Aplikace kybernetiky ve strojírenství

10. přednáška

Doc. Ing. Eduard Janeček, CSc.

Ing. Jan Jakl

Podpořeno v rámci projektu CZ.1.07/2.2.00/15.0383
Inovace studijního oboru Dopravní a manipulační technika
s ohledem na potřeby trhu práce

Programovatelné logické automaty – PLC (Programmable Logic Controller)

- programovatelný řídicí systém umožňující řízení v reálném čase
- již od první poloviny 80. let
- původně jako náhrada za reléové obvody
- výrobci – Allen Bradley, Siemens, Teco, Omron,…

zdroj: Allen Bradley, www.ab.com

Rozdělení PLC

- kompaktní – PLC s pevně daným počtem vstupů a výstupů.
- modulární – možnost připojit k PLC další moduly (vstupní, výstupní, vstupně/výstupní).

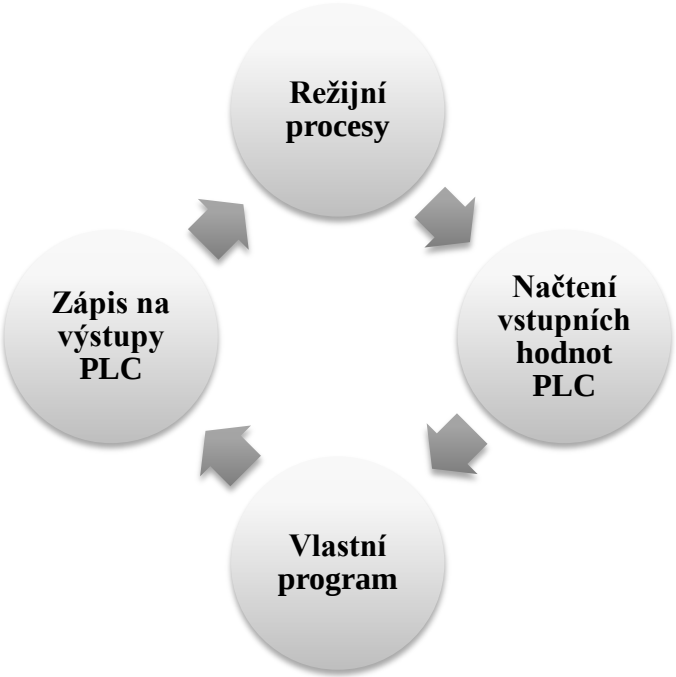
zdroj: www.shahelectric.com

zdroj: Allen Bradley, www.ab.com

Dále lze PLC rozdělit podle počtu vstupů a výstupů na mikro (méně než 10 I/O), malé (desítky I/O), střední (stovky I/O) a velké (I/O v řádech tisíců).

Vykonávání programu

Charakteristickým rysem PLC je, že program je prováděn cyklicky.



Mezi režijní procesy patří aktualizace proměnných, plánování aktivace jednotlivých procesů atd…

Programování PLC

Metody programování PLC definuje norma IEC 61131-3. Sjednocení programovacích metod umožňuje používat stejné kódy i na odlišných PLC.

Možnosti programování PLC

Textově

- seznam instrukcí (IL)
- strukturovaný text (ST)

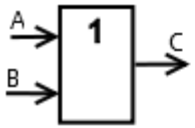
Graficky

- diagramy funkčních bloků (FBD)
- liniová schémata (LD)
- sekvenční funkční graf (SFC)

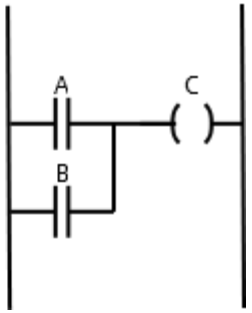
IL:
```
ld   A
or   B
st   C
```

ST:
```
C:= A or B
```

FBD:



LD:



Společné prvky programovacích jazyků

Programy pro PLC se skládají z *jednoduchých prvků.* Mezi jednoduché prvky patří

- oddělovače – speciální znaky, např. (,), -, :, mezera
- identifikátory – slouží pro názvy proměnných, funkcí, funkčních bloků. Je možné používat písmena, číslice a podtržítko, např. Pom, Pom_1, Tlacitko_On, Pohyb_Nahoru,…
- literály – hodnoty proměnných, např. 1; 5.6; 1.1E5; 2#1001; 16#E3; FALSE; TRUE; ‘hodne’; ‘malo’; D#2011-01-10 (aktuální datum), TOD#11:22:33 (aktuální čas),…
- klíčová slova – nesmějí se používat jako názvy proměnných, např. FUNCTION, VAR_OUTPUT, REAL,…
- komentáře – obecné, uzavřené mezi znaky (* a *)
  – řádkové, začínají znaky // a končí na konci řádku

Identifikátory

Identifikátory musí začínat písmenem nebo podtržítkem a nemohou obsahovat mezery. Programy nerozlišují velká a malá písmena v identifikátorech, tj. identifikátory POM, Pom a pom ukazují na tutéž proměnnou.

