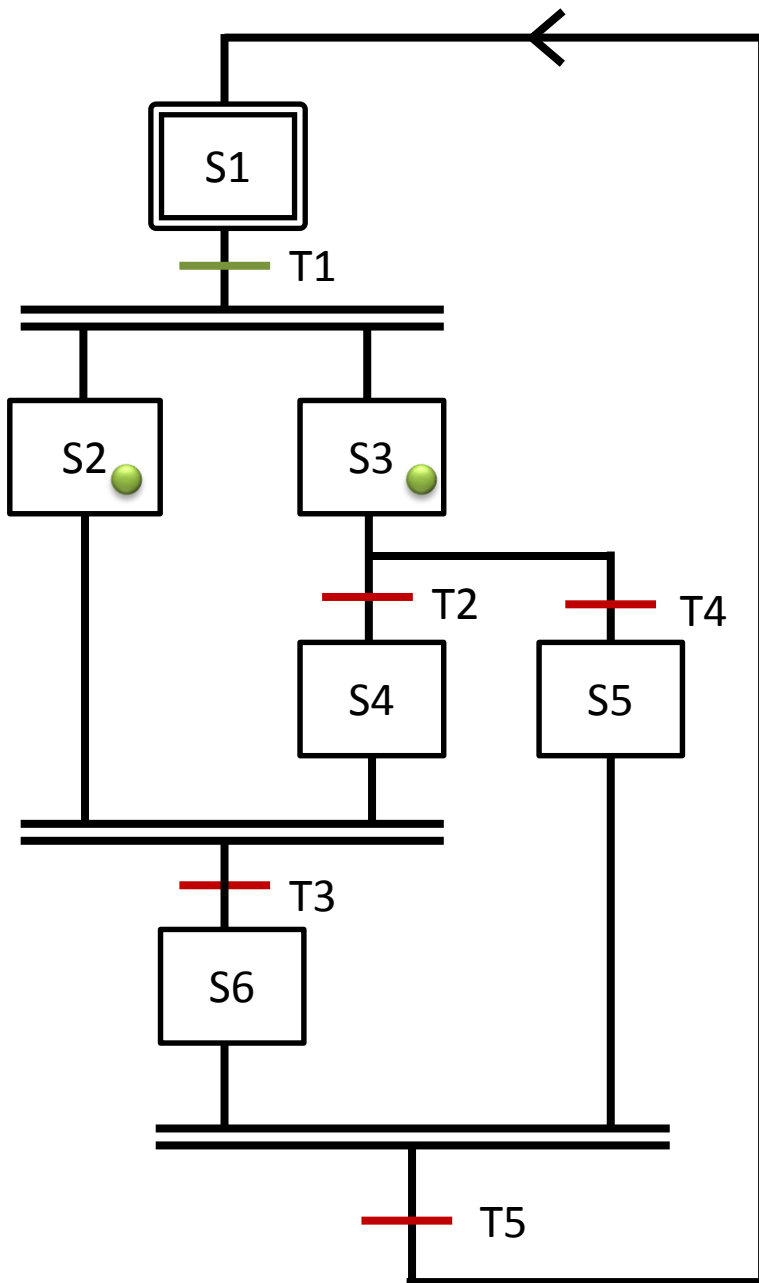
- A holtpontot tartalmazó hálózatok nem elérhetők (*unreachable*)
- A nem elérhető hálózatok hibát okoznak fordítás során

# Nem elérhető hálózatok

- A holtpontot tartalmazó hálózatok nem elérhetők (*unreachable*)
- A nem elérhető hálózatok hibát okoznak fordítás során

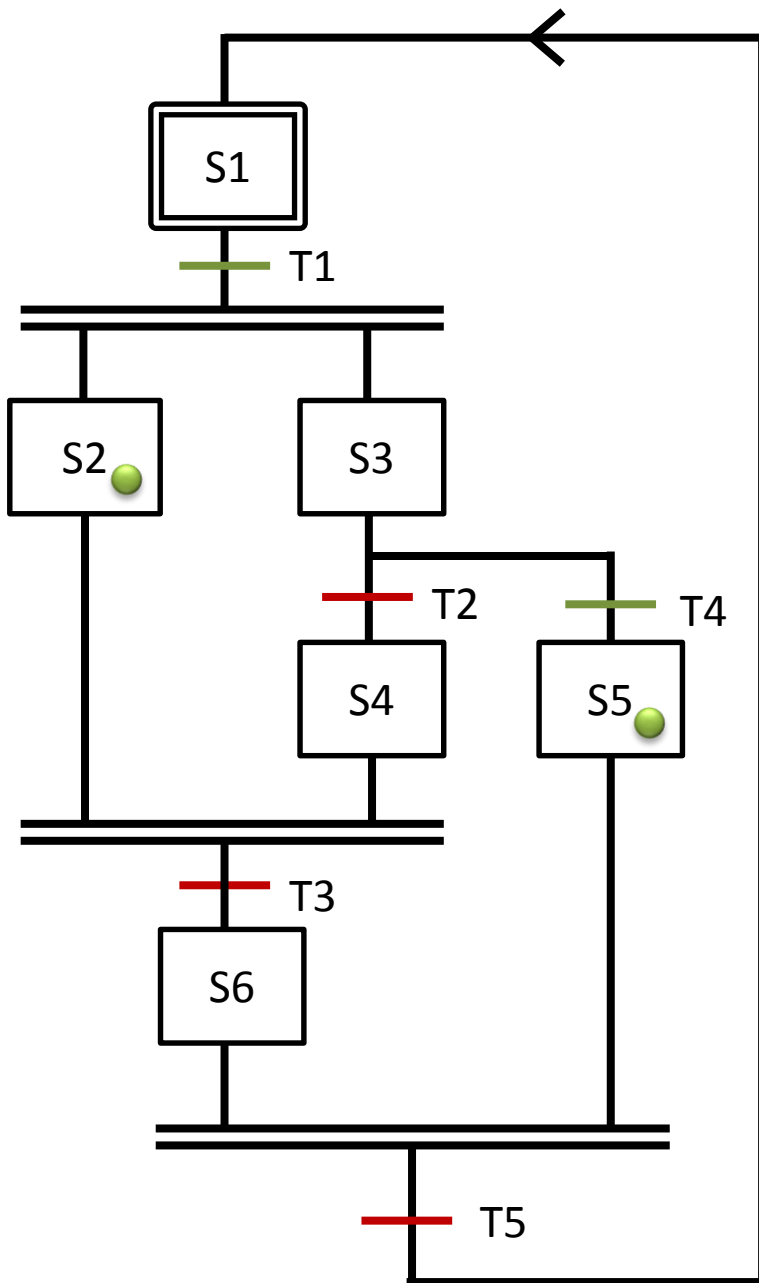


# Nem elérhető hálózatok



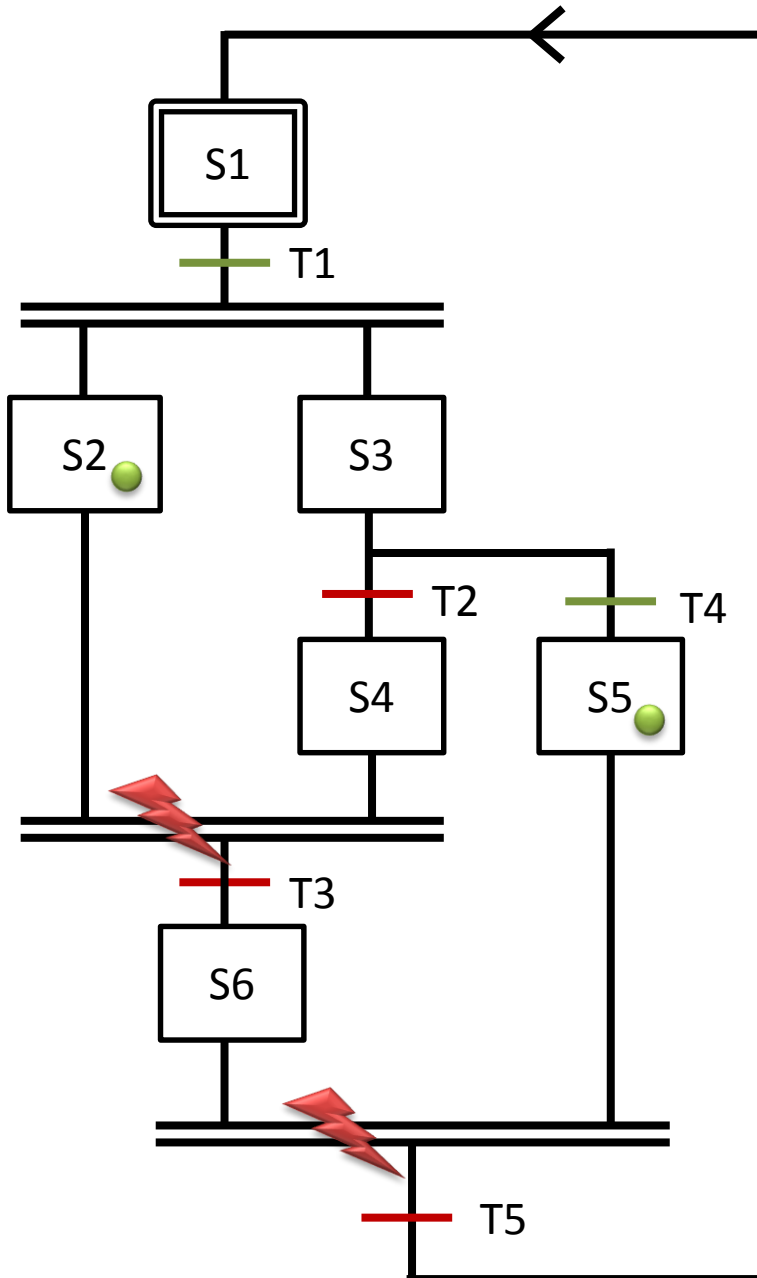
- A holtpontot tartalmazó hálózatok nem elérhetők (*unreachable*)
- A nem elérhető hálózatok hibát okoznak fordítás során

