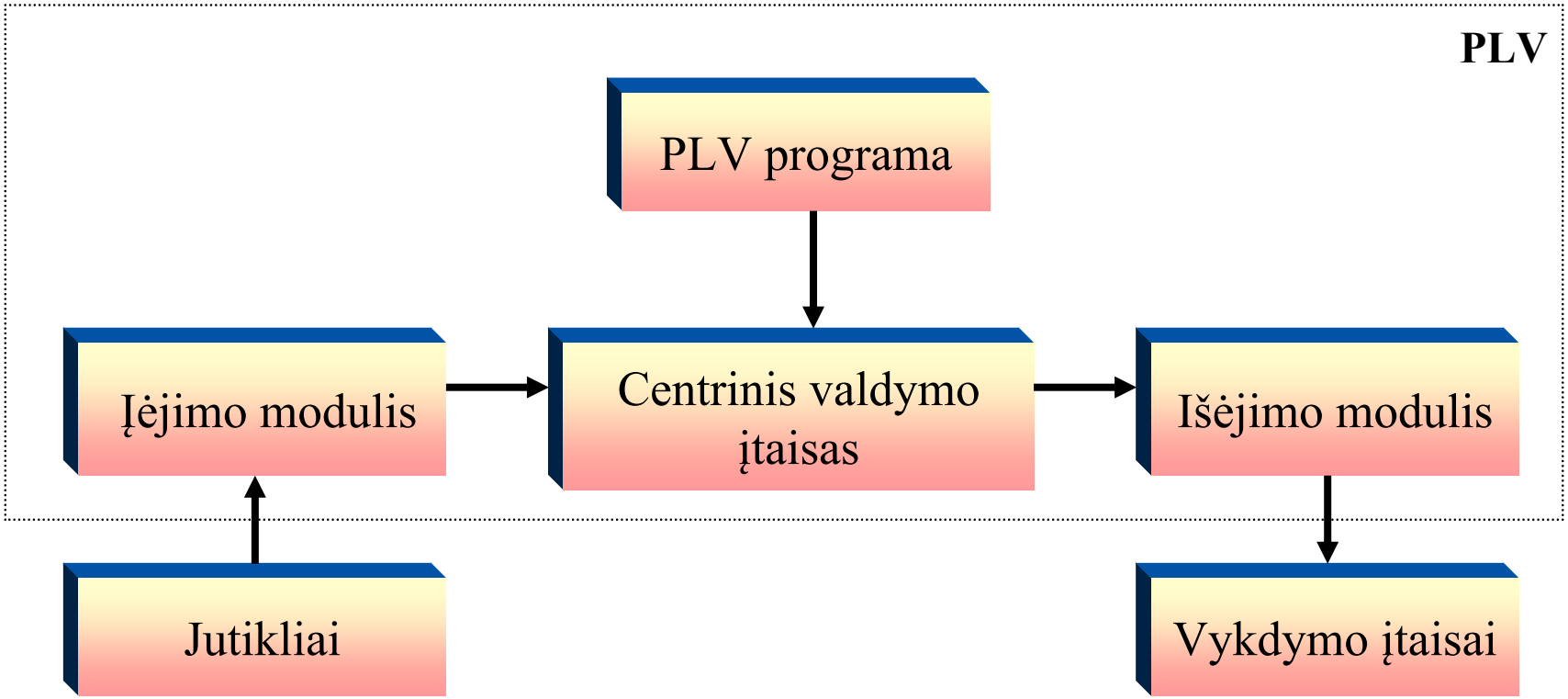
Funkcinė PLV schema (Mechatroninė sistema)



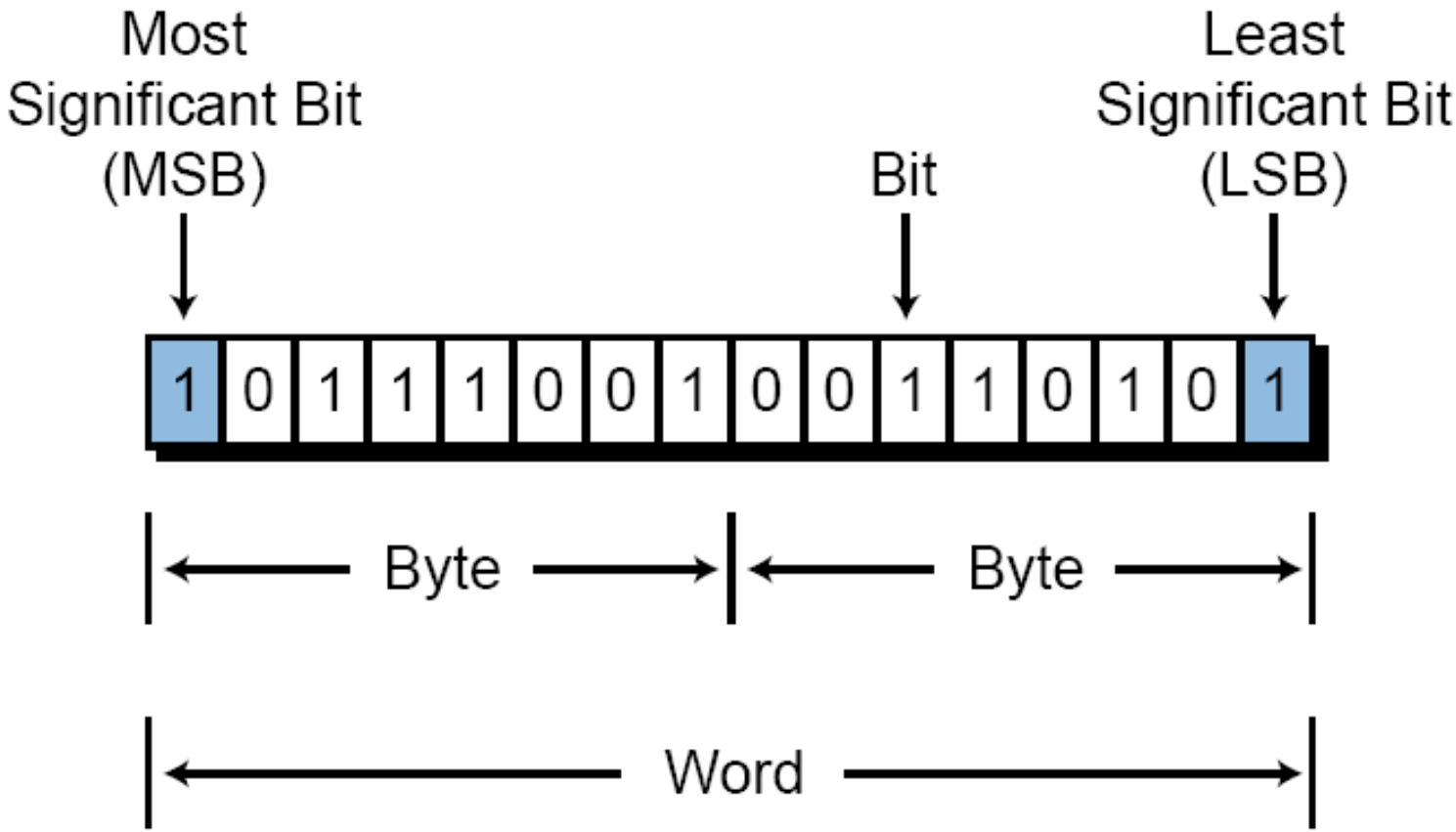
Bendrieji programavimo kalbų elementai

Įėjimai, išėjimai, atmintis (pagal IEC 1131-3 standartą)