Datové typy

- BOOL (Boolean) – 1 bit, hodnoty 0 a 1
- SINT (Short Integer) – 8 bitů, hodnoty -128 až 127
- INT (Integer) – 16 bitů, hodnoty -32768 až 32768
- DINT (Double Integer) – 32 bitů, hodnoty -2^{31} až 2^{31}
- USINT (Unsigned Short Integer) – 8 bitů, hodnoty 0 až 255
- UINT (Unsigned Integer) – 16 bitů, hodnoty 0 až 2^{16}
- UDINT (Unsigned Double Integer) – 32 bitů, hodnoty 0 až 2^{32}
- REAL (Real) – 32 bitů, ±2.9E-39 až ±3.4E+38
- TIME (doba trvání)
- DATE (datum)
- TIME_OF_DAY, TOD (denní čas)
- DATE_AND_TIME, DT (absolutní čas)
- STRING – textový řetězec, max 255 znaků
- BYTE – 8 bitové slovo

- WORD – 16 bitové slovo
- DWORD – 32 bitové slovo
- LWORD – 64 bitové slovo
- WSTRING – textový řetězec v 16 bitovém kódu, max 65535 znaků

Mezi další datové typy se řadí tzv. odvozené datové typy. Mezi ně patří například datové typy ARRAY (pole) a STRUCT (struktura).

ARRAY

ARRAY[I_1..I_2] OF *DATOVÝ TYP*

P0	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**

Pole je tvořeno z několika prvků stejného datového typu. Pozice prvků je číslována od indexu I_1 a k jednotlivým prvkům lze přistupovat. Syntaxe je taková, že za slovem ARRAY následuje v hranatých závorkách dvojice čísel určující nejnižší a nejvyšší index v daném poli (tj. jeho velikost), poté následuje slovo OF, za kterým je uvedeno jakého datového typu mají být jednotlivé elementy daného pole. Dané pole lze také při jeho vytvoření naplnit hodnotami. To se provádí tak, že za datovým typem následuje := a do hranatých závorek se zapíší jednotlivé prvky pole.

Příklad:

pole1 : ARRAY [0..5] OF INT – vytvoří proměnnou pole1 typu ARRAY s 6 prvky indexovanými od 0 do 5. Všechny prvky jsou typu INT.

pole2 : ARRAY [1..8] OF INT := [1, 2, -4, -5, 14, -22, 30, -3] – vytvoří datovou proměnnou pole2 typu ARRAY s 8 prvky indexovanými od 1 do 8. Prvky v poli jsou typu INT a mají zadané hodnoty.

V případě potřeby lze pracovat také s vícerozměrnými poli. Dále je možné definovat vlastní odvozené datové typy.

Proměnné

Proměnné mohou být trojího typu: vstupní, výstupní a vnitřní. Dále můžeme proměnné rozdělit podle jejich působnosti

- globální proměnné (zálohované, nezálohované, konstanty, externí)
- lokální proměnné (lokální, přechodné)
- proměnné pro předávání parametrů (vstupní, výstupní, vstupně/výstupní)

Deklarace globálních proměnný

```
VAR_EXTERNAL
        pom1            : WORD      // externí globální proměnná
END_VAR

VAR_GLOBAL
        jede            : BOOL
        teplota         : REAL
END_VAR
```

Globální proměnné jsou viditelné mezi všemi částmi daného programu (podprogramy).

Deklarace lokálních proměnných

```
PROGRAM PrikladProgramu
    VAR
        pom1    : INT;
        pom2    : INT;
        pom3    : INT;
    END_VAR
    pom3 := pom1 + pom2;
END_PROGRAM
```

Funkce

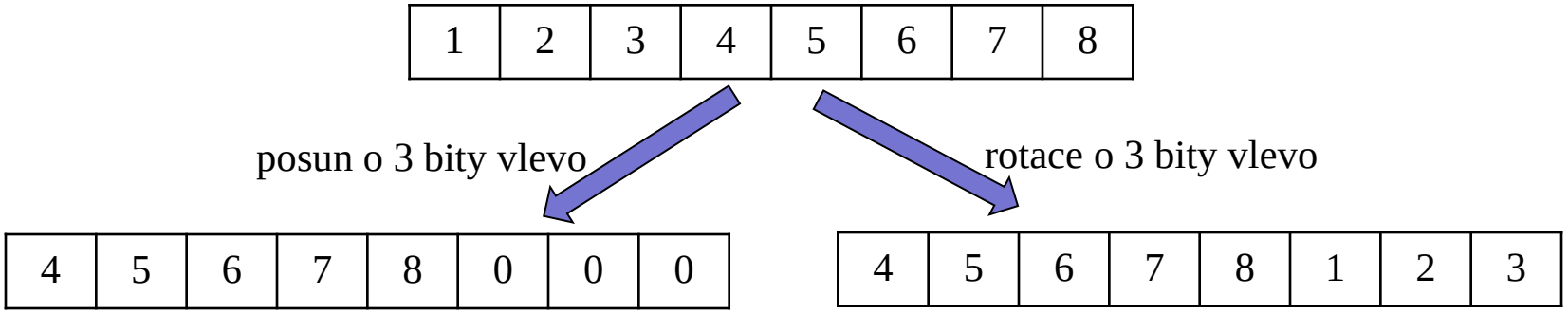
Stejně jako v případě jiných programovacích jazyků (např. Java) slouží funkce k vykonání určité činnosti a vrací jednu hodnotu. Funkce se mohou používat také jako součásti výrazů

např. pom := pom1+SIN(pom2).