# Nem elérhető hálózatok



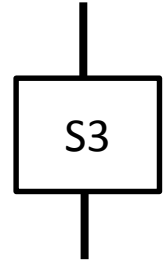
- A holtpontot tartalmazó hálózatok nem elérhetők (*unreachable*)
- A nem elérhető hálózatok hibát okoznak fordítás során

# Nem elérhető hálózatok



- A holtpontot tartalmazó hálózatok nem elérhetők (*unreachable*)
- A nem elérhető hálózatok hibát okoznak fordítás során

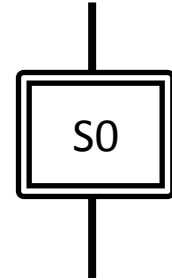
# Lépések



- Téglalap és azonosító: `StepName`
- Lépés flag: `StepName.X`
  - Logikai változó, értéke igaz, ha az adott lépés aktív
- Lépésidő: `StepName.T`
  - Időtartam-változó, értéke a lépés aktiválása óta eltelt idő
  - Értéke a lépés deaktiválásakor befagyasztódik, aktiválásakor  $t_{\#0s}$ -ról indul újra
- A lépés-flag és a lépésidő csak olvashatók

# Kezdeti lépés

- Jelölés: kettős körvonal
- Tetszőleges azonosító
- A hozzá tartozó lépés-flag kezdeti értéke TRUE
- Minden hálózat egy és csakis egy kezdeti lépést tartalmaz





# Lépések szöveges megadása

- Lépés

```
STEP StepName  
    (* step body *)  
END_STEP
```

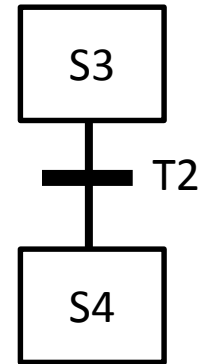
- Kezdeti lépés

```
INITIAL_STEP StepName  
    (* step body *)  
END_STEP
```

- A szöveges megadás lehetősége a szabványban szerepel, de a fejlesztői környezetek általában nem támogatják

# Átmenetek

- Vízszintes vonal a lépéseket összekötő függőleges élen
- Minden átmenethez egy és csakis egy feltétel tartozik
  - Logikai értékű kifejezés
  - Az átmenet akkor tüzel, ha igazra értékelődik ki
  - Feltétel nélküli átmenetek konstans TRUE (1) feltétellel valósíthatók meg



# Átmenetek szöveges megadása

```
TRANSITION TranName FROM Step1 TO Step2  
    (* body *)  
END_TRANSITION
```

- A szabványban szerepel, de a fejlesztői környezetek általában nem támogatják

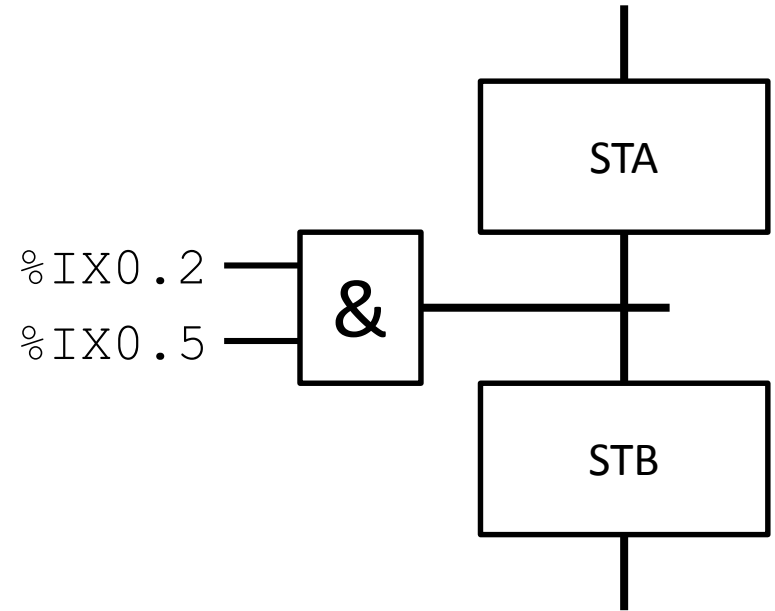
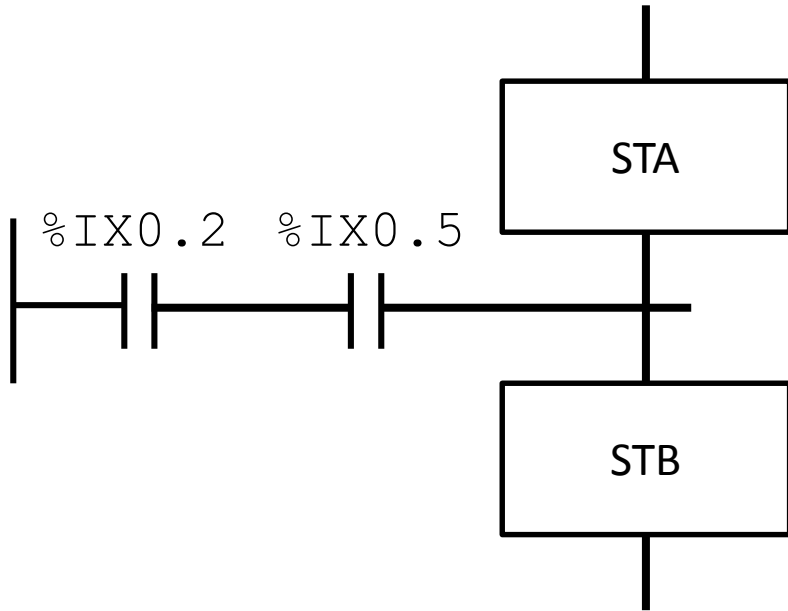
# Átmenet-feltételek megadása

- Közvetlen megadás ST nyelven

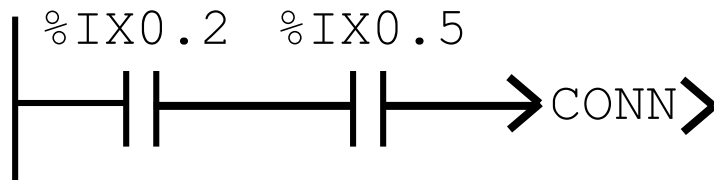


# Átmenet-feltételek megadása

- Közvetlen megadás LD vagy FBD nyelven

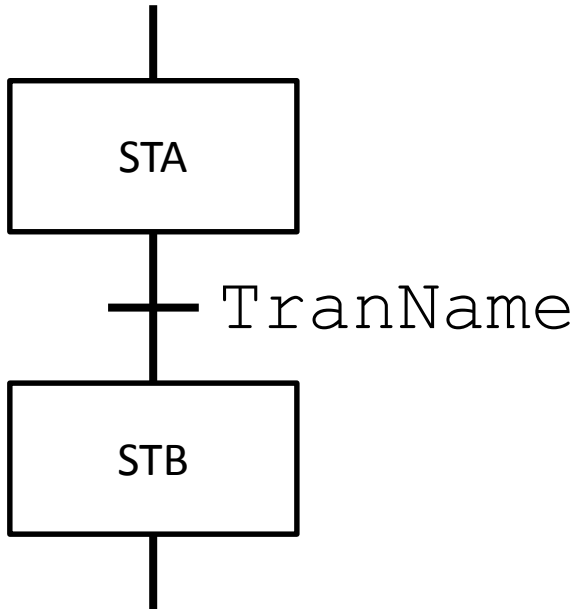


- Az összekötők használata megengedett



# Átmenet-feltételek megadása

- Közvetett módon az átmenet nevével



```
TRANSITION TranName FROM STA TO STB:  
(* LD, IL, FB, ST *)  
END_TRANSITION
```

# Átmenet-feltétel törzse

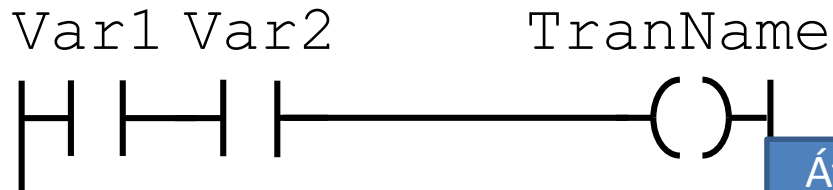
- ST: hozzárendelés egy kifejezéshez (bal oldalon hiányzik a változó)

`:=Var1 & Var2;`

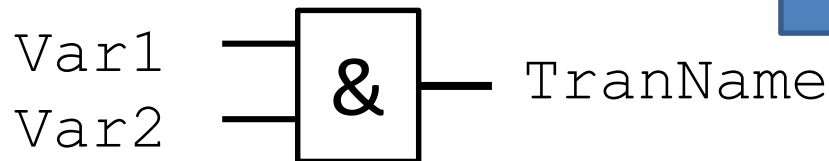
- IL: A feltétel értéke az akkumulátor értéke az utolsó művelet után

