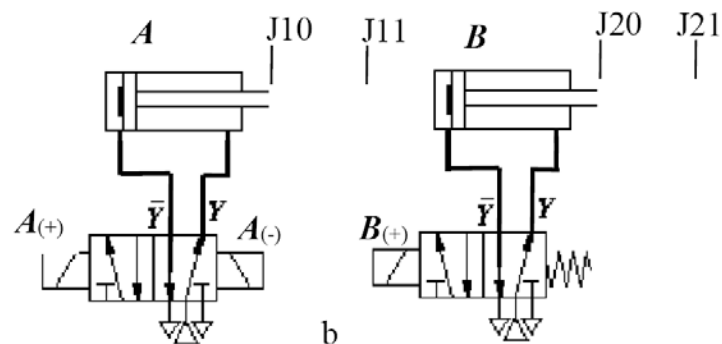
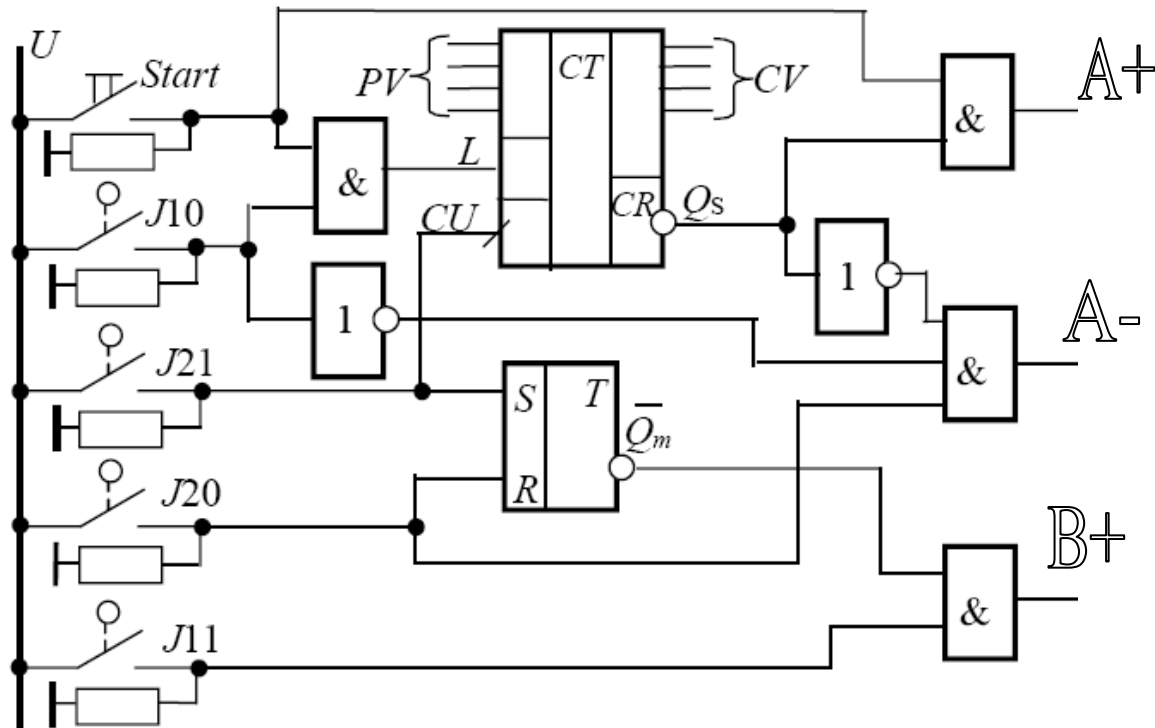