Standartní funkce

- Typová konverze – obecně *datový typ vstupu_TO_datový typ výstupu*, např. INT_TO_REAL, dále pak funkce TRUNC (vstupem je REAL a výstupem INT, oříznutí)
- Aritmetické – sčítání ADD (i pro více než 2 vstupy), rozdíl SUB, součin MUL (i pro více než 2 vstupy), podíl DIV, modulo MOD, mocnina EXP, přiřazení MOVE
- Matematické – absolutní hodnota ABS, odmocnina SQRT, přirozený logaritmus LN, desítkový logaritmus LOG, přirozená exp. funkce EXP, sinus SIN, kosinus COS, tangens TAN, arcus sinus ASIN, arcus kosinus ACOS, arcus tangens ATAN, u goniometrických funkcí musí být vstupní hodnota v radiánech.
- Bitové posuny – posun vlevo o N bitů SHL(vstupní proměnná, N), posun vpravo o N bitů SHR(vstupní proměnná, N), rotace vlevo o N bitů ROL(vp, N), rotace vpravo o N bitů ROR(vp, N).

Rozdíl mezi rotací a posunem



Další standartní funkce

- Logické – logický součin AND, logický součet OT, negace NOT,…
- Porovnávací – větší GT, menší LT, rovno EQ, větší nebo rovno GE, menší nebo rovno LE, ne rovno NE
- Textové řetězce – délka řetězce LEN, vložení jednoho řetězce do druhého INSERT(V1, V2), prvních N znaků řetězce zleva LEFT(V1, N), zprava RIGHT(V1, N), prvních N znaků řetězce od pozice M MID(V1, N, M),….

Programování PLC - Seznam instrukcí IL

- jedná se o nízkoúrovňový programovací jazyk (podobný assembleru)
- vhodný pro malé aplikace
- základní syntaxe

návěští:	operand	operátor	komentář

Návěští a komentář jsou nepovinné. Program vykonává jednotlivé instrukce po sobě a aktuální výsledek si uchovává v (průběžné) proměnné.

Příklad

```
LD  A
AND       B
```

v aktuální výsledku je hodnota výsledek:=A AND B

```
OR C
```

v aktuální výsledku je hodnota výsledek:=výsledek OR C.

Mezi použitelné funkce patří ty, které byly uvedeny. Některé funkce pak lze doplnit o modifikátory N a (.

Příklad použití modifikátoru N

ANDN B – výsledek := výsledek AND NOT B, tj. dochází k negaci operandu B

Příklad použití operátoru (- tzv. odložený start

OR (A

AND B

) - výsledek := výsledek OR (A AND B)

Mezi další funkce, které lze použít patří ST (uloží výsledek na místo operandu), S (jestliže je aktuální výsledek 1, nastaví operand na hodnotu 1), R (jestliže je aktuální výsledek 1, nastaví operand na 0). Dále lze používat funkce pro skoky a volání. Ty mohou mít další modifikátor C, který značí, že se daná operace provede pouze pokud je aktuální výsledek 1. Do kategorie funkcí pro skoky a volání patří“ JUMP (skok na návěští), JUMPN (skok na návěstí pokud je akt. výsledek 0), JUMPC (skok na návěstí pokud je akt. výsledek 1), CAL (volání funkčního bloku), RET (návrat z funkce nebo funkčního bloku), Název_funkce (zavolá danou funkci, bez modifikátorů).

Příklad programu v IL

```
FUNCTION Test: INT
    VAR_INPUT
        StranaA     : INT
        StranaB     : INT
        C           : BOOL
    END_VAR
    VAR_OUTPUT
        Vysledek    : BOOL
    END_VAR
        LD      C           // nastaví hodnotu C jako aktuální výsledek
        JMPC    OBSAH       // pokud je C = 1 skoč na navěští OBSAH
        JMP     OBVOD       // skoč na návěští OBVOD
OBSAH:  LD      StranaA     // do výsledku se nahraje hodnota proměnné StranaA
        MUL     StranaB     // výsledek := výsledek * StranaB
        JMP     KONEC       // skoč na návěští KONEC
OBVOD:  LD      StranaA     // do výsledku se nahraje hodnota proměnné StranaA
        ADD     StranaB     // výsledek := výsledek + StranaB
        MUL     2           // výsledek := výsledek * 2
KONEC:  ST      Vysledek    // aktuální výsledek se uloží do proměnné Vysledek
END_FUNCTION
```

Programování PLC – Strukturovaný text ST

Jedná se o vyšší programovací jazyk, který má základy v C a Pascal. Na rozdíl od IL obsahuje také příkazy pro rozhodování (IF – THEN – ELSE, CASE OF) a také příkazy pro smyčky (FOR, WHILE, REPEAT), nelze však používat skoky.

Mezi příkazy, které lze v ST používat jsou

- závorky: (výraz), např. (A+B), slouží ke stanovení pořadí při výpočtu
- matematické funkce: LOG(A), SIN(B), MIN(A,B), SQRT(A),…
- negace: -
- doplněk: NOT
- mocnina: **, např. A**B = A^B, lze použít také EXPT(A,B)
- sčítání, odčítání, násobení, dělení: +, -, *, /
- porovnání: <, >, <=, >=, = (rovnost), <> (nerovnost)
- logický součin, součet, xor, -, & nebo AND, OR, XOR

Příklady přiřazení:

A:=B; A = COS(X); A = A + 2;

Rozhodování

```
D:= B**2 – 4*A*C
IF D >= 0 THEN
    X1:= (-B + SQRT(D))/(2*A);
    X2:= (-B - SQRT(D))/(2*A);
ELSE
    X1:= 0;
    X2:= 0;
END_IF

A = 1;
CASE A OF
1 : B = 1;
2 : B = 2;
END_CASE
```