Įėjimai	I
Išėjimai	Q
Atmintis	M

Bool	1 bitas
Byte	8 bitai
Word	16 bitų

I,Q,M arba IX,QX,MX	Įėjimo bitas,išėjimo bitas, atminties bitas	1 bitas
IB,QB,MB	Įėjimo baitas, išėjimo baitas, atminties baitas,	8 bitai
IW,QW,MW	Įėjimo žodis, išėjimo žodis, atminties žodis	16 bitų


Most
Significant Bit
(MSB)
Bit
Least
Significant Bit
(LSB)
1 0 1 1 1 0 0 1 0 0 1 1 0 1 0 1
Byte
Byte
Word

Įėjimai, išėjimai, atmintis (pagal IEC 1131-3 standartą)

I1	Pirmasis įėjimas
IX9	Devintasis įėjimas
I15	Penkioliktasis įėjimas
QW3	Trečiasis išėjimo žodis
MB5	Penktasis atminties baitas
MX2	Antroji atmintis

Numeracija gali prasidėti tiek nuo 0, tiek nuo 1.

Hierarchijos lygiai

Pvz., **I 3.8.5**

Tiesiogiai adresuojami kintamieji

%IX12 arba %I12	Dvyliktasis įėjimo bitas
%IW5	Penktasis įėjimo žodis
%QB8	Aštuntasis išėjimo žodis
%MW27	Dvidešimt septintasis atminties žodis

Tiesioginį kintamųjų adresavimą galima taikyti tik programose.

Simbolinis adresavimas

Kad programiškai sudarytos funkcijos ar funkciniai blokai (standartinės paprogramės) išliktų nepriklausomi nuo valdiklio tipo ir juos būtų galima kuo plačiau panaudoti, jie programuojami taikant *simbolinius adresus*.

Simbolinis adresas arba *identifikatorius* sudaromas pasitelkiant didžiąsias arba mažąsias lotyniškas raides, skaičius ar apatinius brūkšnelius. *Identifikatoriu*s visada turi prasidėti raide ar brūkšneliu. Apatinis brūkšnelis gali būti panaudotas norint padaryti identifikatorių lengviau suprantamą.

Pvz., identifikatoriai Stop_var ir Stopvar yra skirtingi, o identifikatoriai Stopvar ir STOPVAR - vienodi.

Duomenų pateikimas

Pavadinimas	*Pavyzdžiai*
Sveikieji skaičiai	12, -8, 123_456*, +75
Skaičiai su plaukiojančiu kableliu	-12.0, -8.0 0.123_4
Dvejetainiai skaičiai	2#1111_1111 (255 dešimtainis) 2#1101_0011 (211 dešimtainis)
Aštuntainiai skaičiai	8#377 (255 dešimtainis) 8#323 (211 dešimtainis)
Šešioliktainiai skaičiai	16#FF ar 16#ff (255 dešimtainis) 16#D3 ar 16#d3 (211 dešimtainis)
Loginis nulis ar vienas	0,1

* Siekiant palengvinti skaitymą, leidžiami apatiniai brūkšneliai tarp skaičių. Pats brūkšnelis nieko nereiškia.

Duomenų pateikimas

Pavadinimas	Pavyzdžiai
Laiko trukmė	T#18ms, t#18ms, t#3.5s t#6h_20m_8s TIME#18ms
Data	D#1994-07-21 DATE#1994-07-24
Dienos metas	TOD#13:18:42.55 TIME_OF_DAY#13:18:42.55
Data ir laikas	DT#1994-07-21-13:18:42.55 DATE_AND_TIME#1994-07-21-13:18:42.55

d	Dienos
h	Valandos
m	Minutės
s	Sekundės
ms	Milisekundės

Duomenų pateikimas

Simbolių sekos, vadinamos eilutėmis (STRING), reikalingos keičiantis informacija, pvz., tarp valdiklių, tarp kitų automatinės sistemos elementų ar programuojant tekstus.

Pavadinimas	Aprašymas
'B'	Vieno simbolio ilgio eilutė, susidedanti iš simbolio B
'Pastaba'	Septynių simbolių ilgio eilutė, sudaranti žodį Pastaba
''	Tuščia eilutė

Duomenų tipai

Raktas	Duomenų tipas	Galimos reikšmės
BOOL	Loginis	0,1
SINT	Trumpas sveikasis	nuo 0 iki 255
INT	Sveikasis	nuo -32 768 iki 32 767
DINT	Dvigubas sveikasis	nuo -2 147 483 648 iki +2 147 483 647
UINT	Sveikasis be ženklo	nuo 0 iki 65 535
REAL	Realus (Skaičius su plaukiojančiuoju kableliu)	nuo +/- 2.9E-39 iki +/-3.4E+38
TIME	Laiko trukmė	Priklauso nuo valdiklio
STRING	Kintamojo ilgio eilutė	Priklauso nuo valdiklio
BYTE	8 bitų seka	Reikšmės neapibrėžtos
WORD	16 bitų seka	Reikšmės neapibrėžtos

Kintamųjų deklaravimas

Nurodomas kintamojo tipas, o patys kintamieji susiejami su resursais. (Programavimo paketai paprastai yra aprūpinti specialiomis kintamųjų deklaravimo priemonėmis).

```
VAR
    Temp            :INT    (*Temperatūra              *)
    Rank            :BOOL (*Rankinio valdymo vėliavėlė    *)
    Pilnas, Atviras  :BOOL (*Vėliavėlės pažymėti „Pilnas“, „Atviras“*)
END_VAR
```

Kintamųjų deklaravimas

Kintamieji priskiriami vienam iš tipų:

VAR_INPUT (įėjimo kintamasis)
VAR_OUTPUT (išėjimo kintamasis)
VAR_IN_OUT (įėjimo ir išėjimo kintamasis)
VAR (lokalus kintamasis)
VAR_GLOBAL (globalus kintamasis)
VAR_EXTERN (išorinis kintamasis).