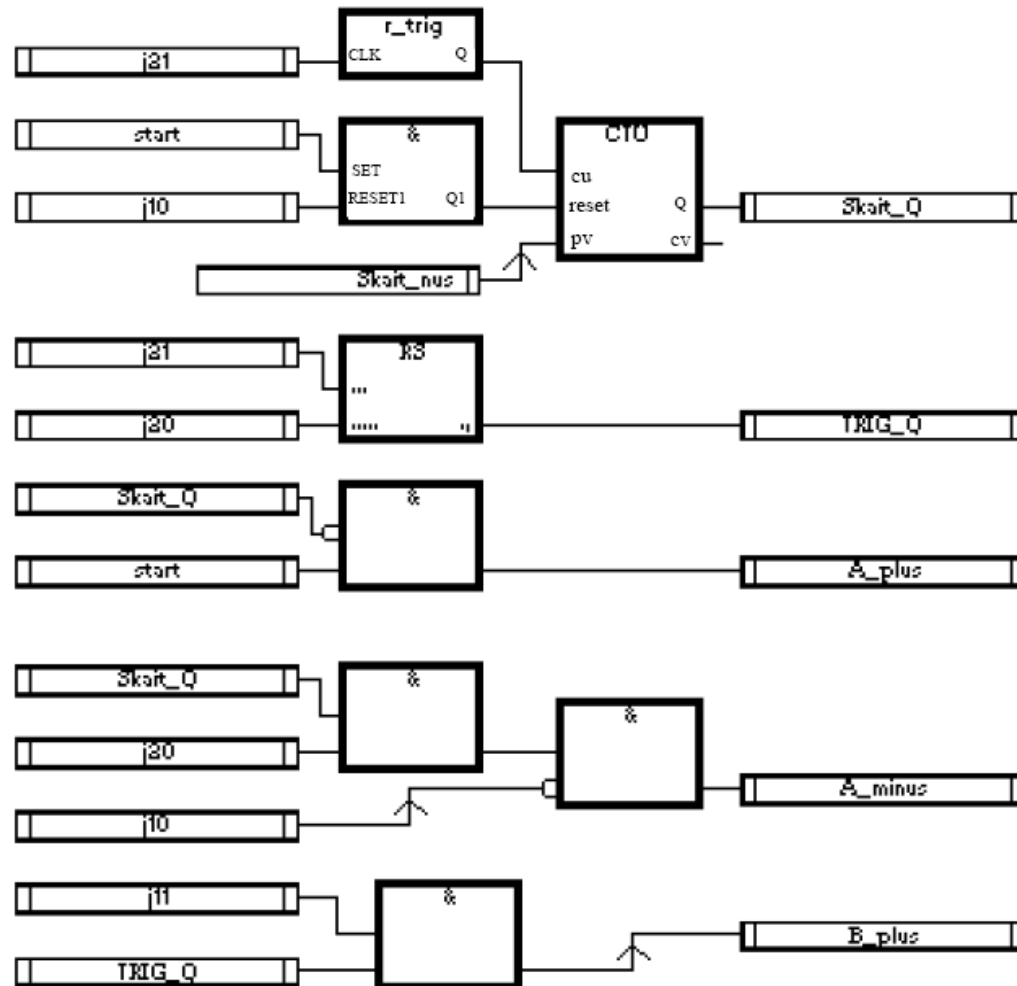