`LD Var1  
AND Var2`

- LD



- FBD



Átmenet nevével  
megegyező  
kimenet

# Akciók (actions)

- Minden lépéshez nulla vagy több akció rendelhető
- Logikai akció: logikai változó, amit az akció állít be
- Nem logikai akció:
  - IL utasítások
  - ST műveletek
  - LD hálózatok
  - FBD hálózatok
  - Egy másik SFC



# Nem-logikai akciók deklarálása

- LD, FBD, SFC: grafikus deklaráció  
(implementációfüggő, általában a POU-kkal  
megegyező módon)
- ST, IL: ACTION kulcsszó

```
ACTION MyAction:
```

```
    %Q0.1 := %IX0.0 & Step8.X;
```

```
END_ACTION
```

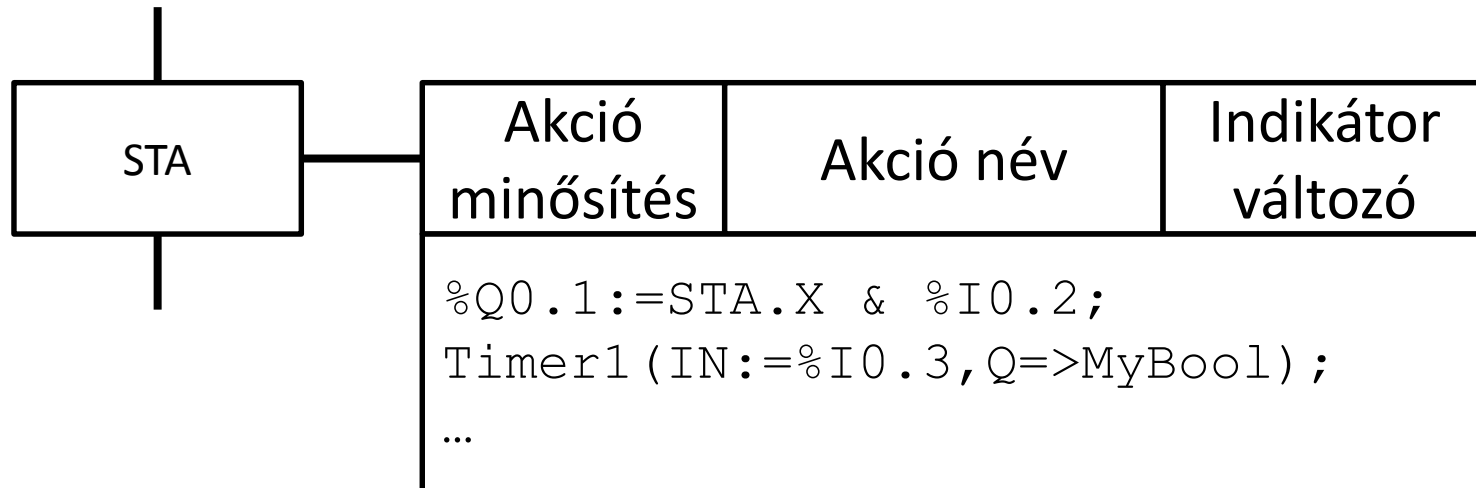
# Lépések és akciók hozzárendelése

- Grafikusan: a lépéshez kapcsolt akcióblokk(ok)al



# Közvetlen akciódefiníció

- Logikai akció: ha létezik az akcióéval megegyező nevű VAR vagy VAR\_OUT típusú változó, akkor az lesz a logikai akció
- Műveletek vagy hálózatok: az akció törzse az akcióblokkon belül is megadható (ekkor az akciónév más akcióblokkokban nem használható)



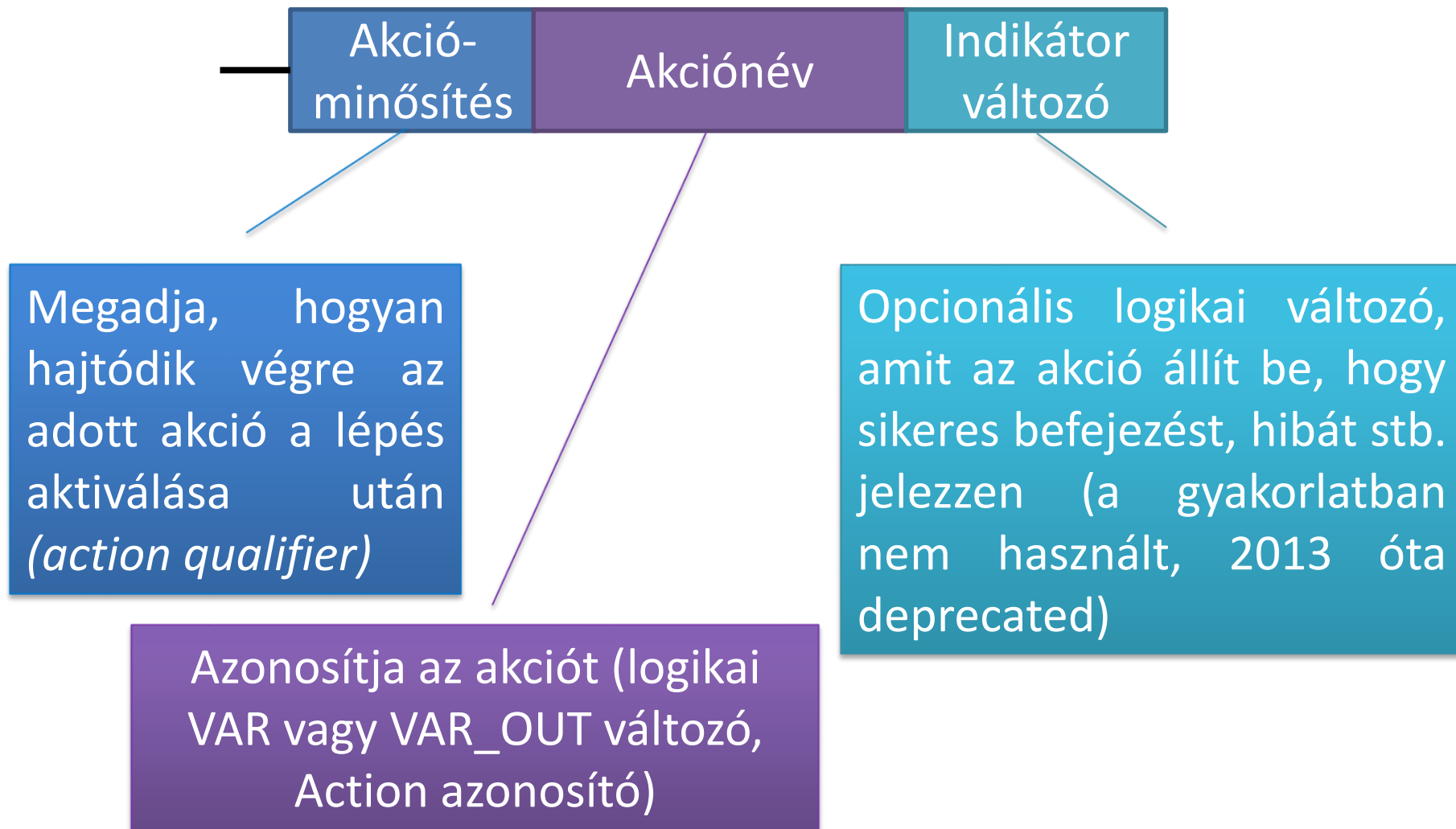
# Lépések és akciók hozzárendelése

- Szövegesen: a STEP blokkokban

```
STEP StepName  
    ActionName (Qual, QualParameter, IndVar) ;  
END_STEP
```

- A fejlesztői környezettől függő módon
  - Általában akcióblokk
  - Az akció törzse külön ablakban szerkeszthető

# Az akcióblokk szerkezete

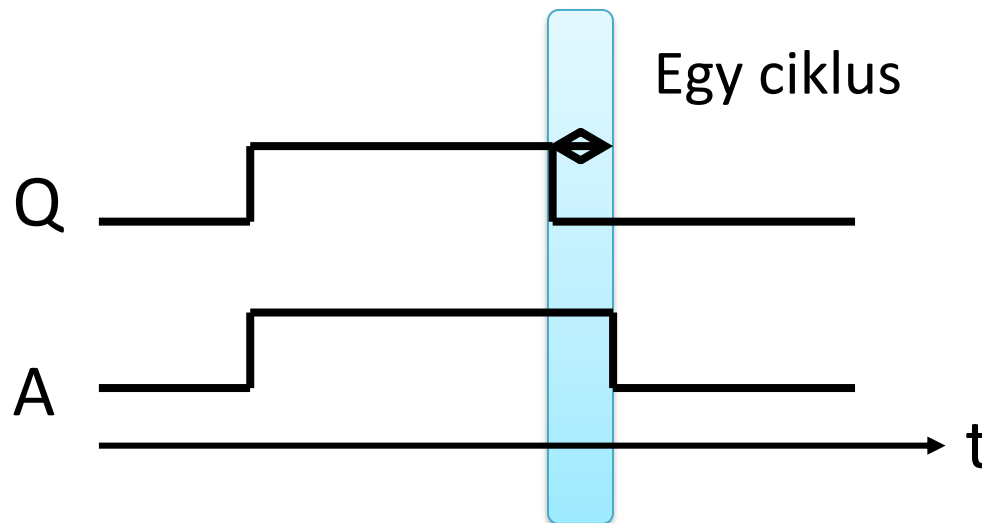


# Akcióvezérlés

- A felhasználó előtt rejtett blokk
- Az akcióminősítéstől függően állítja a flageket:
  - Akció-flag (Q) – nem logikai akciók esetén  
`ActionName.Q` néven érhető el
  - Aktivitás-flag (A) – csak nem logikai akciók esetén
- Logikai akció: a változót az akció-flag értékére állítjuk
- Nem logikai akció: ciklikusan fut, amíg az akció-flag TRUE értékű

