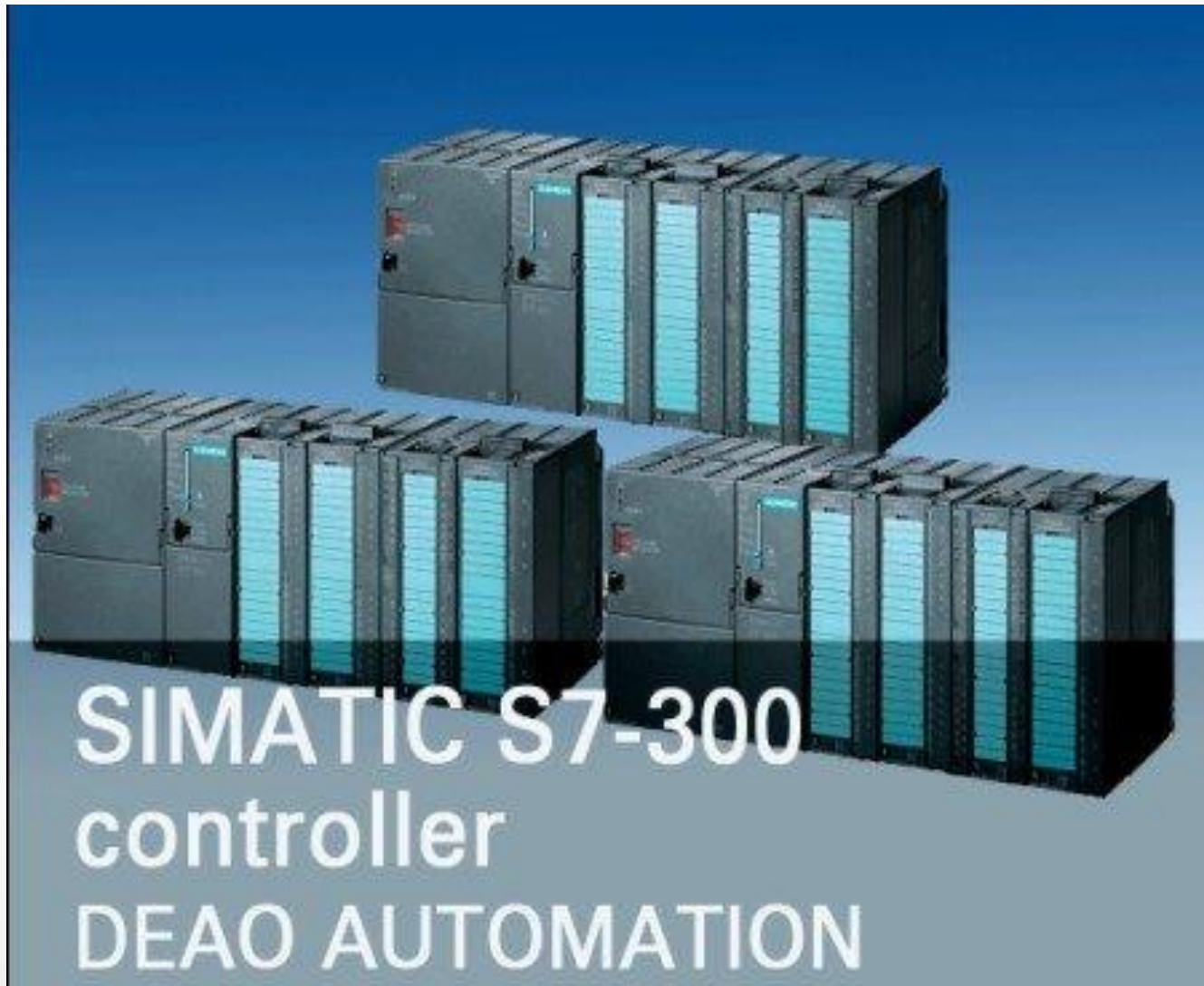


# Názvy programovacích jazyků IEC 61131-3

## PLC podle normy IEC 61131-3



# Programovacích jazyky podle standardu IEC 61131-3

- Jazyky grafické (LAD, FBD, SFC)
- Jazyky textové (IL, ST)

*Poznámka:*

*„Typové zkratky jsou uvedeny v němčině a v angličtině“*

# Grafické jazyky

- **obecné symboly** společné jazyku LAD, FBD, a případně i SFC (vodorovné a svislé úsečky, kolmé úsečky s propojením a bez něj, rohové útvary, **bloky** a konektory)
- **speciální symboly**
- **příčky, sítě (networks)** a pravidla jejich vyhodnocování