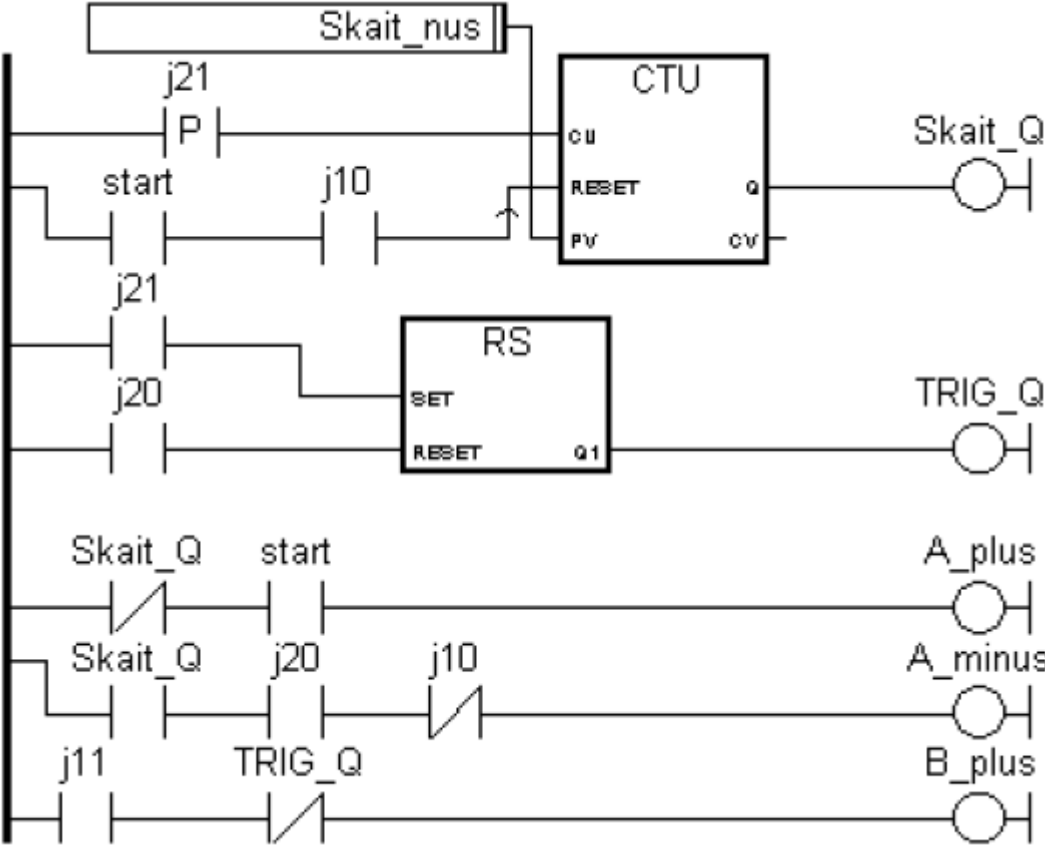
Sistemos su skaitikliu schema



FBD programos pavyzdys



LD/FBD programos pavyzdys



Komandų sąrašo kalba parašyta programa

LD j21	(* Kelio jungiklio <i>j21</i> formuojamas signalas	*)
ST R_trig1.clk	(* priskiriamas fronto trigerio <i>R_trig1</i> įėjimo argumentui <i>clk</i>	*)
CAL R_trig1	(* Kreipiamasi į fronto trigerio <i>R_trig1</i> funkcinį bloką <i>R_trig1</i>	*)
LD Start	(* Paleidimo mygtuko	*)
AND j10	(*ir kelio jungiklio <i>j10</i> signalų AND operacijos rezultato reikšmė	*)
ST Skait_1.reset	(*priskiriama skaitiklio <i>Skait_1</i> įėjimo argumentui <i>reset</i>	*)
LD R_trig1.Q	(* <i>R_trig1</i> trigerio išėjimo <i>Q</i> reikšmė	*)
ST Skait_1.CU	(* priskiriamas skaitiklio <i>Skait_1</i> įėjimo argumentui <i>CU</i>	*)
LD Skait_nus	(* Skaitiklio nustatymo konstanta <i>Skait_nus</i>	*)
ST Skait_1.PV	(*įrašoma į nustatymo registrą <i>PV</i>	*)
CAL Skait_1	(* Kreipiamasi į skaitiklio <i>Skait_1</i> funkcinį bloką	*)
LD j21	(*Kelio jungiklio <i>j21</i> formuojamas signalas	*)
ST Trig_1.reset1	(*priskiriamas atminties vėliavėlei (trigeriui <i>Trig_1</i>) išjungti	*)
LD j20	(*Kelio jungiklio <i>j20</i> formuojamas signalas	*)
ST Trig_1.set	(*priskiriamas atminties vėliavėlei (trigeriui <i>Trig_1</i>) įjungti	*)
CAL Trig_1	(*Kreipiamasi į trigerio <i>Trig_1</i> funkcinį bloką	*)
LD Start	(* Formuojama skirstytuvo <i>A_plus</i>	*)
ANDN Skait_1.Q	(*įjungimo sąlyga – $A_plus = Start \cdot \overline{Q_s}$	*)
ST A_plus	(* loginės sąlygos rezultatas priskiriamas <i>A_plus</i> išėjimui	*)
LD Skait_1.Q	(* Formuojama skirstytuvo <i>A_minus</i>	*)
AND j20	(*įjungimo sąlyga –	*)
ANDN j10	(* $A_minus = Q_s \cdot j20 \cdot \overline{j10}$	*)
ST A_minus	(* loginės sąlygos rezultatas priskiriamas <i>A_minus</i> išėjimui	*)
LD j11	(* Formuojama skirstytuvo <i>B_plus</i>	*)
ANDN Trig_1.Q1	(*įjungimo sąlyga – $B_plus = j11 \cdot \overline{Q_m}$	*)
ST B_plus	(* loginės sąlygos rezultatas priskiriamas <i>B_plus</i> išėjimui	*)