# Aktivitás-flag

- Az akció-flaggel együtt állítódik be, annak lefutó éle után még egy ciklus idejéig aktív
- Az akció adott végrehajtási ciklusa az utolsó, ha  
 $(\text{Action.Q} = \text{FALSE}) \ \& \ (\text{Action.A} = \text{TRUE})$



# Aktivitás-flag

- A fejlesztői környezetek többségében nem érhető el közvetlenül
- Hasonló konstrukciók
  - Exit akció – egyszer, a lépés deaktiválásakor fut le
  - Az akcióból elérhető Last Scan bit – az utolsó végrehajtáskor aktív



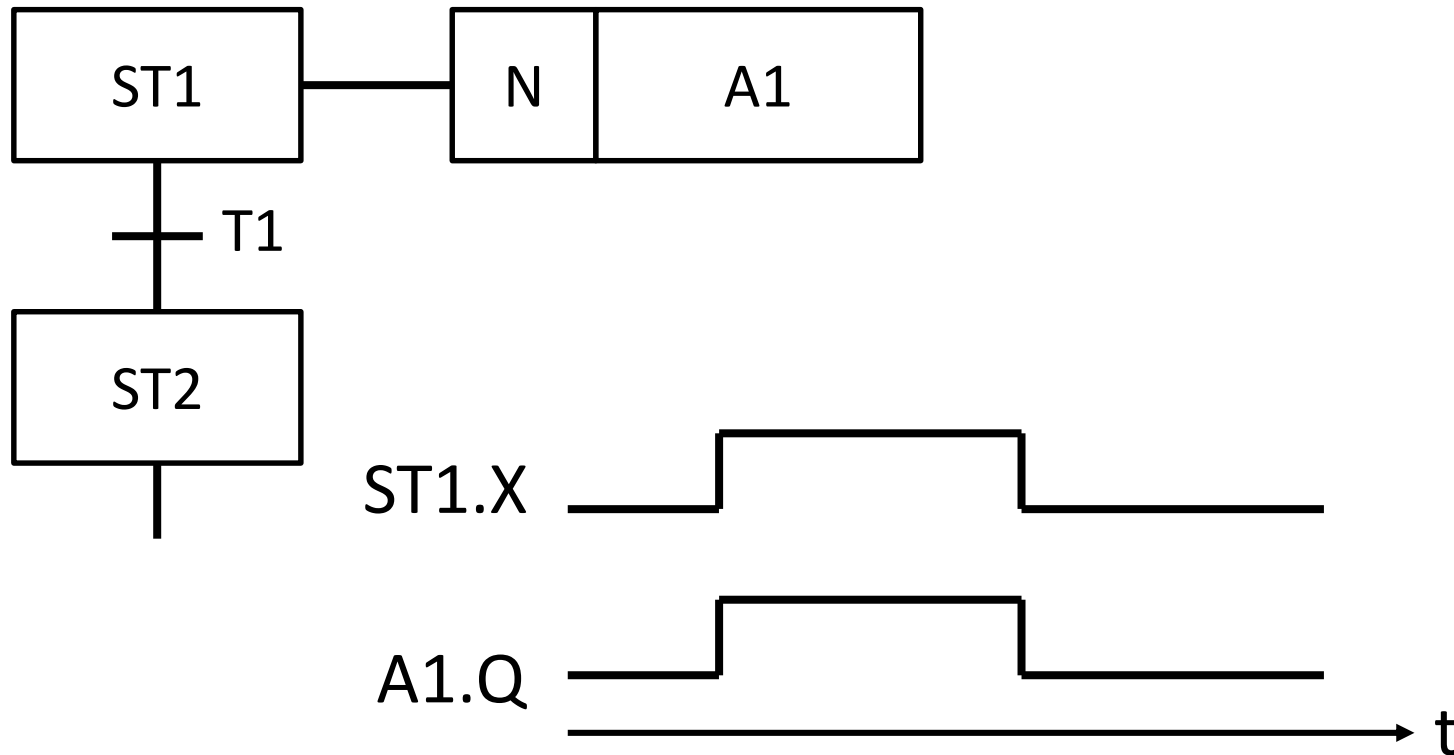
# Akcióminősítések

Minősítés	Értelmezés
None / N	Nem tárolt ( <b>N</b> on-stored, null qualifier)
R	Reset (Overriding <b>R</b> eset, tárolt akcióra)
S	Tárolt ( <b>S</b> et, <b>S</b> tored)
L	Időben korlátozott (Time <b>L</b> imited)
D	Időben késleltetett (Time <b>D</b> elayed)
P	Impulzus ( <b>P</b> ulse)
SD	Tárolt és késleltetett ( <b>S</b> tored and <b>D</b> elayed)
DS	Késleltetett és tárolt ( <b>D</b> elayed and <b>S</b> tored)
SL	Tárolt és korlátozott ( <b>S</b> tored and <b>L</b> imited)
<i>P1</i>	<i>Felfutó él érzékeny (<b>P</b>ulse (rising edge))</i>
<i>P0</i>	<i>Lefutó él érzékeny (<b>P</b>ulse (falling edge))</i>