Cykly

```
j = 0;
FOR i:= 1 TO 10 (možnost nastavit hodnotu inkrementu přidáním BY hodnota inkrementu)
DO
      j := j + i;
END_FOR   // cyklus FOR provádí dané operace po přesný počet iterací

j := 0;
WHILE j<10 DO
      j := j + 1;
END_WHILE    // cyklus WHILE provádí dané operace pouze pokud je splněna vstupní podmínka

j := 0;
REPEAT
      j := j + 1;
UNTIL j < 10
END_REPEAT  // podobné jako cyklus WHILE, rozdíl je ten, že podmínka je na konci cyklu, tj. tělo cyklu se
                  //  provede alespoň jedenkrát.
```

Příkazem EXIT je možné průběh cyklu předčasně ukončit.

```
j := 1;
FOR i:=1 TO 100
    DO
    j:=j*i;
    IF      j>1000 THEN
            EXIT
    ELSE
            ; // prázdný příkaz
    END_IF
END_FOR
```

Programování PLC – Liniová schémata (LD, Ladder Diagram)

Jedná se o ekvivalent k reléovým schématům a k jeho základnímu využití patří vyhodnocení logických výrazů.

Základními prvky LD:

- napájecí sběrnice
- spojnice
- kontakty a cívky
- prvky pro řízení provádění programu
- prvku pro volání funkcí nebo funkčních bloků

Typy kontaktů:

- spínací
- rozpínací
- kontakt s detekcí náběžné hrany
- kontakt s detekcí sestupné hrany

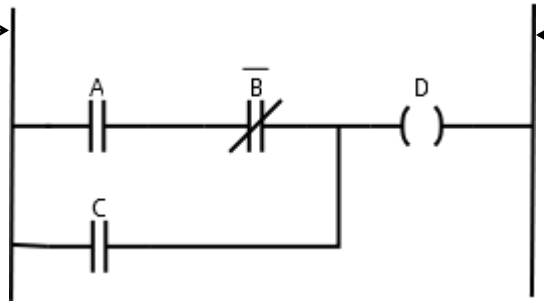
Typy cívek:

- normální cívka
- negovaná cívka
- nastavující cívka (cívka s pamětí)
- resetující cívka
- cívka s detekcí náběžné hrany
- cívka s detekcí sestupné hrany

Elementární prvky LD mohou být propojovány sériově nebo paralelně.

levá napájecí sběrnice →

- stav levé napájecí sběrnice je vždy ON, 1
- stav pravé napájecí sběrnice závisí na aktuálních hodnotách zapojených kontaktů



$$D = \left(A \wedge \bar{B}\right) \vee C$$

← prává napájecí sběrnice

- kontakty A a B jsou v sériovém zapojení, kontakt B je rozpínací (tj. je sepnut když B je 0)
- kontakt C je v paralelním zapojení vzhledem ke kontaktům A a B

Programování PLC – Diagramy funkčních bloků (FBD)

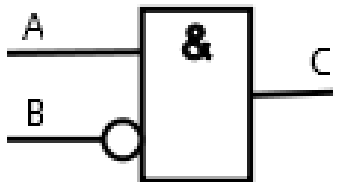
V tomto programovacím jazyce je program konstruován propojováním funkčních bloků.

Rozdíl mezi FBD a LD je v tom, že v případě LD se mezi jednotlivými bloky přenáší pouze hodnoty 0/1 (kontakt je rozepnut kontakt je sepnut), zatímco v FBD se mohou přenášet libovolné hodnoty.

Při sestavování programu pomocí FBD platí následující pravidla

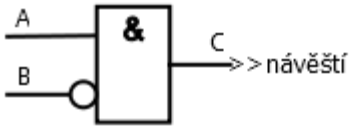
- vstupy bloků musí být připojeny
- datové typy vstupní proměnné funkčního bloku a připojené proměnné se musí shodovat (tj. pokud je vstupní proměnná FB typu BOOL nemůže mít na vstupu proměnou typu STRING).
- propojení FB je orientované zleva doprava
- informace z levého zakončení je může větvit

V případě potřeby lze na vstup přivést negovanou hodnotu vstupní proměnné. K tomu se používá označení formou kolečka na vstupu FB – pozor, ne všechny PLC toto umožňují.



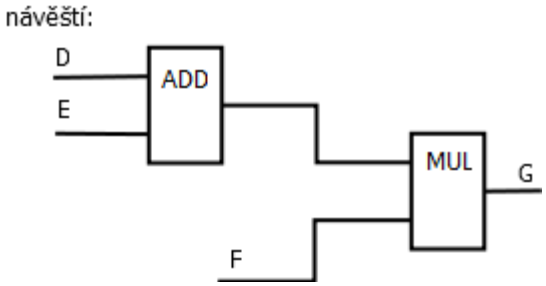
$$C = A \wedge \bar{B}$$

Možnost používání skoků (platí také v LD)



