

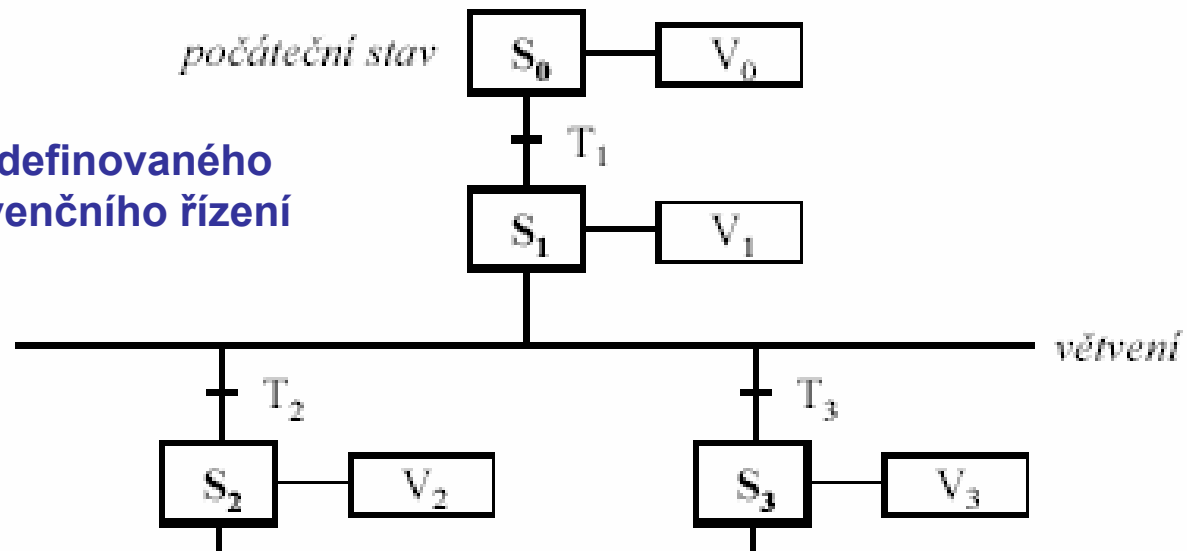
Grafické nástroje pro řešení sekvenčního řízení

Graf přechodů – již v předchozí přednášce, - je grafickým nástrojem vhodnějším pro řešení sekvenčních úloh s obvodem (synchronní a asynchronní LSO)

Funkční diagram sekvenčního řízení – matematické vyjádření grafem s dvěma druhy uzlů: stavový uzel - stav (také krok) a přechodový uzel – přechod, propojení grafu realizují hrany

GRAFCET - standardizovaný grafický nástroj - základ diagram sekvenčního řízení (navržen v letech 1975-1977 francouzskou organizací AFCET a v roce 1987 mezinárodním standardem)

Příklad volně definovaného diagramu sekvenčního řízení



S_i ... stavy

T_i ... logické výrazy definující podmínku přechodu ze stavu **S_{i-1}** do stavu **S_i**

V_i ... logické výrazy definující hodnoty výstupů ve stavu **S_i**

Grafcet - úvod

Grafcet* - grafický návrhový nástroj pro řídicí systémy, popisuje pouze logický automat v matematickém smyslu, nezávisle na technologii a konečné realizaci.

Grafcet vychází z Petriho sítí, které jsou matematickým nástrojem pro modelování systémů diskrétních událostí.

Grafcet je vhodný pro návrh algoritmů řízení PLC (programovatelných logických automatů).

1. Grafcet obsahuje dva základní prvky :

krok a přechod.

2. Každý krok se může vyskytovat ve dvou stavech:

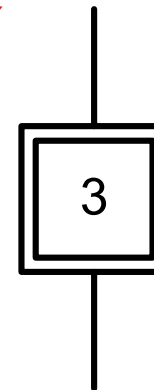
aktivní nebo **neaktivní**

aktivita kroku je znázorněna **značkou** (tečka)

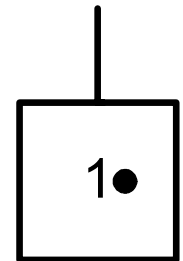
3. Ke kroku lze vázat **akci**, jež je výstupem Grafcetů.

4. K přechodu je vázána **podmínka**, je vstupem Grafcetů

Počáteční
krok



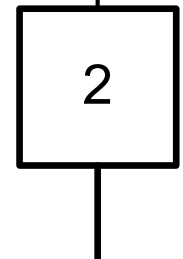
Aktivní
krok



PŘECHOD



Neaktivní
krok



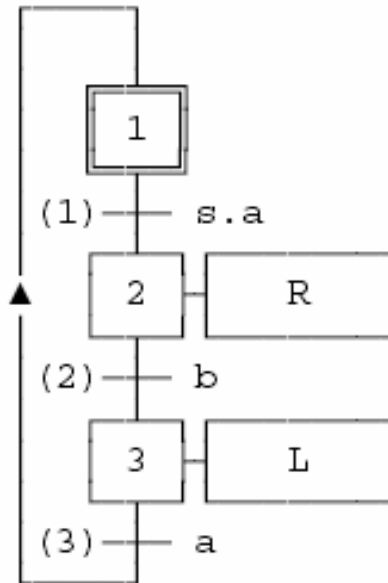
* **Grafcet** ve francouzštině "Graphe Fonctionnel de Connexion Etapes Transitions,,