# Textové jazyky

- **obecné symboly**, např.
  - používání tištěných znaků
  - vnější reprezentace dat
  - datové typy
  - proměnné
  - jednotky pro organizaci programu
  - elementy pro strukturování programu ...
- **speciální symboly**

# Jazyk příčkového diagramu

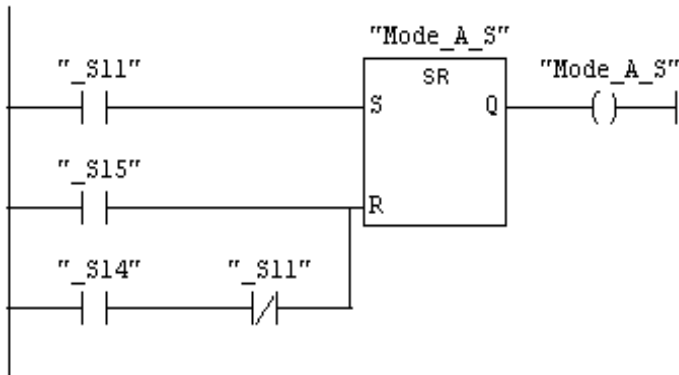
## Ladder Diagram – LAD

- reléové schéma , liniové schéma, žebříčkové schéma, kontaktní schéma
- reléový diagram, liniový diagram, žebříčkový diagram, příčkový diagram, reléový příčkový diagram
- kontaktní plán

# Příklad programu v jazyku příčkového diagramu, LAD

## Network 15 : Start/Stop Auswertung

Comment:



*Zápis \_ pomocí \_ logického \_ výrazu:*

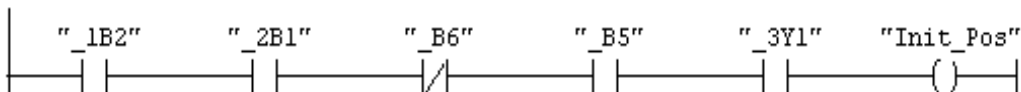
$$S(\text{Mode\_A\_S}) = \_S11$$

$$R(\text{Mode\_A\_S}) = \_S15 + \_S14 * \overline{\_S11}$$

$$\text{Mode\_A\_S} = Q(\text{Mode\_A\_S})$$

## Network 16 : Grundstellung

Comment:



$$\text{Init\_Pos} = \_1B2 \_2B1 \overline{\_B6} \_B5 \_3Y1$$

# Speciální symboly v jazyku LAD

## Vstupy:

- levý a pravý vodič s připojením příčky
- úsek příčky vodorovný
- nebo svislý (může být ve stavu ON-log. 1
- nebo ve stavu OFF-log. 0)
- kontakty (spínací **--]** [**--**,
- rozpínací **--]/**[**--**,
- s detekcí náběžné hrany **--]P**[**--**,
- s detekcí sestupné hrany **--] N** [**--**)

## Výstupy:

- cívky ( cívka **--( )--**,
- negovaná cívka **--(/)--**,
- set **--(S)--**, s pamětí
- reset **--(R)--**,
- memory **--(M)--**,
- set memory **--(SM)--**,
- reset memory **--(RM)--**,
- cívka na náběžnou **--(P)--**
- cívka na sestupnou **--(N)--**
- funkce a funkční bloky

# Jazyk funkčního blokového schématu

## A:Function Block Diagram–FBD

- Jazyk funkčních bloků (FB)



# Příklad programu v jazyku funkčního blokového schématu – FBD

*Zápis \_ pomocí \_ logického \_ výrazu:*

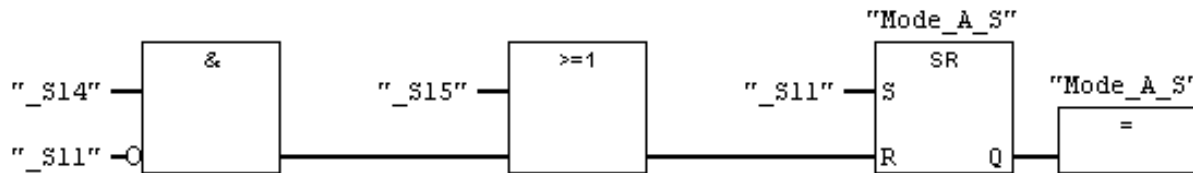
$$S(\text{Mode\_A\_S}) = \_S11$$

$$R(\text{Mode\_A\_S}) = \_S15 + \_S14 * \overline{\_S11}$$

**Network 15:** Start/Stop Auswertung

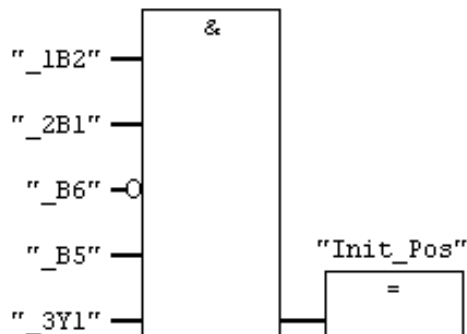
Comment:

$$\text{Mode\_A\_S} = Q(\text{Mode\_A\_S})$$



**Network 16:** Grundstellung

Comment:



$$\text{Init\_Pos} = \_1B2 \_2B1 \overline{\_B6} \_B5 \_3Y1$$

# FBD – standardní funkční bloky

- bistabilní prvky (**RS**-paměť s dominantním vypnutím, **SR** – paměť s dominantním sepnutím, semafor)
- prvky pro detekci náběžné a sestupné hrany, ((**R\_TRIG**), (**F\_TRIG**))
- čítače (inkrementální (**CTU**), dekrementální (**CTD**), kombinované (**CTUD**))
- časovače (impulsní (**TP**), zpožděné sepnutí (**TON**), zpožděné vypnutí (**TOFF**))
- komunikační bloky (definovány v normě IEC 1131-5).

## FBD – speciální bloky

- doplňovány podle potřeby
- každá firma nabízí ve svém programovacím prostředí poněkud odlišný soubor těchto bloků
- např. spínací hodiny týdenní, roční, generátory impulsů, komparátory .....

# Jazyk seznamu instrukcí

## A: Statement List – STL

- Jazyk pokynů (povelů)
- Seznam instrukcí

# Příklad programu v jazyku seznamu instrukcí STL

## Network 15: Start/Stop Auswertung

Comment:

```

U   "_S11"
S   "Mode_A_S"
O   "_S15"
O
U   "_S14"
UN  "_S11"
R   "Mode_A_S"
U   "Mode_A_S"
=   "Mode_A_S"

```

*Zápis \_ pomocí \_ logického \_ výrazu:*

$$S(\text{Mode\_A\_S}) = \_S11$$

$$R(\text{Mode\_A\_S}) = \_S15 + \_S14 * \overline{\_S11}$$

$$\text{Mode\_A\_S} = Q(\text{Mode\_A\_S})$$

## Network 16: Grundstellung

Comment:

```

U   "_1B2"
U   "_2B1"
UN  "_B6"
U   "_B5"
U   "_3Y1"
=   "Init_Pos"

```

$$\text{Init\_Pos} = \_1B2 \_2B1 \overline{\_B6} \_B5 \_3Y1$$

# Jazyk seznamu instrukcí – STL

IEC 61131-3

- program složen ze sekvence instrukcí
- každá instrukce začíná na novém řádku
- v řádku instrukce se může vyskytovat:
  - návěští (nepovinné)
  - operátor (příp. s tzv. modifikátorem)
  - operand
  - komentář (nepovinný)

# Jazyk seznamu instrukcí – STL

## operátory a modifikátory

- povolené operátory :
  - LD, ST; S, R; AND, &, OR, XOR; ADD, SUB, MUL, DIV; GT, GE, EQ, NE, LE, LT; JMP, CAL, RET
- modifikátory:
  - pro označení negace N, případně levá závorka, písmeno C nebo N u podmíněných a nepodmíněných instrukcí skoků, volání a návratů

# Jazyk seznamu instrukcí – STL

## funkce

- volání funkcí vložením jména funkce na místo operátoru
  - do prvního argumentu funkce se zapíše výsledek vyhodnocení funkce.
  - jsou-li vyžadovány další argumenty, píší se do pole operandu



# Jazyk seznamu instrukcí – STL

## funkční bloky

- funkční blok volán podmíněně nebo nepodmíněně operátorem CAL
  - příkaz následován seznamem vstupních parametrů v závorce za názvem funkčního blokunebo
  - jednotlivé parametry uloženy do předdefinovaných proměnných před vyvoláním funkčního bloku

# Jazyk strukturovaného textu

## A: Structured Text – ST

Syntaxe jazyka dána

- povolenými **výrazy**
- povolenými **příkazy**

Vyhodnocením **výrazu** dostaneme hodnotu  
v některém z definovaných datových typů

# Příklad programu v jazyku strukturovaného textu ST

```
K:=0
FOR I:=1 TO 5 DO
    FOR J:=1 TO 4 DO
        IF chyba THEN EXIT
        END_IF
        K:=K+J
    END_FOR
    K:=K+I
END_FOR
```

# Jazyk strukturovaného textu – ST výrazy

Výraz se skládá z **operátorů** a **operandů**.