<i>Operatorius</i>	<i>Modifikatorius</i>	<i>Operandas</i>	<i>Aprašymas/Reikšmė</i>
LD	N		Įrašo nurodytą operandą į kaupiklį ir palygina rezultata su šiuo operandu
ST	N		Priskiria rezultatą nurodytajam operandui
S		Loginis	Loginiam operandui suteikia <i>TRUE</i> reikšmę, su sąlyga, jei rezultato loginė reikšmė taip pat <i>TRUE</i>
R		Loginis	Loginiam operandui suteikia <i>FALSE</i> reikšmę, su sąlyga, jei rezultato loginė reikšmė yra <i>TRUE</i>
AND	N (Loginis	Loginis IR
&	N (Loginis	Loginis IR
OR	N (Loginis	Loginis ARBA
XOR	N (Loginis	Loginis išskirtinis ARBA
ADD	(Sudėtis
SUB	(Atimtis
MUL	(Daugyba
DIV	(Dalyba
GT	(Palyginimas:>
GE	(Palyginimas:>=
EQ	(Palyginimas:=
NE	(Palyginimas:<>
LE	(Palyginimas:<=
LT	(Palyginimas:<
JMP	C N	Žymė	Šuolis prie žymės
CAL	C N	Vardas	Funkcinio bloko iškvietimas
RET	C N		Funkcijos ar funkcinio bloko pabaiga
)			Operacijos pabaiga

Modifikatorius *N* informuoja jog po loginės operacijos reikia atlikti inversiją. Modifikatorius “(“ rodo jog reikia atidėti prieš šį modifikatorių esančią operaciją, o vykdyti tas operacijas, kurios yra už šio modifikatoriaus. Modifikatorius “)” informuoja, jog reikia grįžti prie atidėtos jos operacijos. Modifikatorius *C* – tai sąlyginės operacijos modifikatorius. Jis gali eiti drauge su modifikatoriumi *N*, rodančiu, jog sąlyginę operaciją galima vykdyti tik jei sąlygos rezultatas yra *FALSE*

Struktūrizuoto teksto ST kalba parašyta programa

```
R_trig1(j21);  
Skait_1(R_trig1.Q, Start AND j10,5);  
Trig_1(j21,j20);  
A_plus:= Start AND NOT Skait_1.Q;  
A_minus:=Skait_1.Q AND j20 AND NOT j10;  
B_plus:= j11 AND Trig_1.Q1.
```

<i>Teiginys</i>	<i>Pavyzdys</i>		
Priskyrimas :=	A:=B; CV:=CV+1; Y := COS(X);		
Funkcinių blokų iškvietimas	RS_trig(S:=Klaida,R1:=mygtukas); Sirena :=RS_trig.Q1		
Išėjimas iš funkcijos arba iš funkcinio bloko RETURN	RETURN		
Išrinkimo teiginiai	<u>IF</u> D:=B*B - 4*A*C; IF D<0.0 THEN Srend_Sk :=0; ELSIF D =0.0 THEN Srend_Sk:=1; X1:=-B/(2.0*A); ELSE Number_Sin :=2; X1:=(-B+SQRT(D))/(2.0*A); X2:=(-B-SQRT(D))/(2.0*A); END_IF;	<u>CASE</u> CASE Itamos Ind 101...200: Indik:=per_auksta; 20...100: Indik:=auksta; 2...29: Indik:=normali; ELSE Indik:=per_zema; END_CASE;	
	Ciklai	<u>FOR</u> Total:=0; FOR I:=1 TO 5 DO Total:=Total + I; END_FOR;	<u>REPEAT</u> Total:=0; I:=0; REPEAT I:=I+1; Total:=Total + I; UNTIL I =5 END_REPEAT
Ciklų pabaiga EXIT		<u>EXIT;</u>	
Tuščia komanda	::		