Příklad 1- řízení pohybu vozíku:

Zadání : V počátečním stavu systému se vozík **T** nachází na kontaktu **a**. Po stlačení tlačítka **s** se pohybuje doprava **R**. Po dosažení kontaktu **b** se vozík vrací ke kontaktu **a** pohybem vlevo **L**.



Řešení:



krok 1 - počáteční krok
kroky 2, 3 - provádí akce pro pohyb
doprava respektive doleva
přechod(1) - je přeskočen, když je krok 1
a obě podmínky (s,a) jsou
splněny
přechody(2) a (3) - jsou vázány na podmínky
(b,a)

Postup návrhu sekvenčního řízení:

1. určíme jednotlivé kroky (stavy) a jejich pořadí v sekvenci,
2. určíme vstupy, pro daný krok významné,
3. vytvoříme logické výrazy definující podmínky pro přechody mezi kroky,
4. pro každý krok vytvoříme logické výrazy, které určují hodnoty výstupů
5. vytvoříme sekvenční diagram řízení a odvodíme logické rovnice pro jednotlivé kroky

Odvození rovnic pro vytvoření řídicího programu PA

Pro realizaci navrženého funkčního diagramu **programovatelným automatem (PA)** => z diagramu odvodit soustavu logických rovnic.

kroky a přechody – budou logickými proměnnými v rovnicích a v programu automatu => **nutné vhodně označit např.:**

K1, K2, K3kroky diagramu

P1, P2, P3 ... přechody diagramu

Pro aktivitu kroku *K2* platí že:

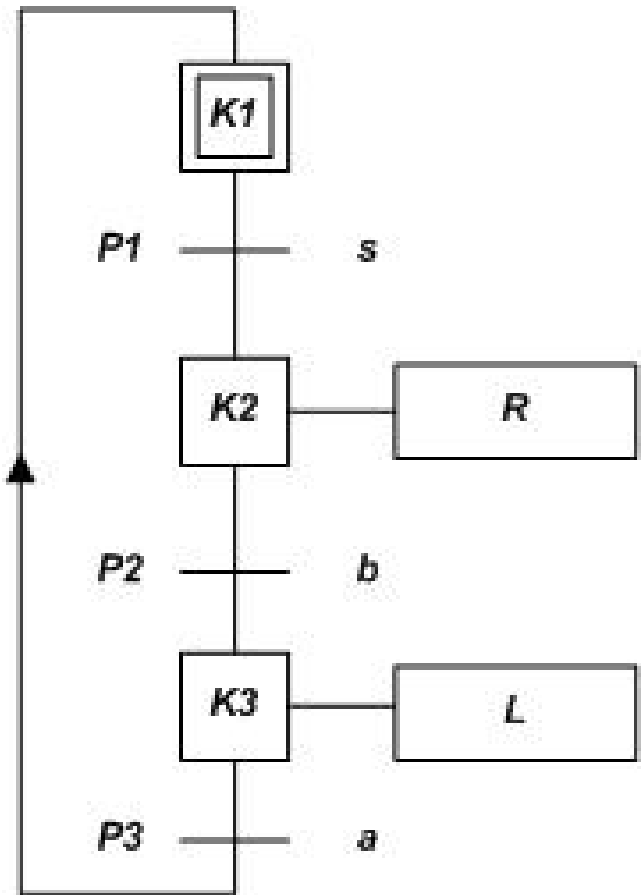
K2 se stane vždy aktivní, jestliže ***K1*** je aktivní a je splněna podmínka přechodu ***P1*** a zůstaneme v něm dokud nebude splněna podmínka ***P2***.

Tato věta charakterizuje aktivitu kroku v čase a **podmínky vstupu do kroku, setrvání v něm a opuštění kroku**. To lze vyjádřit boolskou funkcí pro krok ***K2***:

$$K2 = \underbrace{K1 \cdot P1}_{\text{podmínka vstupu do kroku}} + \underbrace{K2 \cdot \overline{P2}}_{\text{podmínka udržení kroku a výstupu z kroku}}$$

podmínka vstupu do kroku

podmínka udržení kroku
a výstupu z kroku



Pokračování odvozování rovnic

Pro všechny kroky dostaneme tyto logické funkce:

$$K1 = K3 \cdot P3 + K1 \cdot \overline{P1} \quad (1)$$

$$K2 = K1 \cdot P1 + K2 \cdot \overline{P2} \quad (2)$$

$$K3 = K2 \cdot P2 + K3 \cdot \overline{P3} \quad (3)$$

Jestliže krok $K_i = 1$ pak je aktivní, je-li $K_i = 0$ je neaktivní, platí pro $i = \{1,2,3\}$