Tomuto FBD odpovídá v IL

LD	A
ANDN	B
JMPC	návěští

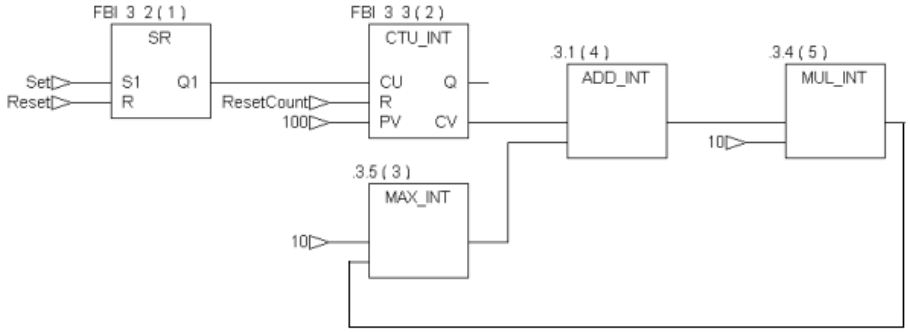


návěští:

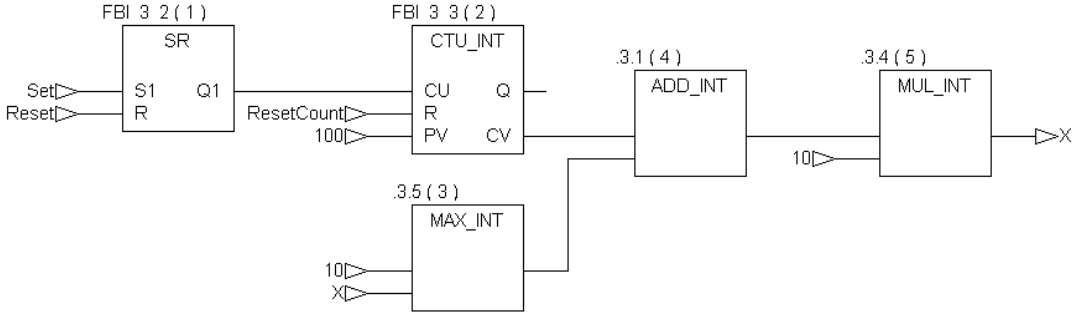
LD	D
ADD	E
MUL	F
ST	G

Zpětnovazební zapojení

Explicitní zpětná vazba – výstup bloku MUL_INT závisí na výstupu MAX_INT a současně je jeho vstupem

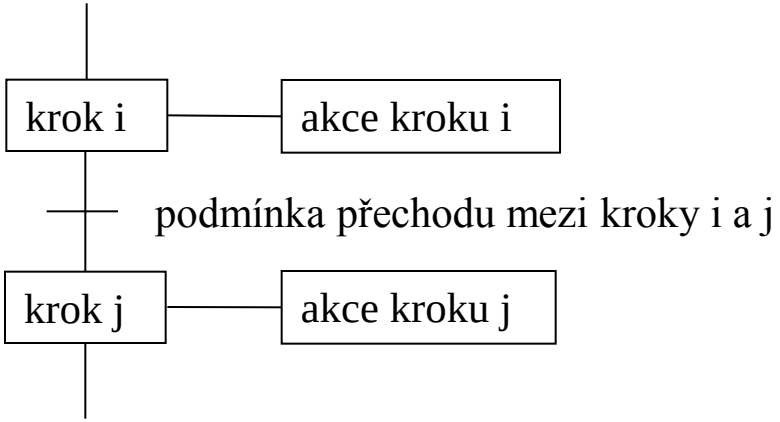


Implicitní zpětná vazba – řeší předchozí problém

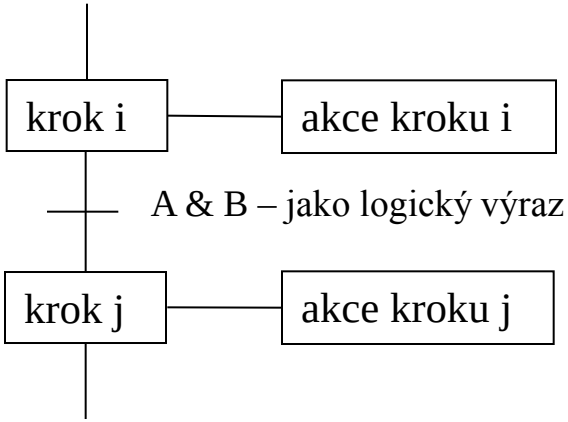


Programování PLC – Sekvenční funkční grafy (SFC)

Tato metoda je vhodná pro detailní analýzu programu a je vhodné ji kombinovat s dalšími metodami programování PLC, zejména s ST. Umožňuje dekompozici složitého problému na jednoduché kroky (stavy) a přechody mezi nimi. V každém kroku je pak určeno několik akcí, které se mají v daném kroku vykonat a zároveň jsou definovány podmínky přechodů mezi jednotlivými kroky.



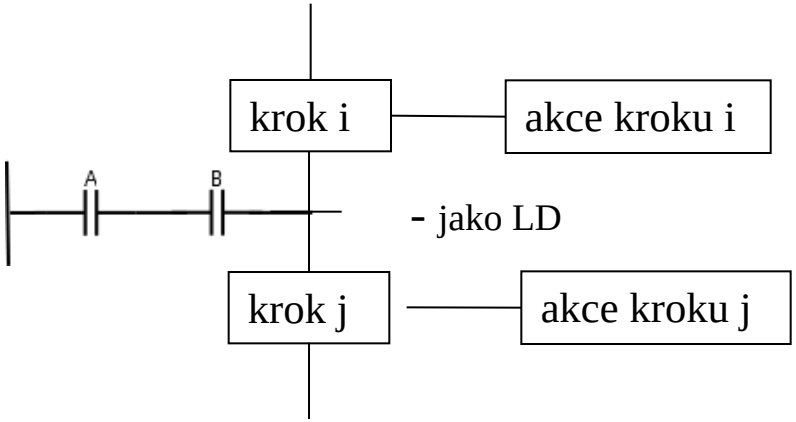
Možnosti jak definovat podmínku přechodu