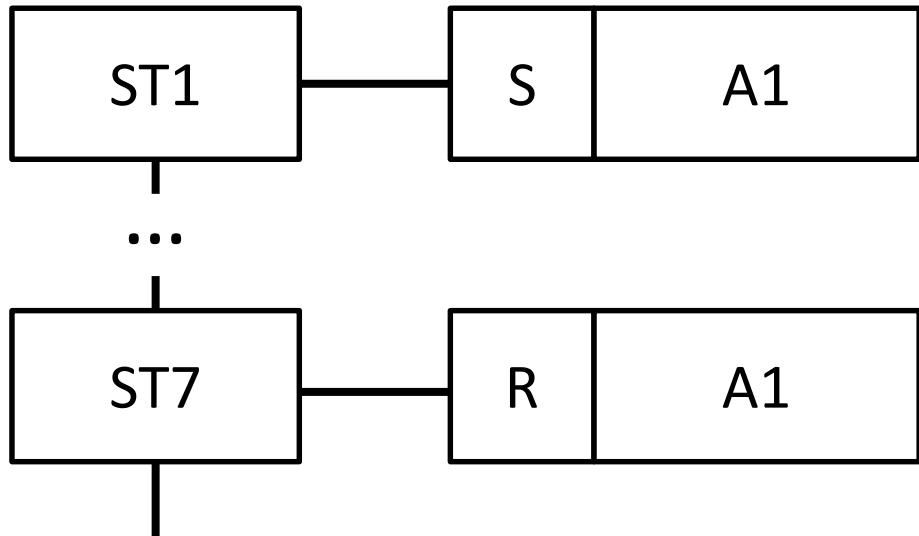
Gyakorlatban  
nem használt

# Nem tárolt akció

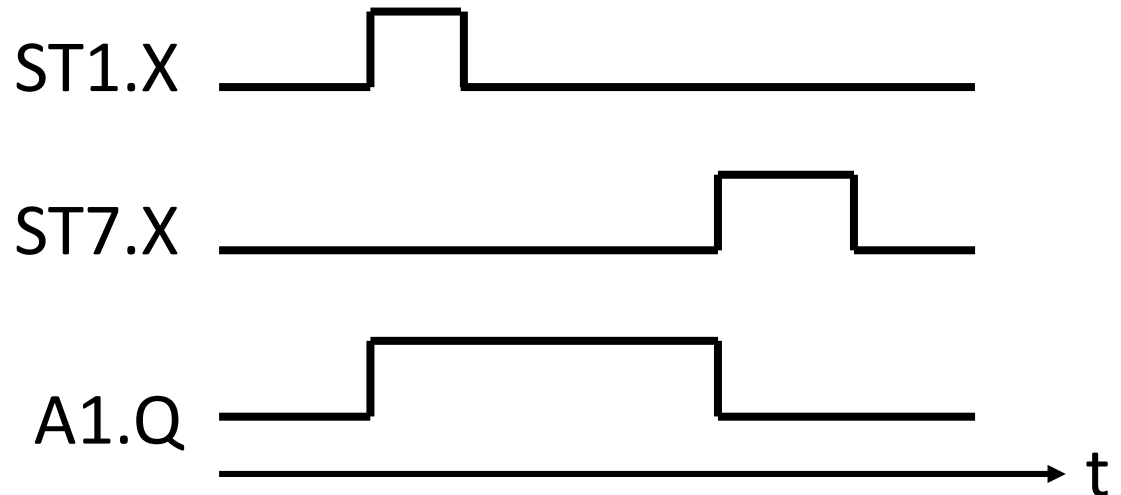
- A Q akció-flag a lépés-flag másolata



# Tárolt akció

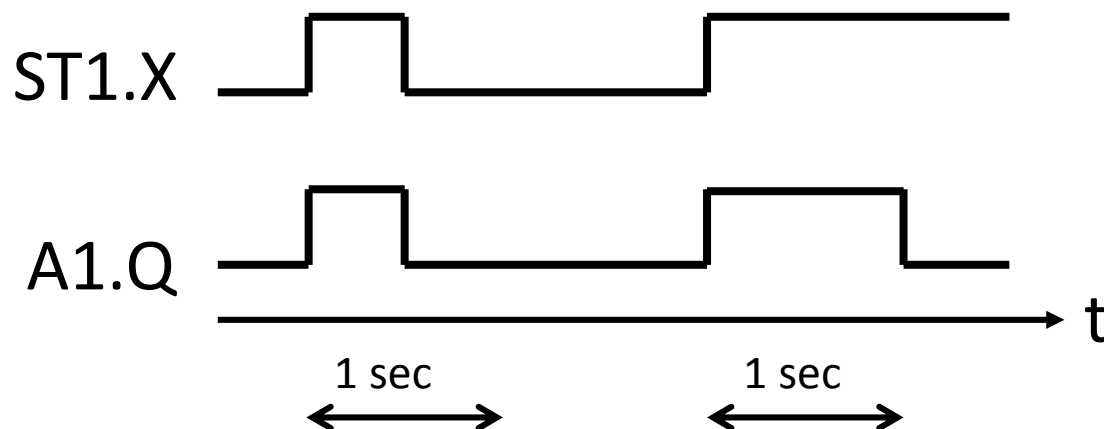
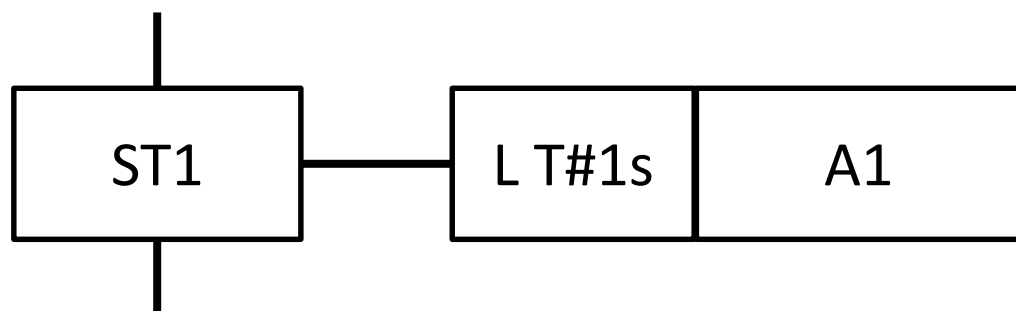


- A Reset művelet magasabb prioritású



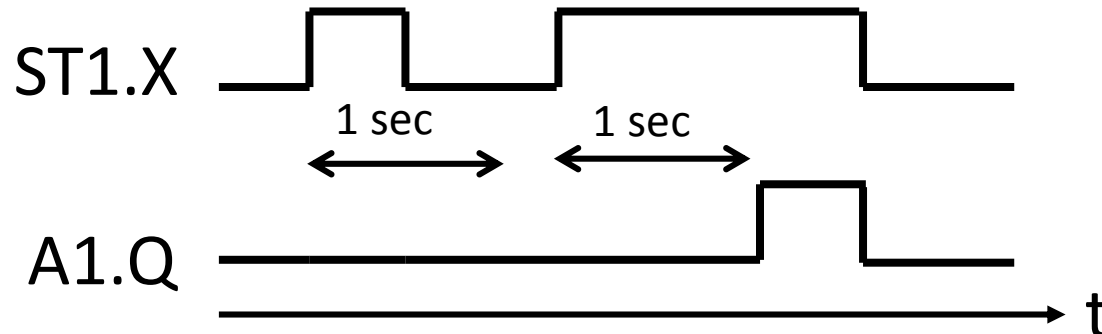
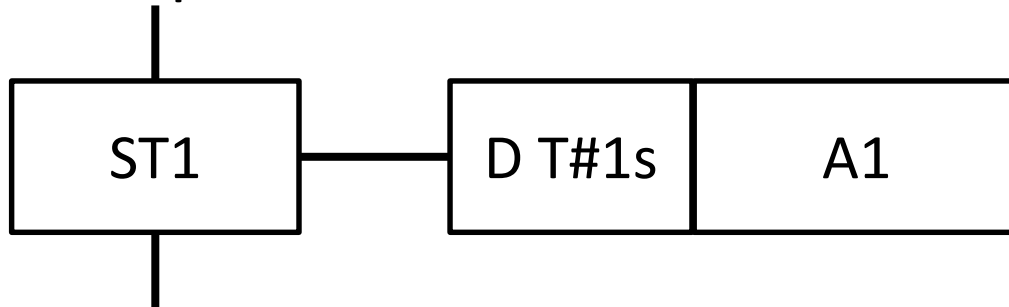
# Időben korlátozott akció

- A Q akció-flag a lépés aktiválásával állítódik be
- Annak deaktiválásáig, de legfeljebb a megadott ideig aktív



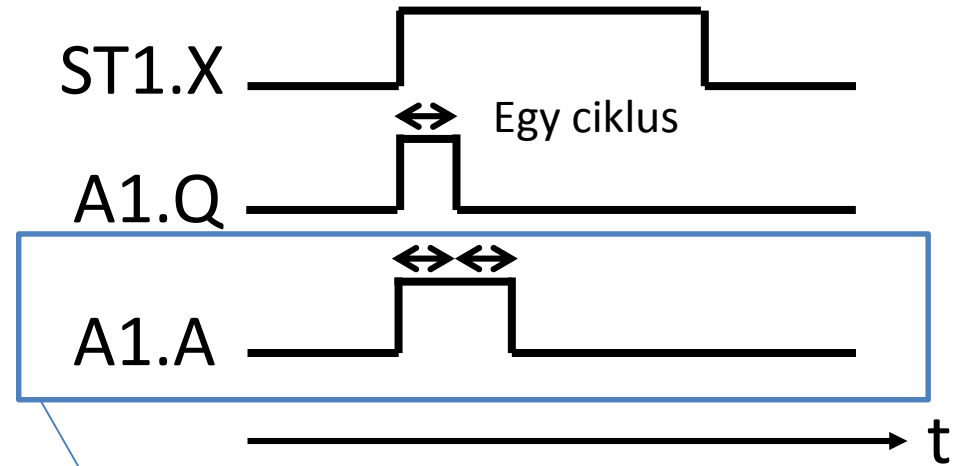
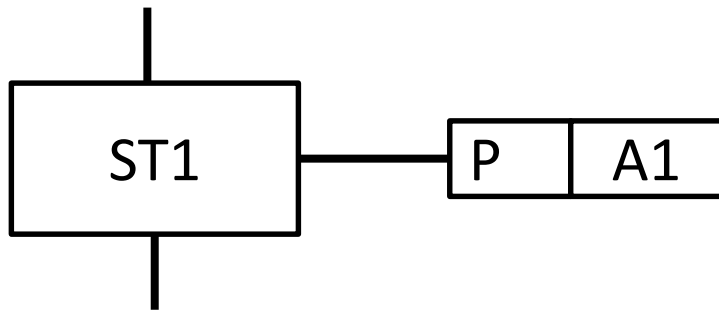
# Időben késleltetett akció

- A Q akció-flag a lépés aktiválása után megadott idővel állítódik be, ha a lépés akkor még aktív
- A lépés deaktiválásakor törlődik



# Impulzus-akció

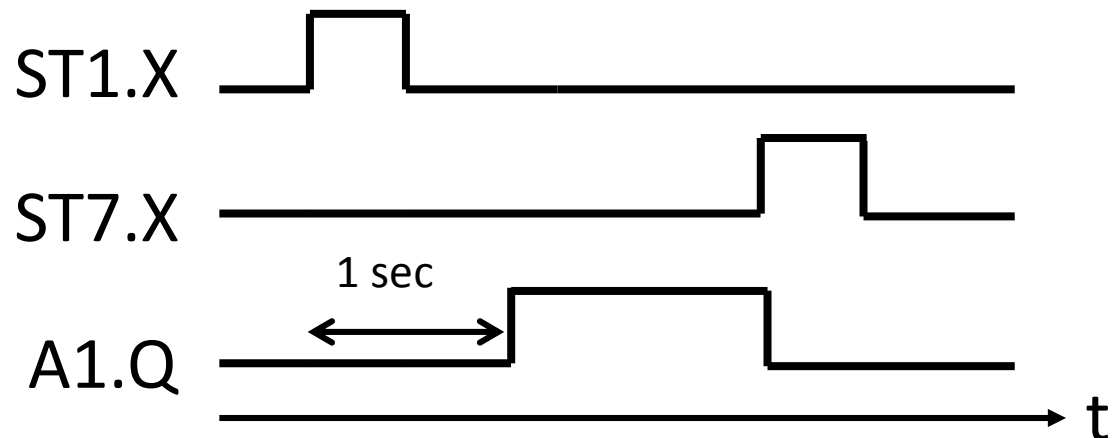
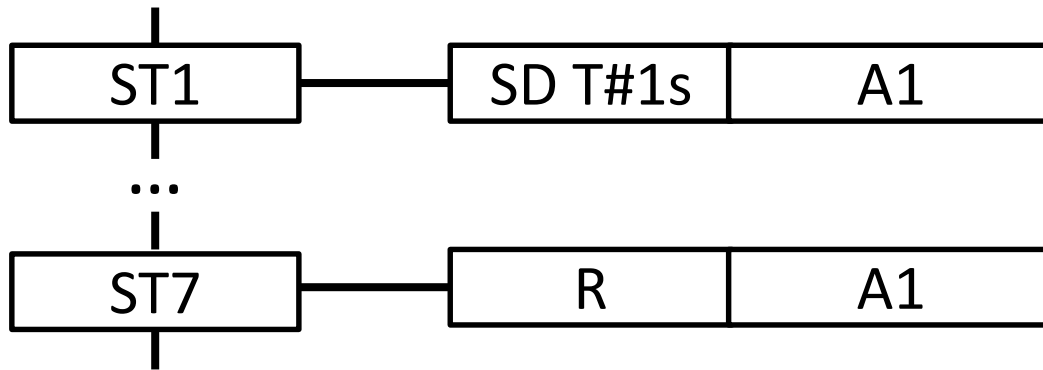
- Az akció-flag a lépés aktiválása után egyetlen ciklus idejéig aktív
- Egyes fejlesztői környezetekben külön Entry Action definiálható



A fejlesztői környezetek túlnyomó többségében nem érhető el.