Lokalūs kintamieji naudojami neišeinantiems į išorę tarpiniams rezultatams įvardyti. Jie prieinami tik viename konkrečiame programos modulyje. Globalieji ir išoriniai kintamieji yra prieinami visuose programos moduliuose.

Kintamųjų inicializavimas

```
VAR_GLOBAL
 Tuzinas :INT :=12;
 END_VAR
```

Tipinės pradinės kintamųjų reikšmės:

Duomenų tipas	*Pradinė reikšmė*
BOOL, SINT, INT, DINT	0
UINT	0
BYTE, WORD	0
REAL	0.0
TIME	T#0s
STRING	'' tuščia eilutė

Programų organizaciniai moduliai

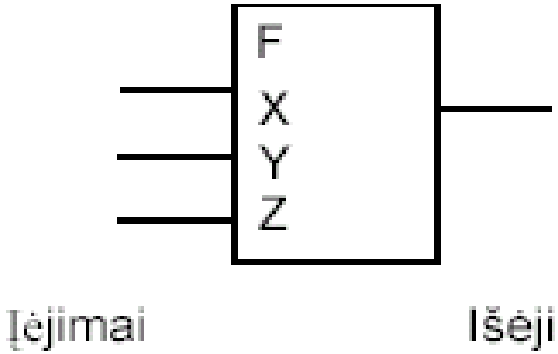
Valdiklių programos skaidomos į individualius organizacinius modulius, sudarančius tokius programos lygius:

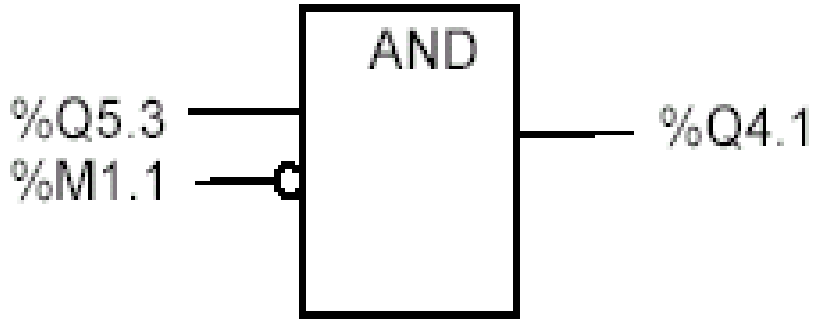
- **programas**;
- **funkcinius blokus**;
- **funkcijas**.

Tipiniams uždaviniams spręsti IEC 1131 standartas numato daugybę standartinių funkcijų ir funkcinių blokų. Be to galima kurti ir savas funkcijas bei funkcinius blokus.

Funkcijos

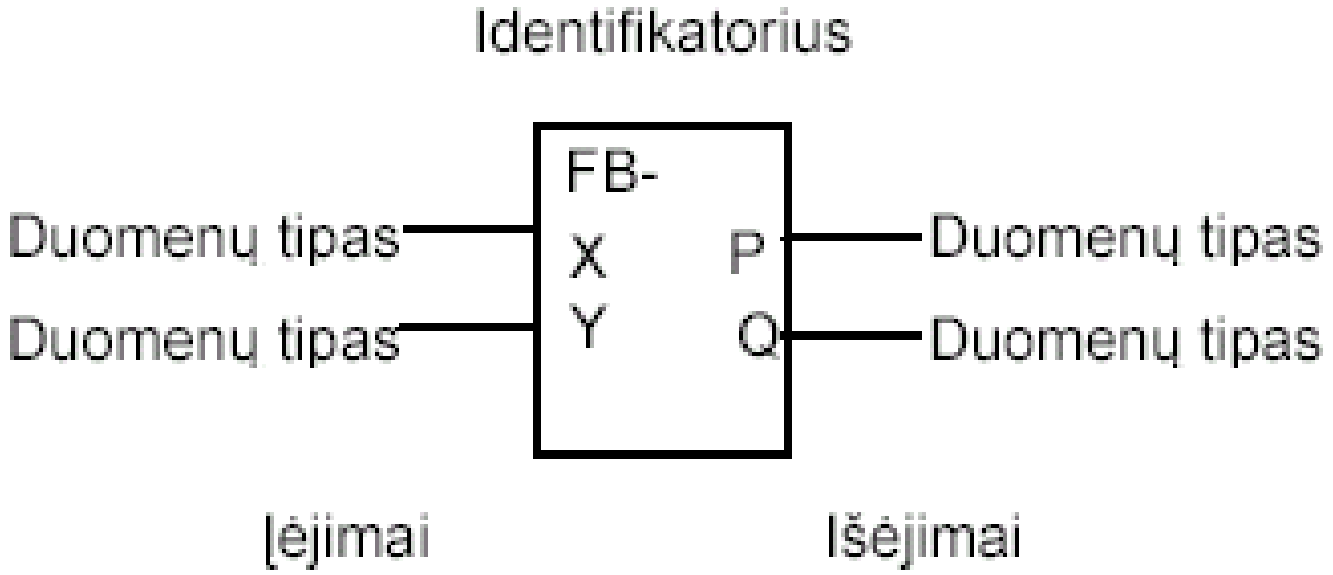
Tai programiniai moduliai, kurie gali formuoti tik vieną rezultatą (duomenų elementą). Funkcija neturi informacijos apie pradinę būseną.



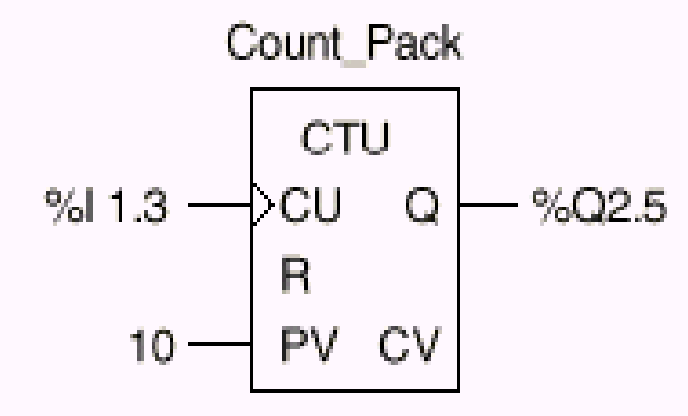


Funkciniai blokai

Tai programiniai moduliai, formuojantys vieną ar kelis išėjimo parametrus.