1. Logické funkce, jejich algebraické výrazy - v PA se provádějí postupně a cyklicky neustále za sebou

2. Provádění - vyčíslování logických výrazů (hodnota pravé strany se přiřadí levé)

3. Start cyklického vyhodnocování funkcí (1) až (3) – Vstup do kroku K1 :

- krok K1 bude aktivní ($K=1$), protože druhý člen funkce (1) je roven 1 (podmínka $P1=0$ nesplněna, $s=0$)
- kroky K2 a K3 budou neaktivní, protože oba členy funkcí (2) a (3) jsou rovny nule
- po změně hodnoty podmínky P1 na jedničku (po načtení hodnoty $s=1$ při otočce cyklu programu) se druhý člen ve funkci stane nulový a tím se deaktivuje krok K1 ($K1=0$).

4. Vstup do kroku K2 a dalších kroků:

- u K2 vzniká problém, -- > v reálných PA se funkce vypočítávají postupně za sebou, ve výpočtu následujících funkcí se uplatní nové hodnoty předcházejících funkcí
- když se krok K1 stane neaktivní ($K1=0$) díky výpočtu funkce (1), nemůže se vzápětí „vstupní“ člen funkce (2) nastavit na 1 a nedojde tedy k aktivaci kroku K2.
- problém lze obejít tím, že zajistíme aby se aktivace následujícího kroku provedla vždy před deaktivací předcházejícího kroku
- toho lze dosáhnout tak, že deaktivaci (opuštění) kroku nebudeme podmiňovat podmínkou přechodu, ale aktivací následujícího kroku.

Funkce (1) až (3) sestavené podle tohoto pravidla mají tvar:

$$K1 = K3 \cdot P3 + K1 \cdot \overline{K2} \quad (1)$$

$$K2 = K1 \cdot P1 + K2 \cdot \overline{K3} \quad (2)$$

$$K3 = K2 \cdot P2 + K3 \cdot \overline{K1} \quad (3)$$

- funkce zabezpečí --> při provedení přechodu aktivuje následující krok dříve, než se opustí předcházející
- během jednoho cyklu programu - dva kroky aktivní současně několik jednotek až desítek milisekund, to v naprosté většině případů řízení technologických procesů nevadí

4. Přiřazení výstupních akcí, které má automat provádět:

- diagram předepisuje akce, akce spočívají v generování logických jedniček u předepsaných výstupních proměnných během aktivity příslušných kroků.
- pro každou výstupní proměnnou stačí připojit jednu funkci, jejíž definiční výraz bude tvořen logickým součtem všech kroků při nichž má mít tato proměnná hodnotu jedna.

Pro uvedený příklad řízení vozíku doplníme logické rovnice (1) až (3) definičními výrazy pro výstupní proměnné:

$$R = K2 \quad L = K3$$

5. Kódování rovnic (proměnných) podle syntaxe zvoleného PA (LOGO Siemens)

- proměnné K_i jsou fiktivní, v PA budou realizovány jako vnitřní proměnné M_i (merkery)
- po zapnutí PA všechny jeho $M_i = 0$, v našich rovnicích se žádný krok tak nestane aktivní, ale počáteční krok musí být aktivní → řešení přidání negované pomocné vnitřní proměnné M (**INIC**) do výrazu funkce (1): $K1 = K3 \cdot P3 + K1 \cdot /K2 + /INIC$
- vstupní proměnné jsou v PA označovány (kódovány) I_i a výstupní proměnné Q_i , potom:

Připojíme-li například u PA LOGO 24/12 vstupy a výstupy takto :

Vstupy: $a = I_1$, $b = I_2$, $s = I_3$

; signál a připojen na vstup I_1 .. atd.

Výstupy: $R = Q_1$ $L = Q_2$; signál R generován na výstupu $Q1$..atd.

a přiřadíme kroky $K1$ až $K3$ i proměnnou $INIC$ konkrétním vnitřním proměnným Mi následovně:

$$K1 = M1, \quad K2 = M2, \quad K3 = M3, \quad INIC = M4$$

=> získáme konečné rovnice pro program PA, které je možno přepsat do formy blokových nebo kontaktních schémat, eventuelně do jiné formy – například textové.

$$M1 = M3 \cdot I1 + M1 \cdot \overline{M2} + \overline{M4} \quad (1)$$

$$M2 = M1 \cdot I3 + M2 \cdot \overline{M3} \quad (2)$$

$$M3 = M2 \cdot I2 + M3 \cdot \overline{M1} \quad (3)$$

$$Q1 = M2 \quad (4)$$

$$Q2 = M3 \quad (5)$$

$$M4 = 1 \quad (6); \text{ po průchodu prvním cyklem je nutné vliv } INIC \text{ zrušit, jinak by nebyly}$$

v dalších cyklech funkční zbývající členy výrazu pro $M1$