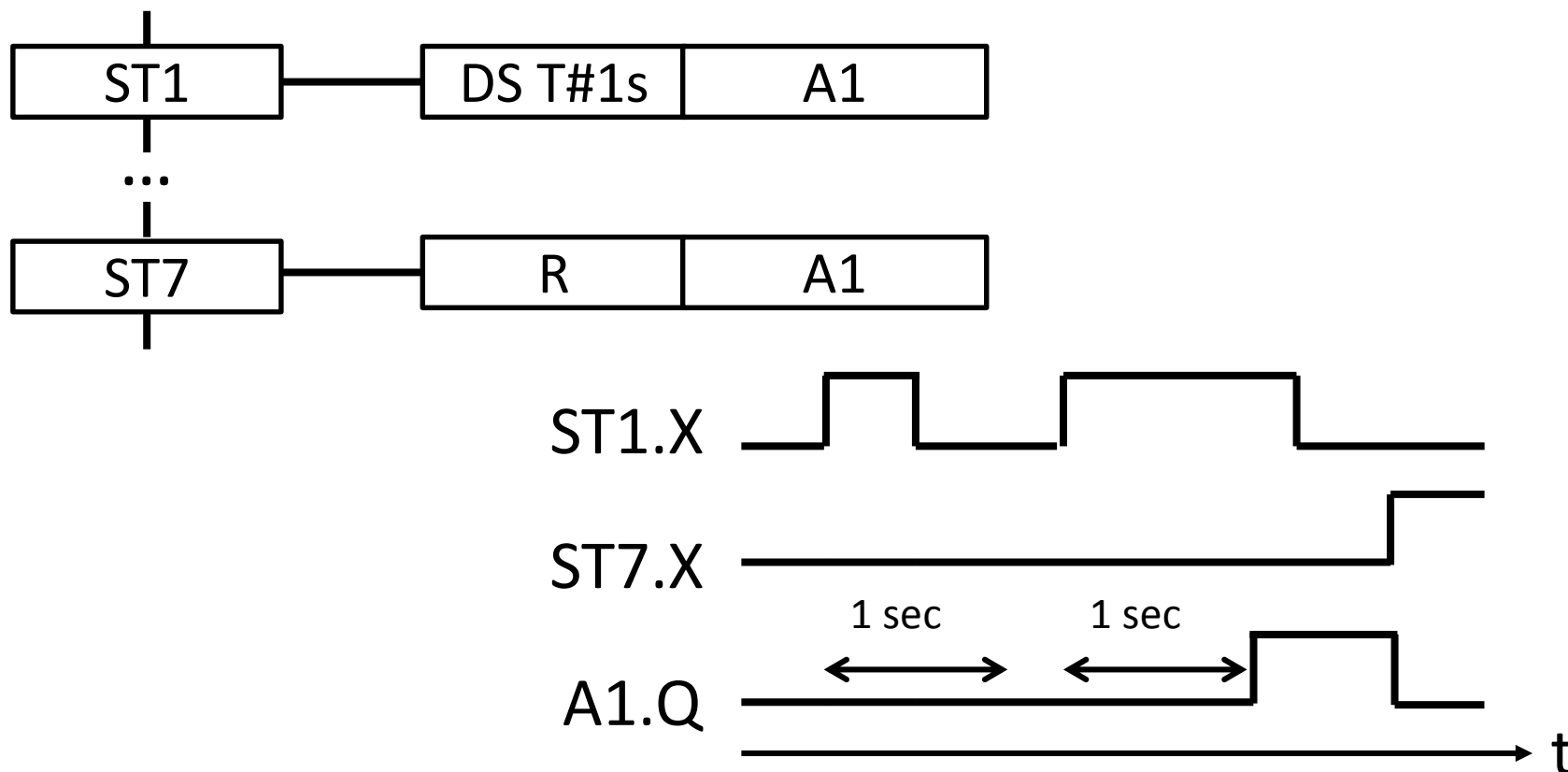
# Tárolt és késleltetett akció

- A Q akció-flag a lépés aktiválása után megadott idővel állítódik be, akkor is, ha a lépés már nem aktív
- Csak Reset-akció törli



# Késleltetett és tárolt akció

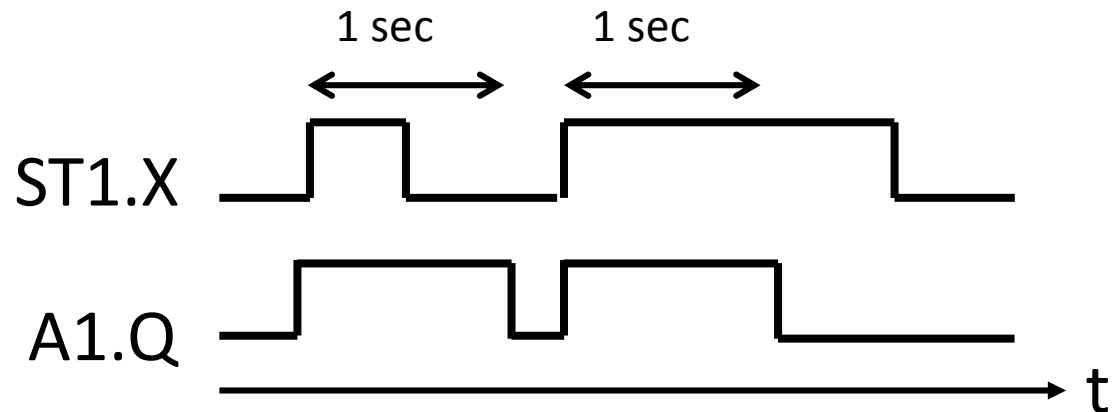
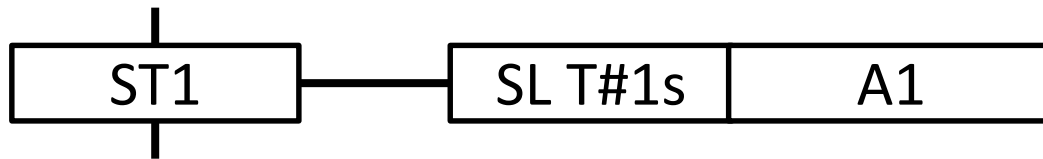
- A Q akció-flag a lépés aktiválása után megadott idővel állítódik be, amennyiben a lépés még aktív
- Csak Reset-akció törli, az állapot deaktiválása nem





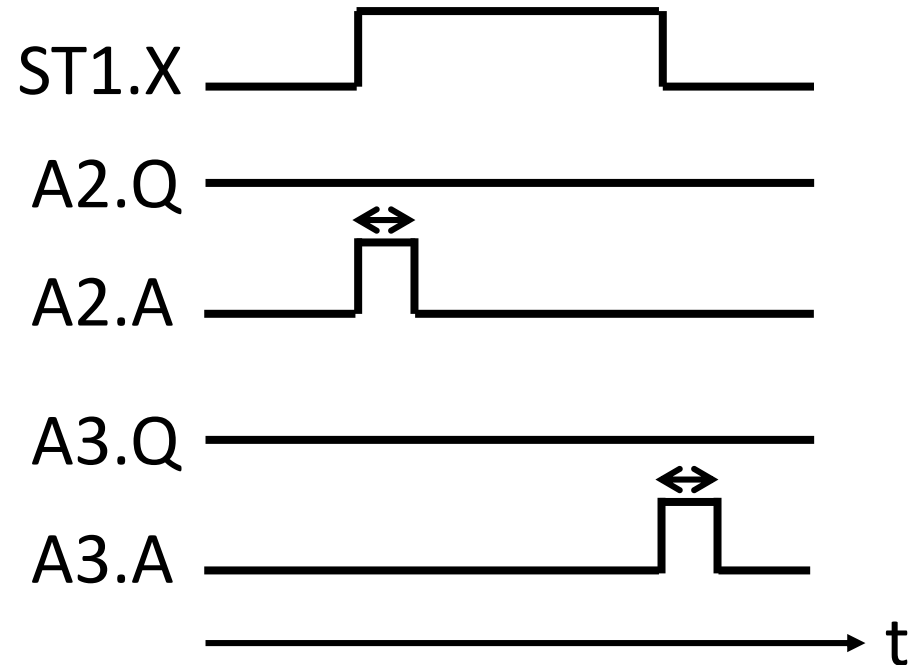
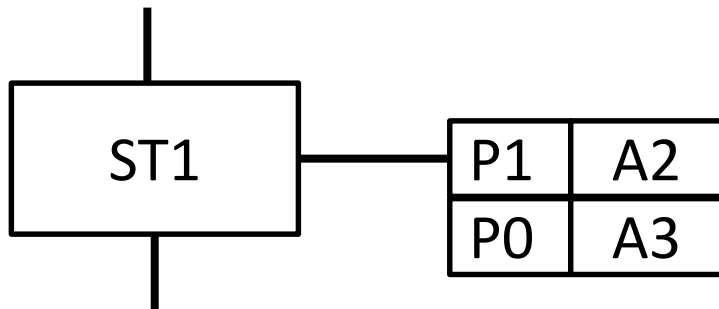
# Tárolt és korlátozott akció