Funkciniai blokai



Standartiniai:

CTU	Sumuojantysis skaitiklis
CTD	Atimantysis skaitiklis
TP	Impulso formuotuvas
TON	Signalo uždelsimas įjungiant
TOF	Signalo uždelsimas išjungiant
R_TRIG	Fronto nustatymas: augantis frontas
F_TRIG	Fronto nustatymas: krintantis frontas

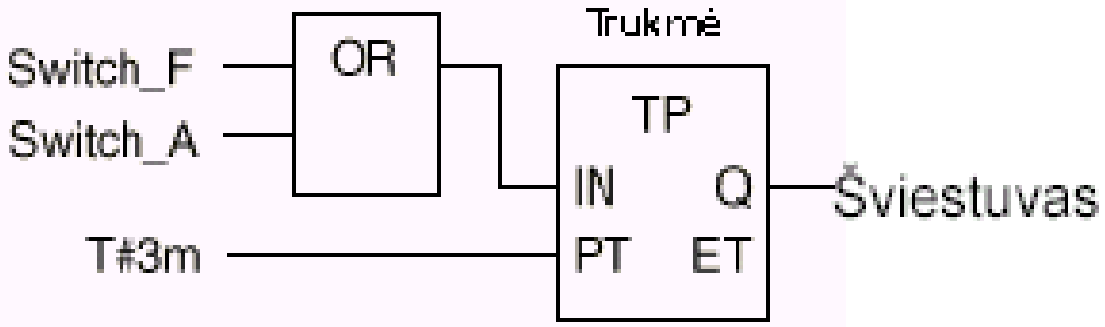
Programos

Susideda iš betkokios programavimo kalbos elementų ir konstrukcijų, reikalingų norimai funkcionavimo tvarkai ar pageidaujamam proceso pobūdžiui gauti.

Programos

```
PROGRAM  laiptinės_apsv

VAR
   Switch_F     AT %IX0.0  : BOOL;   (*Jungiklis prie lauko durų *)
   Switch_A     AT %IX0.1  : BOOL;   (*Jungiklis prie buto durų *)
   Lamp         AT %QX0.0  : BOOL;   (*Šviestuvas *)
   Duration                : TP;     (*Apšvietimo trukmė *)
END_VAR
```

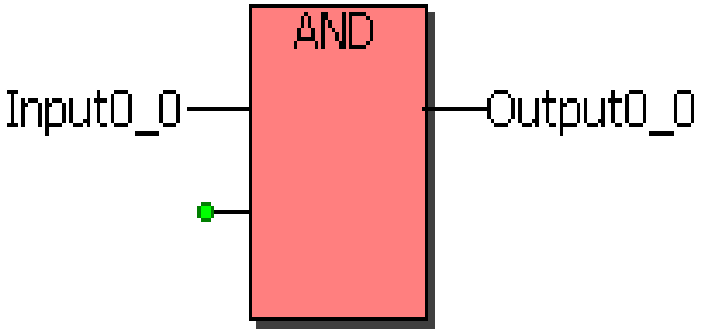


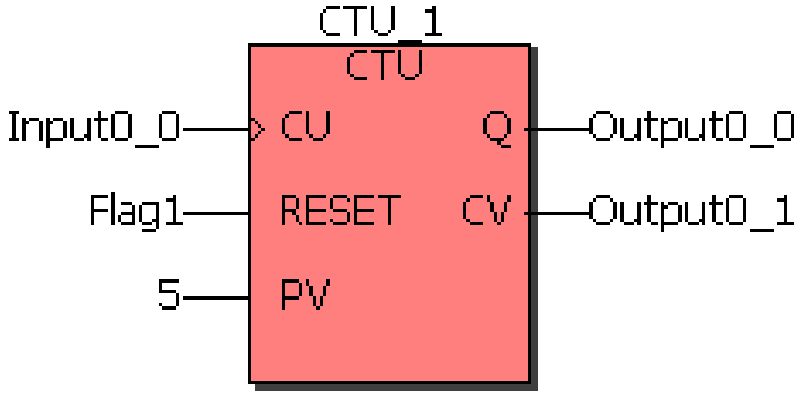

```
END_PROGRAM
```

Programavimo kalbos

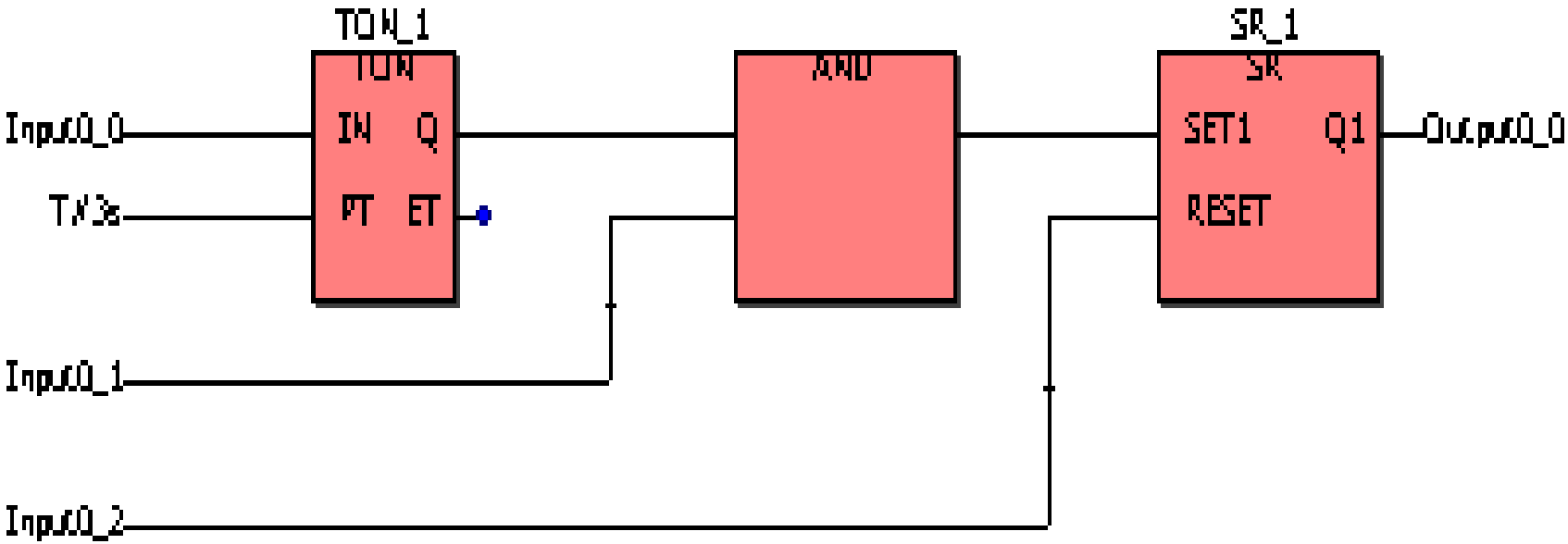
- Funkcinė blokdiagrama (FBD - Function Block Diagram)
- Kontaktų diagrama (LD - Ladder Diagram)
- Komandų sąrašas (IL - Instruction List)
- Struktūrizuotas tekstas (ST - Structured Text)
- Nuosekli funkcinė diagrama (SFC - Sequential Function Chart)