- Operandy: konstanta, proměnná, funkce, výraz
- Operátory: pro sedmnáct typů operací
  - závorkování, vyhodnocení funkce, umocnění, negace, doplněk, násobení, dělení, modulo, sčítání, odčítání, porovnávání, rovnost a nerovnost, booleovské AND, booleovské XOR, booleovské OR.

Operátory uvedeny od nejvyšší k nejnižší prioritě.

# Jazyk strukturovaného textu – ST příkazy

definováno **deset typů** příkazů:

- příkaz přiřazení
- příkaz pro vyvolání funkce, použití její výstupní hodnoty a pro návrat (RETURN)
- příkazy pro výběr (IF, CASE)
- iterační příkazy (FOR, WHILE, REPEAT, EXIT)
- prázdný příkaz.

## Sekvenční funkční diagram

### A: Sequential function chart – SFC

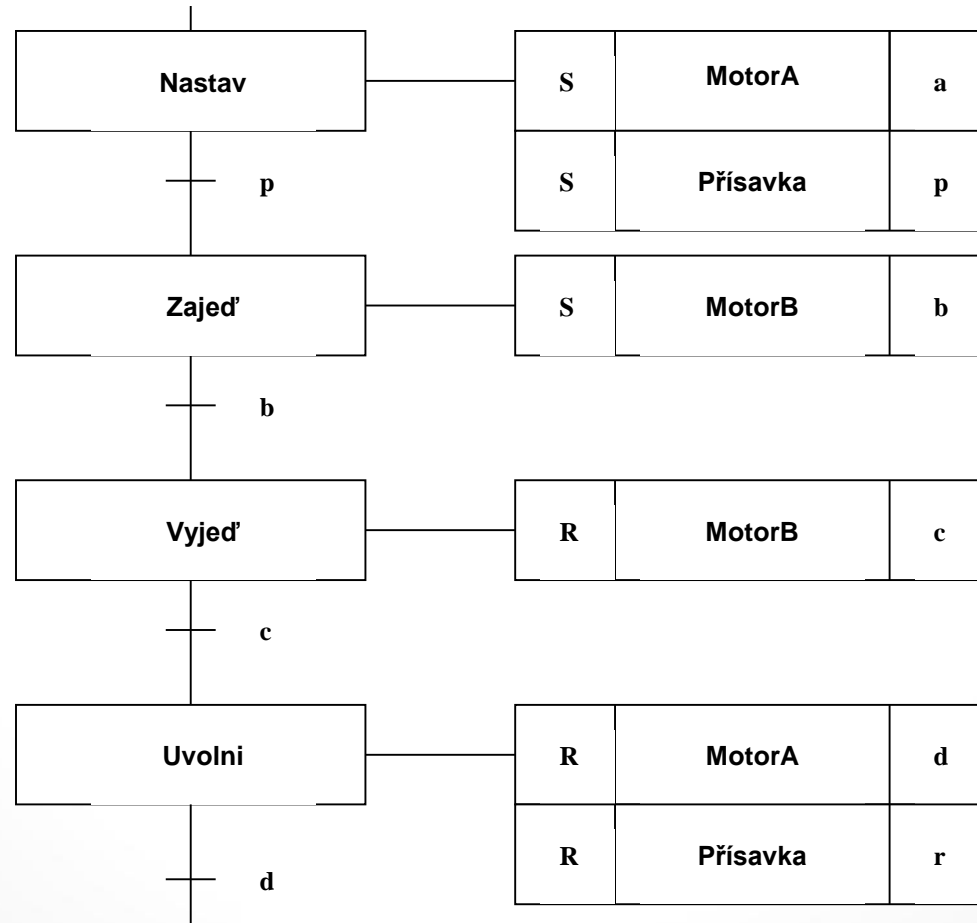
- vychází z obecněji definovaného **funkčního diagramu** (ČSN IEC 848).
- definice: „Funkční diagram je definován souborem symbolů pro kroky, přechody a směrové spojnice, propojující kroky a přechody.....“
- statické znázornění
- dynamický aspekt

# Sekvenční funkční diagram SFC

## prvky jazyka

- krok s identifikátorem, počáteční krok
- blok akcí přiřazených ke kroku
  - čtyři pole: kvalifikátor akce, jméno akce, jméno zpětnovazební proměnné, popis akce,
- přechod s identifikátorem nebo s podmínkou přechodu
- alternativní rozvětvení, spojení alternativních větví
- paralelní (simultánní) rozvětvení, spojení simultánního rozvětvení

# Příklad programu v jazyku SFC





- posun směrem ke **graficky** orientovaným prostředkům
- malé systémy - jazyk blokových schémat logických funkcí FBD (FUP)
- velmi rychle se stane běžným nástrojem programování
- určitá systematičnost při návrhu se vyplácí