- A Q akció-flag a lépés aktiválásakor állítódik be
- A megadott idő után törlődik a lépés aktivitásától függetlenül



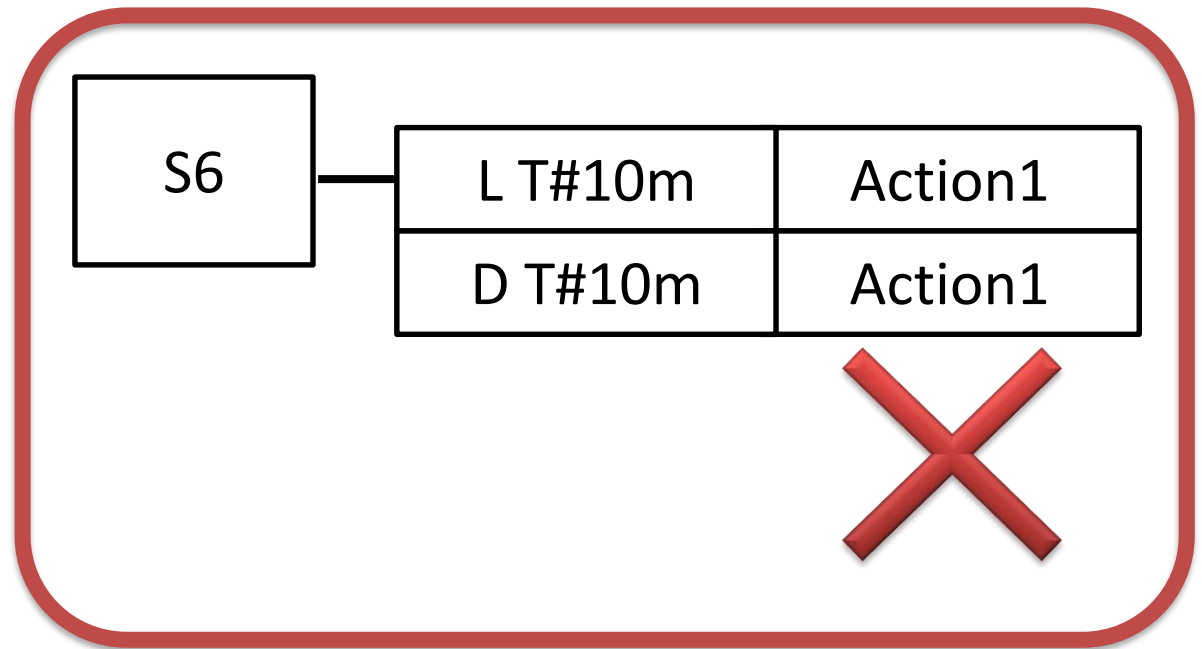
# P1 és P0 impulzus-akciók

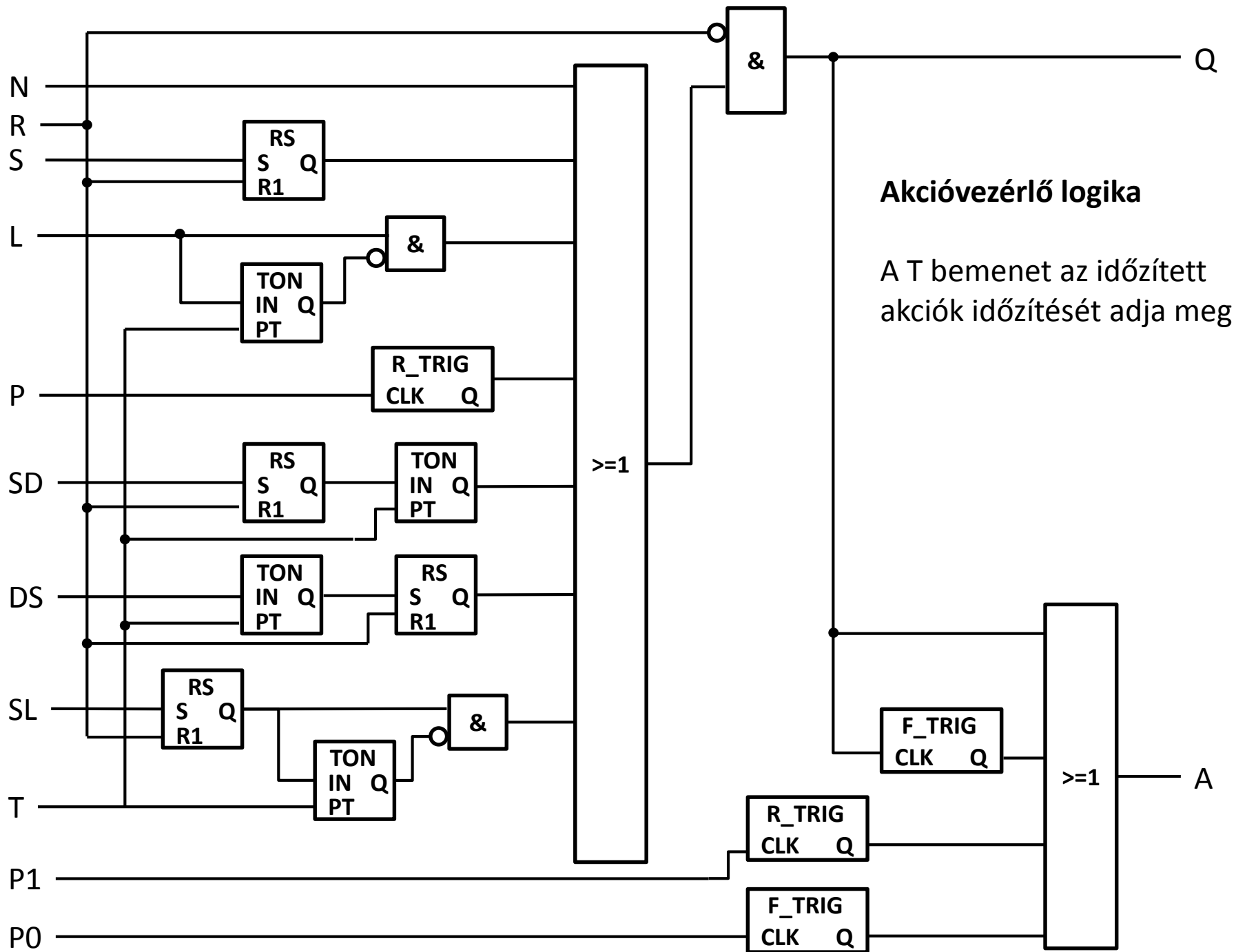
- A szabványban definiáltak, de a fejlesztői környezetek általában nem implementálják (helyettük más megoldások – lépéshez kapcsolt entry és exit action)
- A P1 és P0 akcióminősítések logikai akciókra nem értelmezettek (hatástalanok)
- Csak az aktivitás flaget állítják, az akció flaget nem



# Időzített minősítések

- Egy akció ugyanazon lépésnél csak egy időzített minősítéshez kapcsolható
  - L
  - D
  - SD
  - DS
  - SL