Funkcinė blokdiagrama (FBD - Function Block Diagram)



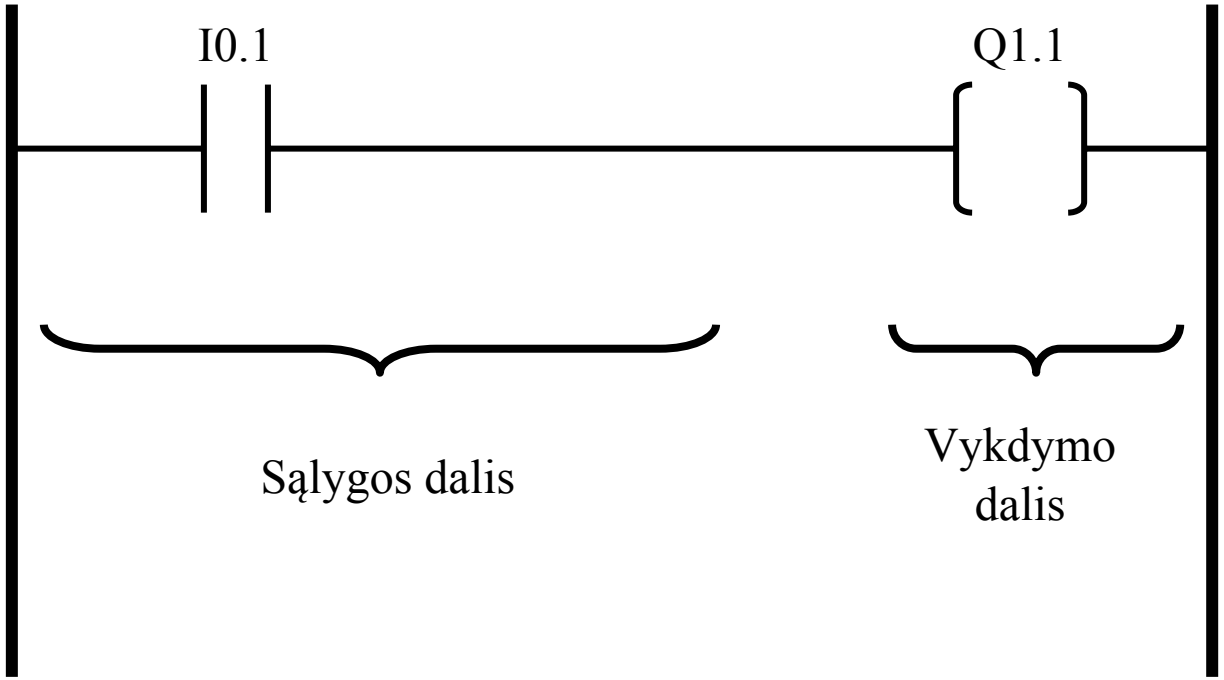


Funkcinė blokdiagrama (FBD - Function Block Diagram)

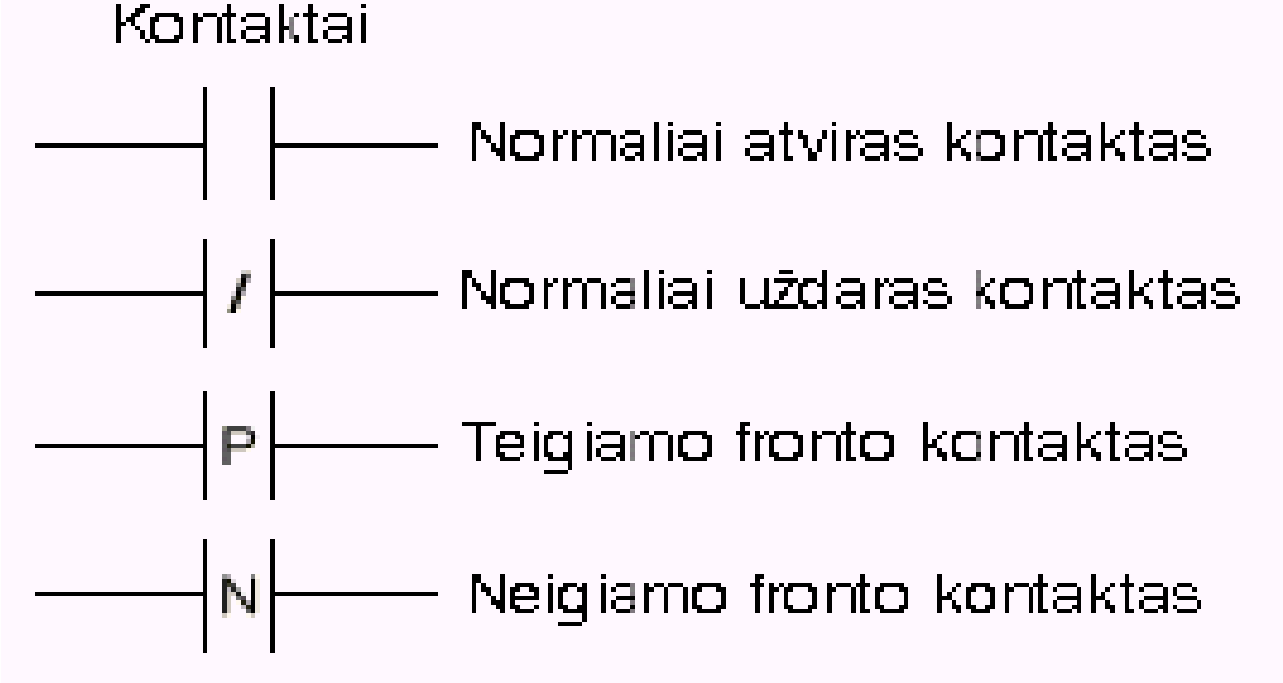


Kontaktų diagrama (LD - Ladder Diagram)