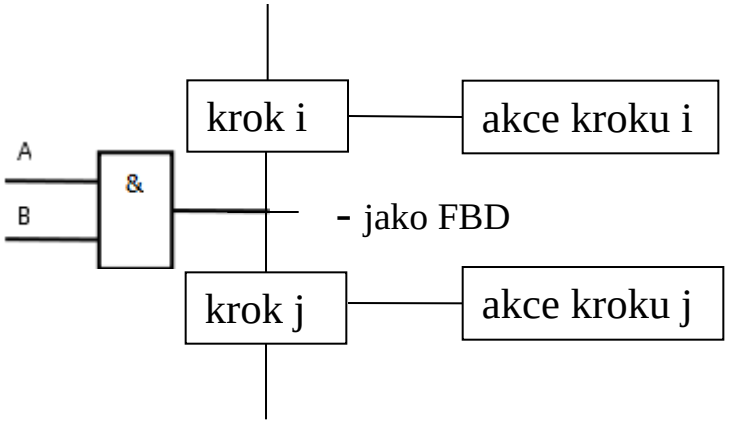


Standardně se používá ST, některé typy PLC umožňují definovat podmínky přechodu také pomocí LD a FDB

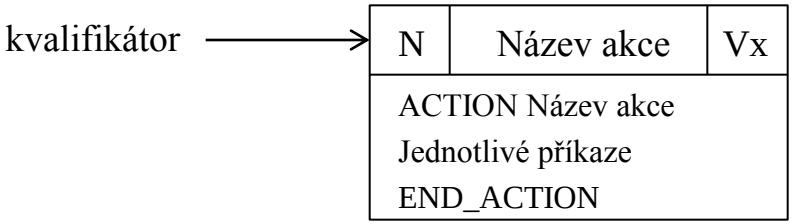
A & B – jako logický výraz v ST, program opustí krok i a přesune se do kroku j až bude splněna podmínka

$$A \& B = 1$$



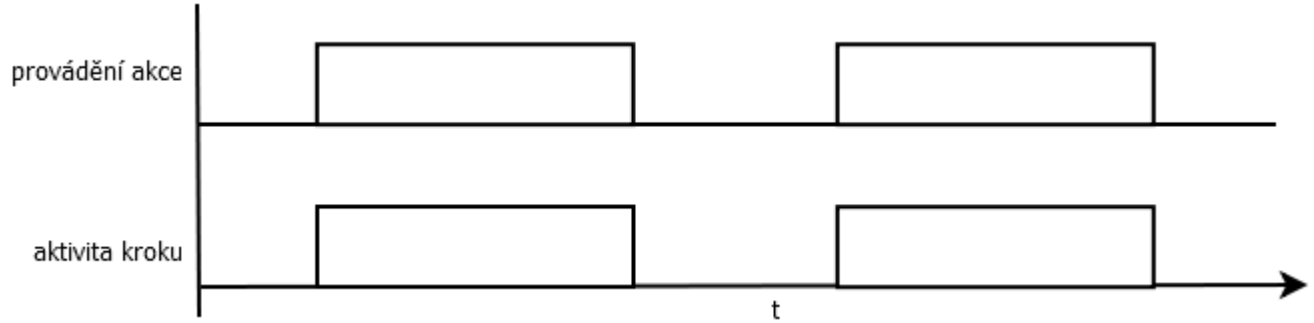


K danému kroku může být definována žádná, jedna nebo více akcí. Pokud není definována žádná akce, je daný krok chápán jako čekací do doby, než je splněna podmínka jeho opuštění.
Akce je definována svým názvem a tělem, které určuje jaké příkazy se mají v rámci dané akce vykonávat. Tělo akce přitom může být programováno pomocí SFC nebo některým z ostatních programovacích jazyků pro PLC.

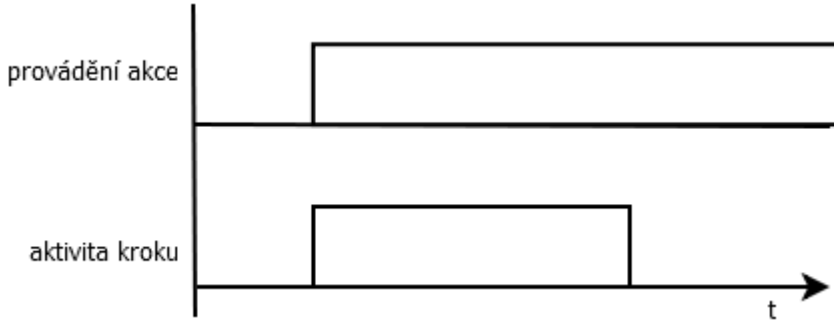


Typy akcí

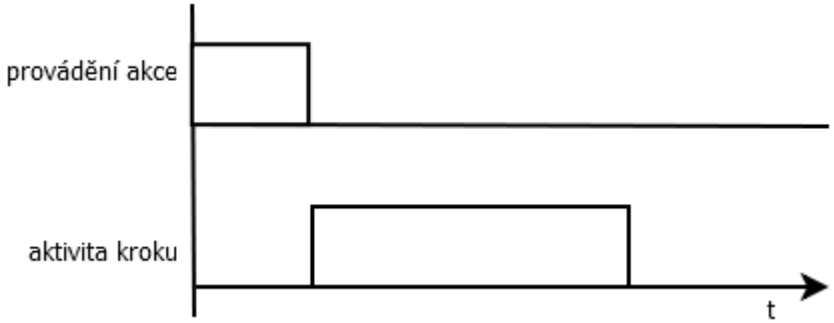
- N nebo žádný – provádí se, když je daný krok aktivní