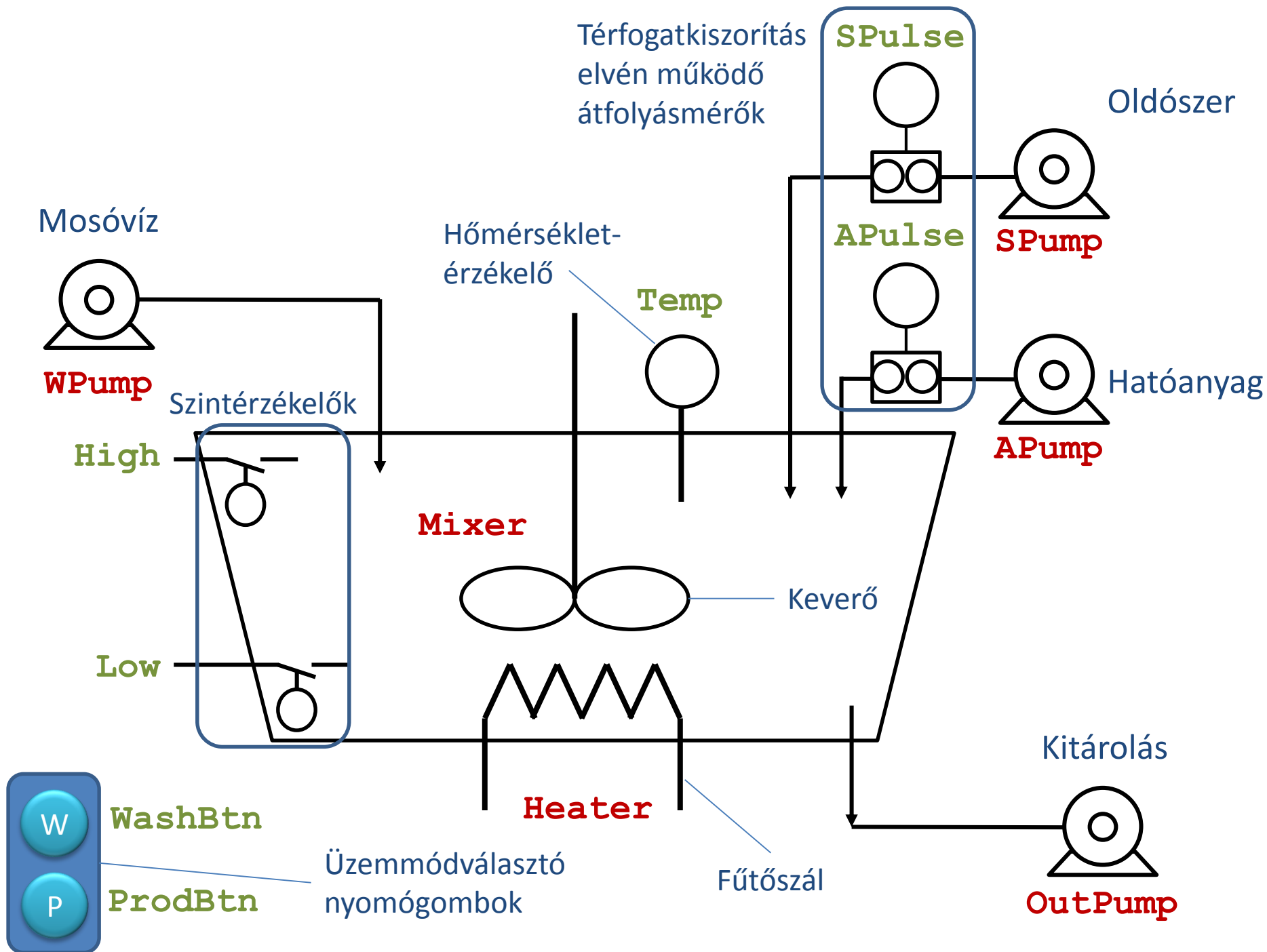




# Példa

Egy keverővel is ellátott autokláv a következő üzemmódokban működik:

- Mosás: a tartályt megtöltjük vízzel, majd 10 percen át fűtjük. Ezután bekapcsoljuk a keverőt, és addig működtetjük, amíg a víz hőmérséklete 30 fok alá nem süllyed, majd kiszivattyúzzuk a vizet.
- Termelés: a tartályba a megadott mennyiségű hatóanyagot (7 egység) és oldószert (30 egység) szivattyúzzuk, majd az elegyet 10 percig fűtjük. Ezután bekapcsoljuk a keverőt, és addig működtetjük, amíg az oldat hőmérséklete 30 fok alá nem süllyed, majd kiszivattyúzzuk azt.



## Bemenetek

Változó	Típus	Értelmezés
APulse	BOOL	Hatóanyag-átfolyásmérő: felfutó éle egy egység betárolását jelzi
SPulse	BOOL	Oldószer-átfolyásmérő: felfutó éle egy egység betárolását jelzi
High	BOOL	Szintérzékelő – teli szint (0: szint alatta, 1: szint felette)
Low	BOOL	Szintérzékelő – üres szint (0: szint alatta, 1: szint felette)
WashBtn	BOOL	Mosás üzemmódválasztó nyomógomb
ProdBtn	BOOL	Termelés üzemmódválasztó nyomógomb
Temp	USINT	Folyadék hőmérséklete [°C]

## Kimenetek

Változó	Típus	Értelmezés
WPump	BOOL	Mosóvíz szivattyú (0: ki, 1: be)
SPump	BOOL	Oldószer szivattyú (0: ki, 1: be)
APump	BOOL	Hatóanyag szivattyú (0: ki, 1: be)
OutPump	BOOL	Kitároló szivattyú – ürítés (0: ki, 1: be)
Mixer	BOOL	Keverő (0: ki, 1: be)
Heater	BOOL	Fűtés (0: ki, 1: be)

# Betárolt mennyiség számlálása

- APulse és SPulse felfutó éleit számoljuk
- A betárolt mennyiséget az AIn és SIn változókba töltjük
- A számlálókat kitároláskor nullázzuk

```
ACTION CountAPulse:
```

```
    CntA(CU:=APulse, R:=NOT LOW, CV=>AIn);
```

```
END_ACTION
```

```
ACTION CountSPulse:
```

```
    CntS(CU:=SPulse, R:=NOT LOW, CV=>SIn);
```

```
END_ACTION
```



PROGRAM MAIN

VAR\_INPUT

APulse AT %IX0.0: BOOL;  
Spulse AT %IX0.1: BOOL;  
Low AT %IX0.2: BOOL;  
High AT %IX0.3: BOOL;  
WashBtn AT %IX0.4: BOOL;  
ProdBtn AT %IX0.5: BOOL;  
Temp AT %IB0.0: USINT;

END\_VAR

VAR\_OUTPUT

APump AT %QX0.0: BOOL;  
SPump AT %QX0.1: BOOL;  
WPump AT %QX0.2: BOOL;  
OutPump AT %QX0.3: BOOL;  
Heater AT %QX0.4: BOOL;  
Mixer AT %QX0.5: BOOL;

END\_VAR

VAR

Ain, Sin: INT;  
CntA, CntS: CTU;

END\_VAR

ACTION CountAPulse:  
CntA( CU:=APulse,  
R:=NOT LOW,  
CV=>Ain

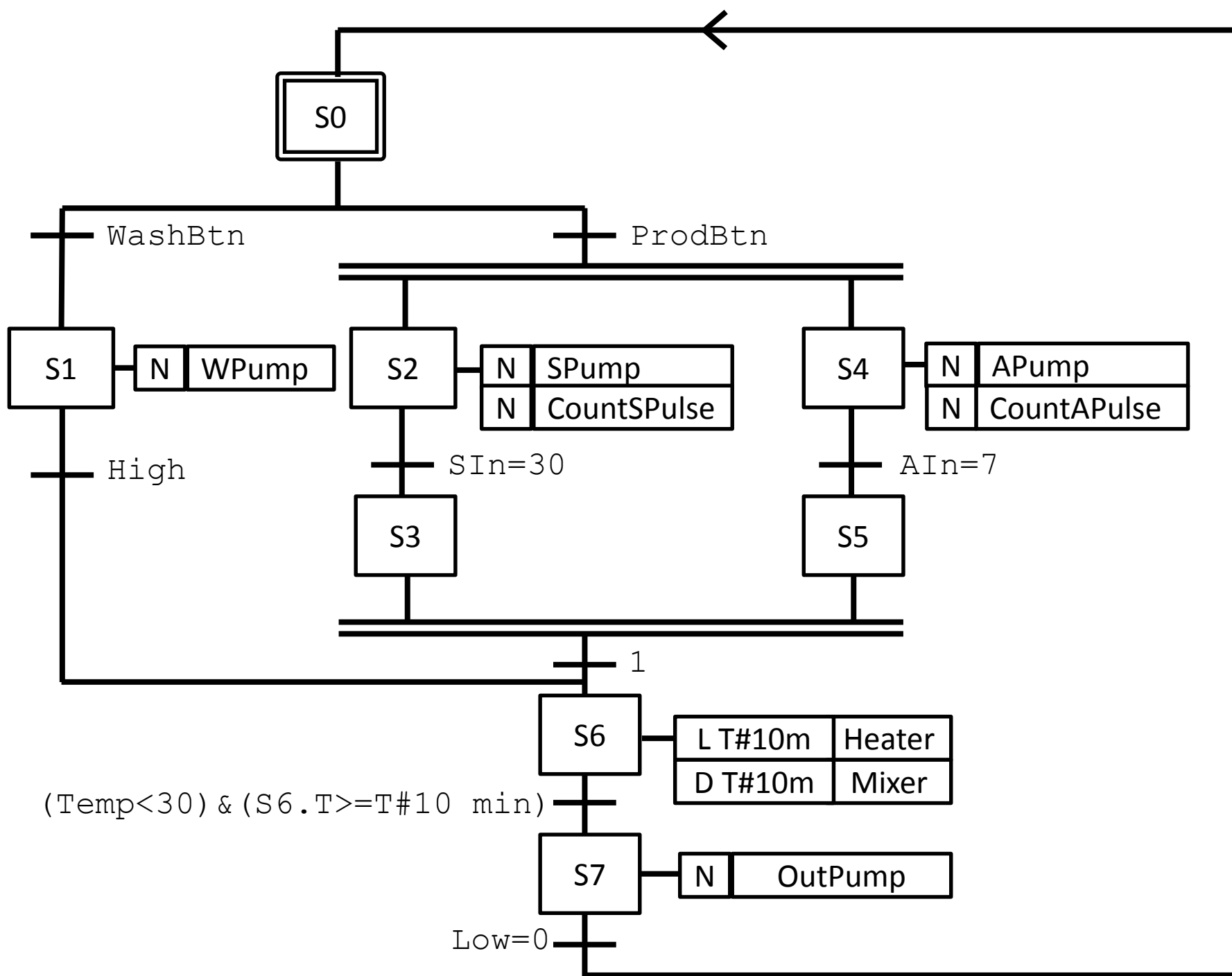
);

END\_ACTION

ACTION CountSPulse:  
CntS( CU:=SPulse,  
R:=NOT LOW,  
CV=>Sin

);

END\_ACTION



# Példa- Akciók

Akció	Magyarázat
Wpump	A mosóvíz szivattyút működtető bit (logikai akció)
Spump	Az oldószer szivattyút működtető bit (logikai akció)
Apump	A hatóanyag-szivattyút működtető bit (logikai akció)
CountSPulse	Betárolt oldószer-mennyiség (SIn) mérése – átfolyásmérő impulzusait számláló akció (nem logikai)
CountAPulse	Betárolt hatóanyag-mennyiség (AIn) mérése – átfolyásmérő impulzusait számláló akció (nem logikai)
Heater	A fűtést működtető bit (logikai akció)
Mixer	A keverőt működtető bit (logikai akció)
OutPump	A kitároló szivattyút működtető bit (logikai akció)