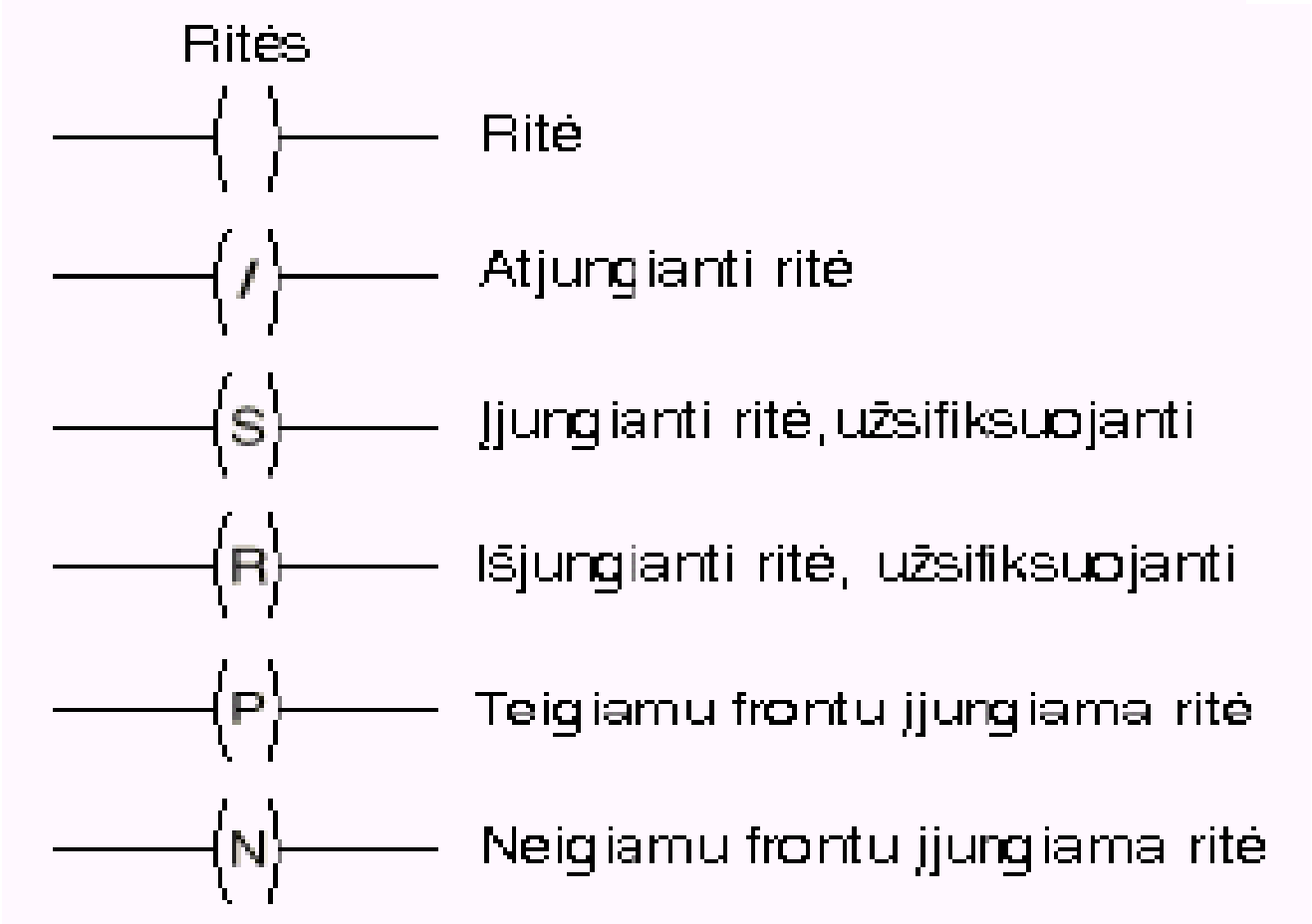
Eilutės struktūra:



Kontaktų diagrama (LD - Ladder Diagram)

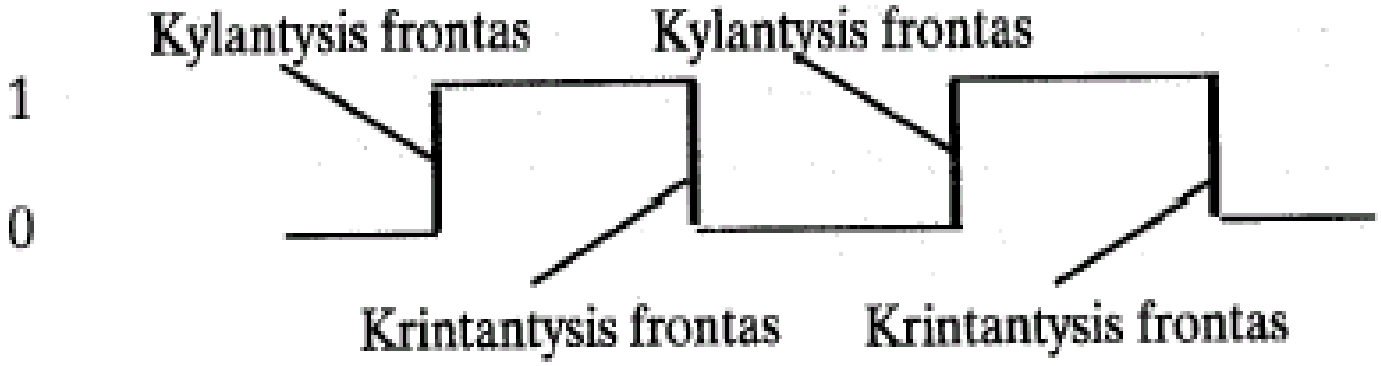


Kontaktų diagrama (LD - Ladder Diagram)



Frontų nustatymas

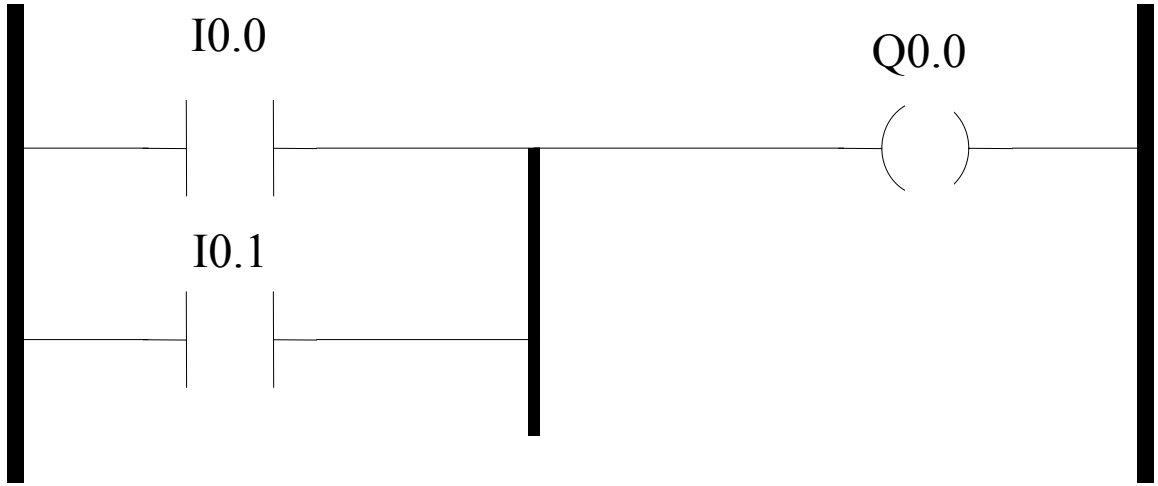
Daugeliu atvejų svarbus ne pats signalas, bet laiko momentas, kada signalas keičia savo reikšmę. Signalo pasikeitimas vadinamas **frontu.**



Kylantieji (teigiami) frontai žymi momentus, kai signalai keičiasi iš 0 į 1.
Krintantieji (neigiami) frontai žymi tuos momentus, kai signalai keičiasi iš 1 į 0.