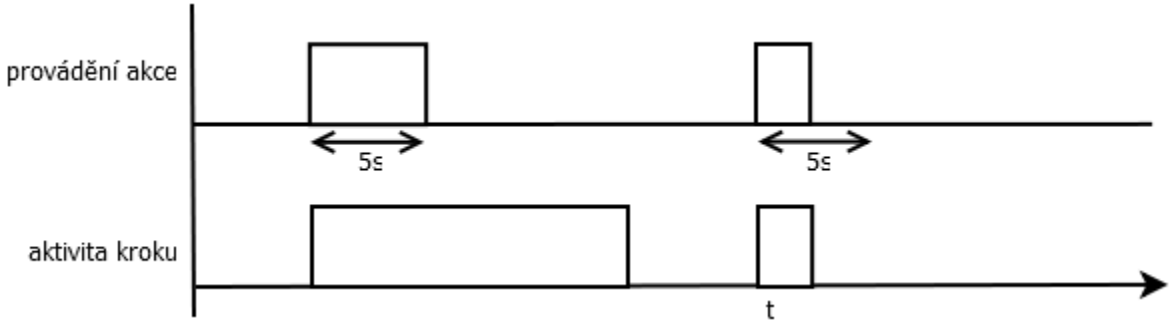
- S – provádí se, dokud není dosažen stav, ve kterém má daná akce kvalifikátor R



- R – ukončuje provádění akcí s kvalifikátorem S, SL a SD

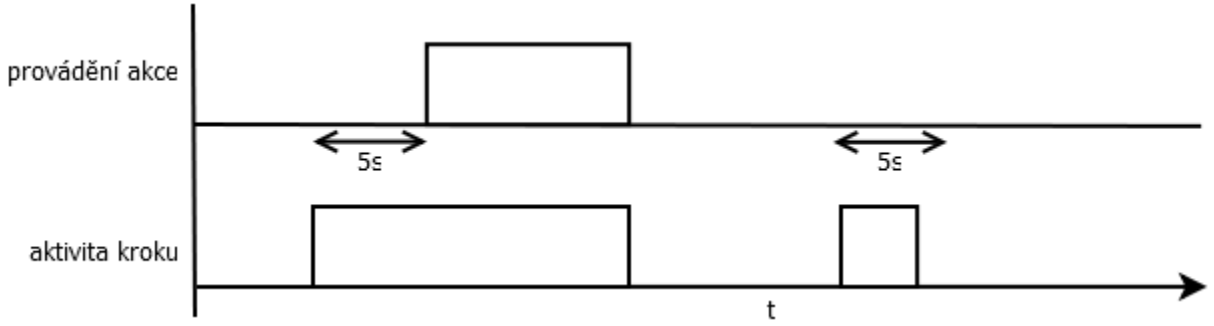


- L – Akce se provádí pouze po určitou dobu, např. t#5s



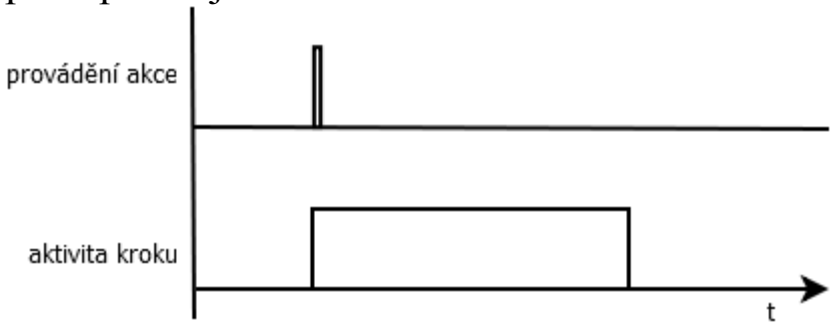
akce je ukončena pokud uplynula doba k jejímu spouštění, nebo pokud přešel daný krok do neaktivního stav.