SFC programos struktūra

```

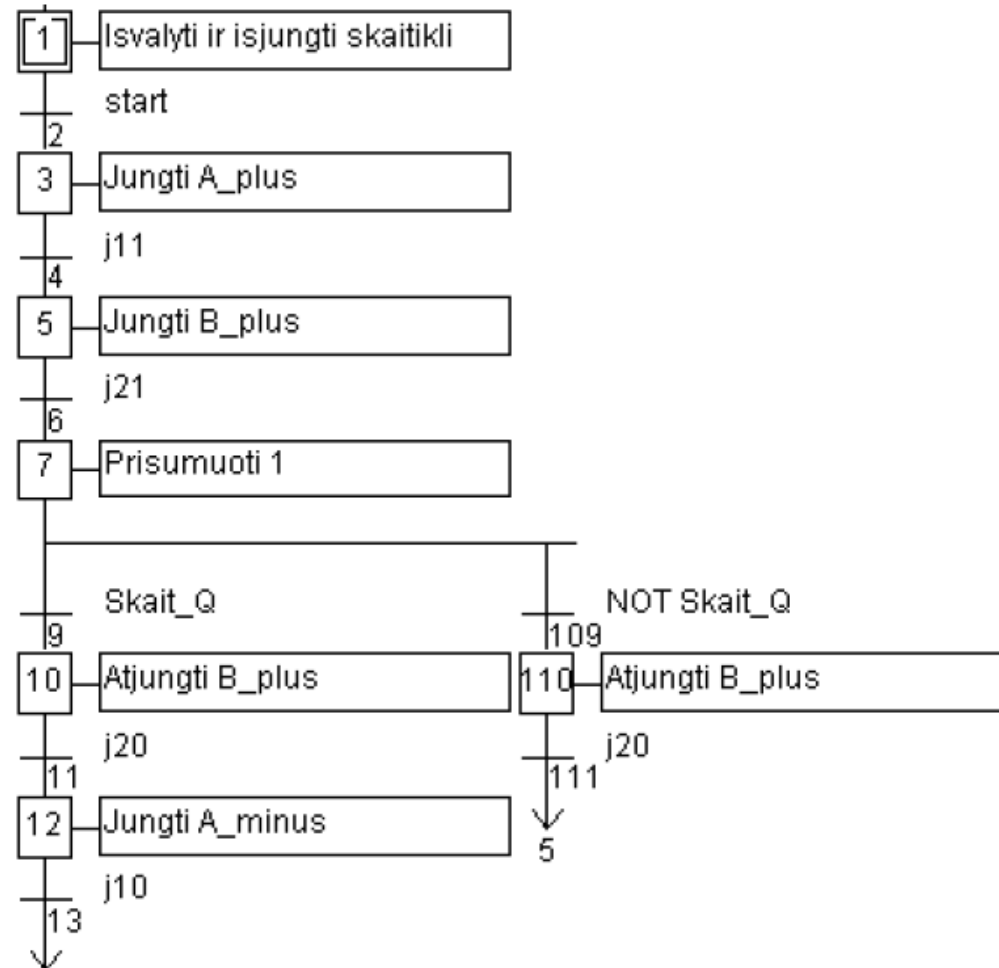
ACTION(N);
  Skait_res:= 0;
  Skait_Q:=FALSE;
  A_plus:= FALSE;
  A_minus:=TRUE;
  B_plus:= FALSE;
END_ACTION;

```

```

ACTION(N);
  Skait_rez:=Skait_rez+1;
  IF Skait_rez=5 THEN
    Skait_Q:=TRUE;
  END_IF;
END_ACTION;

```

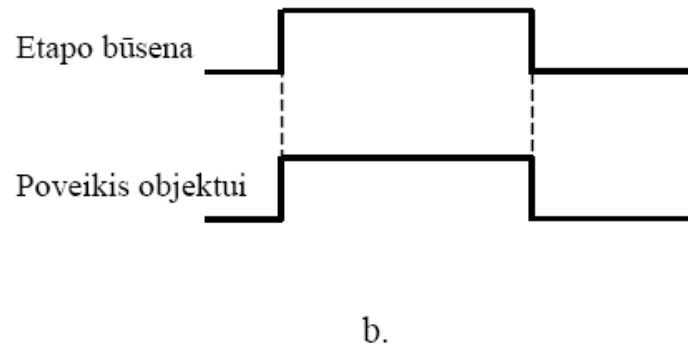
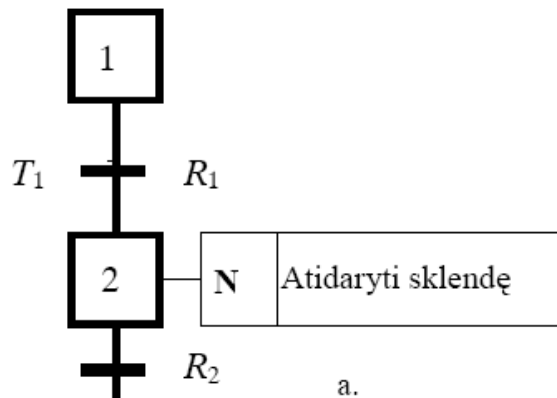


SFC diagramomis susietų poveikių pažymiai

- N* – neįsimintieji poveikiai;
- S* – įsimintieji poveikiai;
- D* – uždelstieji poveikiai;
- L* – nustatytos trukmės poveikiai;
- P* – impulsiniai poveikiai
- C* – sąlyginiai poveikiai.