Kontaktų diagrama (LD - Ladder Diagram)

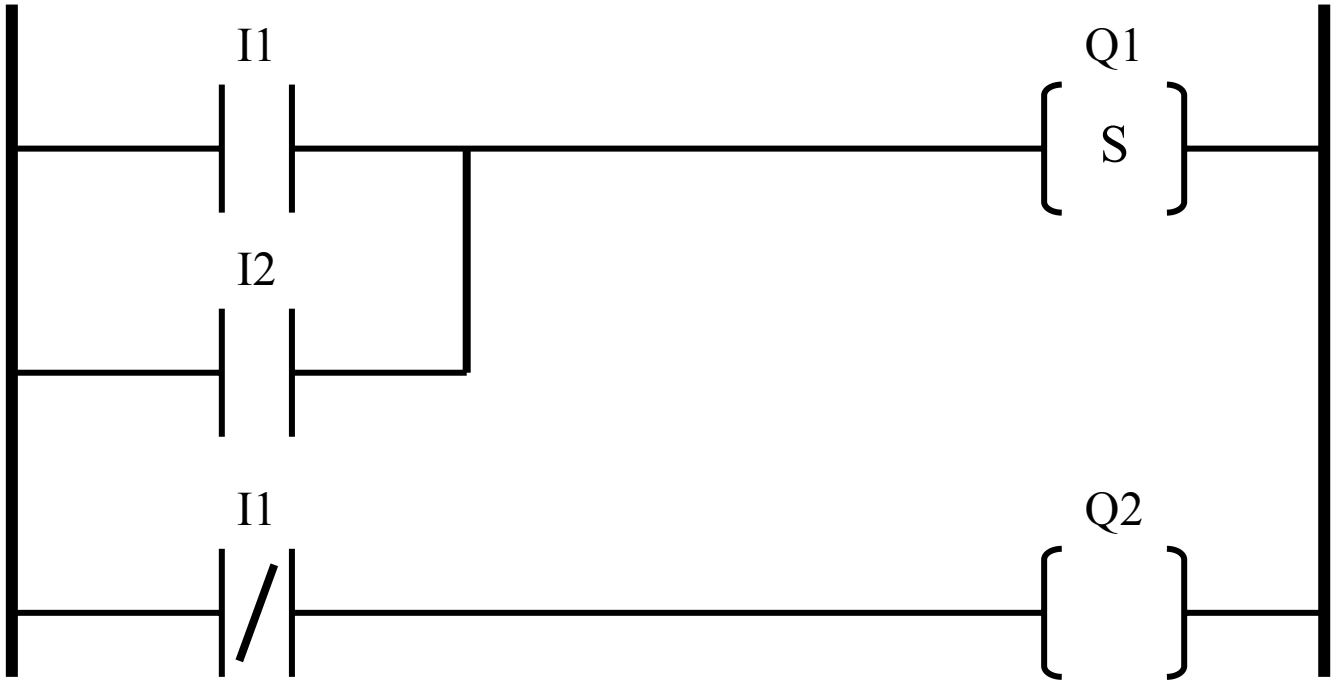
ARBA funkcija



Loginis signalas sklinda iš kairės į dešinę

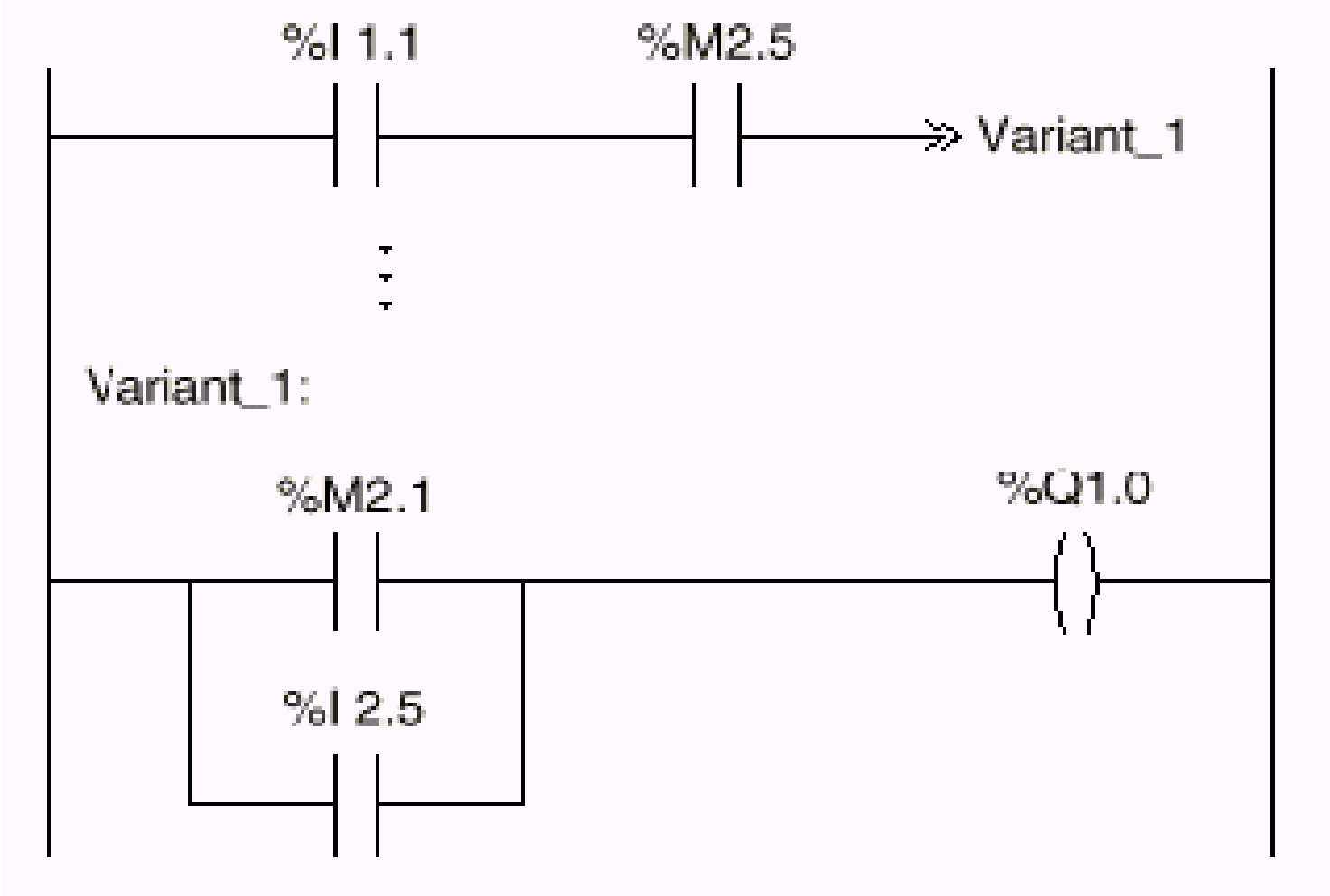
Kontaktų diagrama (LD - Ladder Diagram)