- D – akce se začne provádět až za určitou dobu



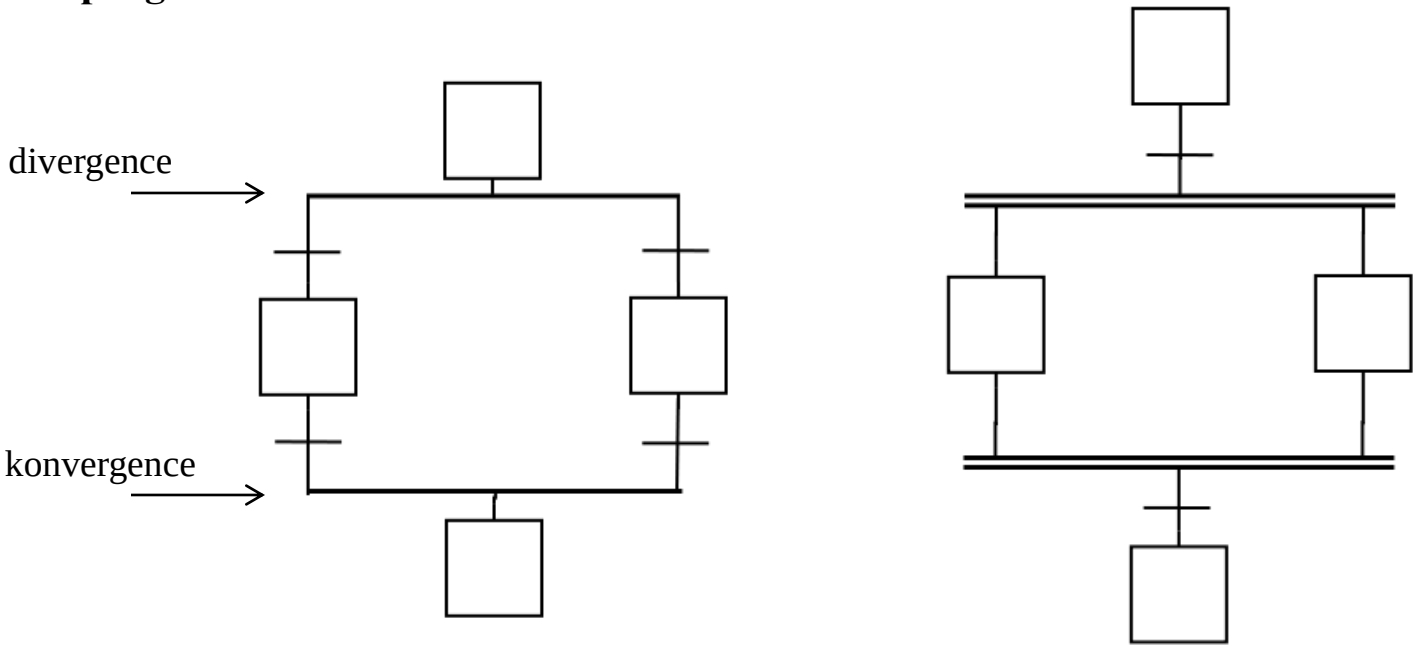
jestliže během zpoždění dojde ke změně aktivity kroku, ke kterému akce náleží, nebude akce po uplynutí doby D spuštěna.

- P – pulsní, akce se spustí pouze jednou



Dále existují kombinace předchozích kvalifikátorů, např. SD, SL, a dále P1 (pulsní, detekuje náběžnou hranu, stejné jako P) a P0 (pulsní, detekuje sestupnou hranu, tj. akce se spustí pouze jednou na konci aktivity kroku).

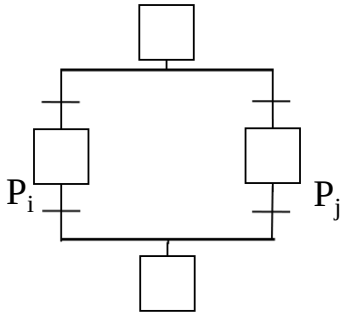
Větvení programu



Divergence – spojení jedné části diagramu (kroku nebo přechodu) a několika částí odlišného typu.

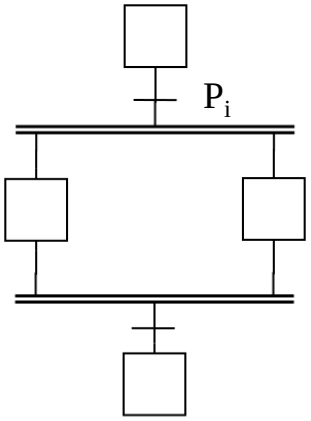
Konvergence – spojení více částí jednoho typu a jedné části odlišného typu.

Větvení typu OR



V případě větvení typu OR (nebo) je prováděna pouze jedna z větví programu. Podmínky přechodu Pi a Pj musí být disjunktní (tj. pokud je platná první musí být druhá neplatná a naopak). Tento požadavek lze zobecnit také pro více větví.

Větvení typu AND



V případě větvení typu AND (a) jsou po splnění podmínky přechodu P_i prováděny všechny větve programu. Procesy v daných větvích se vykonávají paralelně – mluvíme o současně běžící sekvenci.

Zdroje a doporučená literatura

- Programování PLC podle normy IEC 61 131-3 v prostředí Mosaic, 2007, dostupné na http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00321_01_Mosaic_ProgIEC_cz.pdf
- P. Balda: elektronické učební texty z předmětu IŘS1, ZČU v Plzni, 2009
- F. Zezulka a kol.: Programovatelné automaty, VUT v Brně, 2003, dostupné na http://www.vaeprosys.cz/Dokumentace/Programovatelne_automaty/Programovatelne_automaty-Skripta_FEKT_VUT_Brno.pdf
- J. Koziorek, L. Chromčák: Logické systémy řízení, učební text, VŠB – TU Ostrava, 2007, dostupné na http://www.elearn.vsb.cz/archivcd/FEI/LSA/Logicke%20systemy%20rizeni.pdf
- WWW stránky Allen Bradley

Poděkování

Tento projekt je spolufinancován
Evropským sociálním fondem a státním rozpočtem České republiky

Projekt CZ.1.07/2.2.00/15.0383
Inovace studijního oboru Dopravní a manipulační technika
s ohledem na potřeby trhu práce