Eilučių vykdymas



Kontaktų diagrama (LD - Ladder Diagram)

Eilučių vykdymas



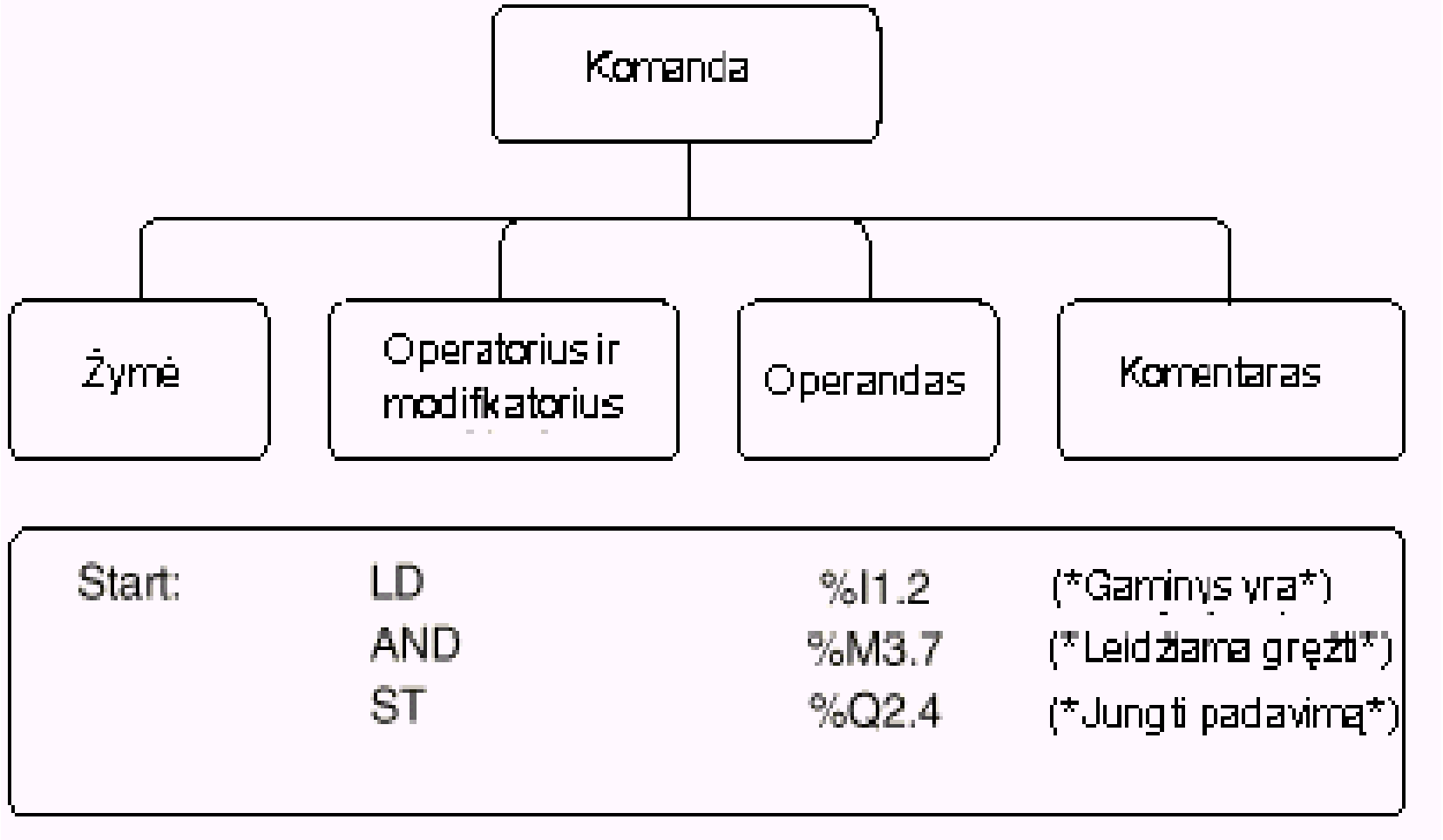
Komandų sąrašas (IL - Instruction List)

Tai tekstinė, asemblerio tipo programavimo kalba. Jos komandos yra labiausiai panašios į PLV vykdomas komandas.

Valdymo programa susideda iš komandų rinkinio, kur kiekviena komanda turi prasidėti iš naujos eilutės.

Komandų sąrašas (IL - Instruction List)

Komandos struktūra



Struktūrizuotas tekstas (ST - Structured Text)

Tai aukšto lygio tekstinė programavimo kalba savo struktūra artima algoritminei *Pascal* programavimo kalbai.

Elementai:

- Žingsnis (Step)
- Sakinys (Sentence)
  - Sąlyginė dalis
  - Vykdymo dalis

Struktūrizuotas tekstas (ST - Structured Text)

```
STEP Start                              (Žingsnio STEP vardas gali būti iki 8 simbolių)
    IF                  I0.0            (Sąlyginė dalis)
    THEN        SET     Q0.0            (Vykdymo dalis)

STEP Stop
    IF                  I0.1
    THEN        RESET   Q0.0
```

Struktūrizuotas tekstas (ST - Structured Text)

```
STEP Start
    IF                  I0.0        Pirmas sakinys
    THEN      SET       Q0.0
    IF                  I0.1        Antras sakinys
    THEN      SET       Q0.1
    IF                  I0.2        Trečias sakinys
   THEN       SET       Q0.2

STEP Stop
   IF                   I0.1
   THEN       RESET     Q0.0
              RESET     Q0.1
              RESET     Q0.2
```

Nuosekli funkcinė diagrama (SFC - Sequential Function Chart)

Valdymo programos struktūrizavimas į individualius etapus ir pereigas (perėjimų iš vieno etapo į kitą sąlygas), tarpusavyje sujungtus orientuotais ryšiais.

Tai hibridinė programavimo kalba - joje derinami grafinės ir tekstinės kalbos elementai.

Nuosekli funkcinė diagrama (SFC - Sequential Function Chart)