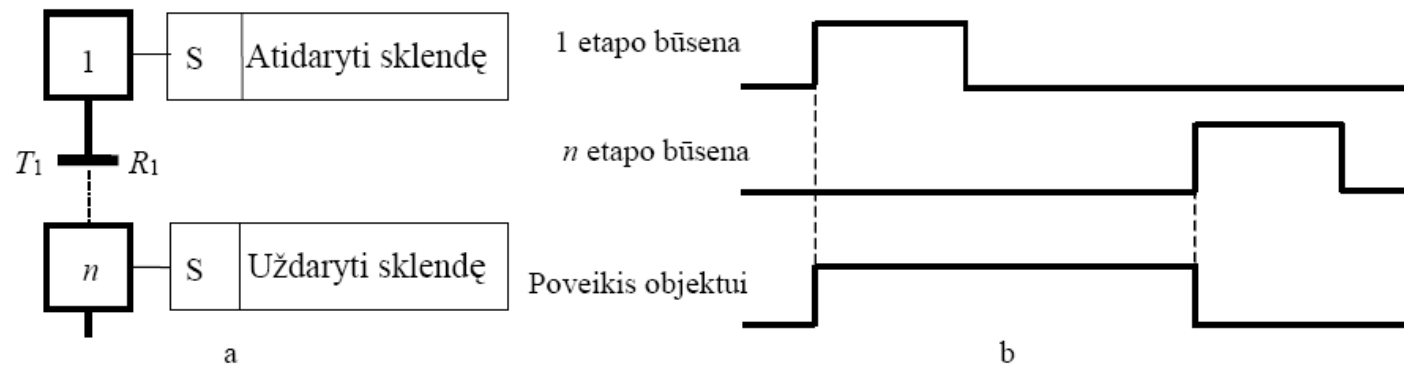
Neįsimintieji poveikiai (*N* – *nostored*)

Neįsimintieji poveikiai – tai tokie poveikiai, kurie įjungiami suaktyvinus atitinkamą etapą ir veikia tol, kol juos suaktyvines etapas vėl aktyvus.



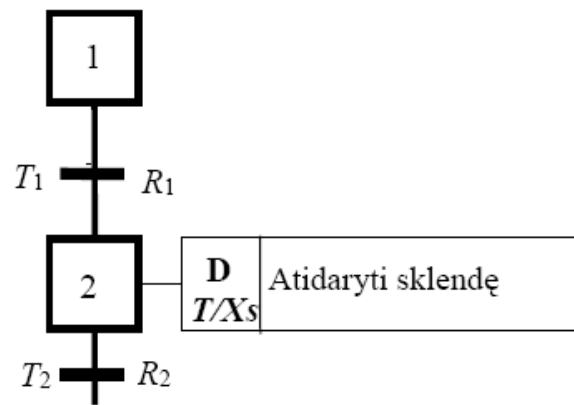
Įsimintieji poveikiai (S-stored)

Įsimintieji poveikiai – tai poveikiai, liekantys aktyvūs ir etapui pasibaigus. Tokie poveikiai žymi operacijos pradžią, kuri sutampa su konkretaus etapo įjungimu. Operacijos pabaiga turi būti modeliuojama kitu etapu, pavyzdžiui vienas etapas susiejamas su poveikiu „proceso A pradžia“, o kitas – su poveikiu „proceso A pabaiga“ Tai reiškia, kad procesas vyksta tarp dviejų įvykių, lemiančių atitinkamų etapų pradžią. Įsimintam poveikiui realizuoti geriausiai tinka bistabilūs valdymo įtaisai (trigeriai, bistabilūs skirstytuvai ir pan.)

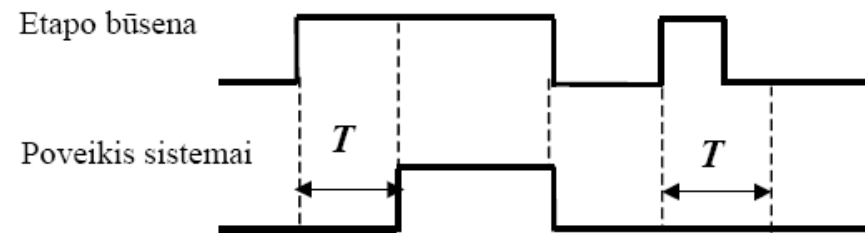


Uždelstieji poveikiai (D – delayed):

Uždelstieji poveikiai – tai poveikiai, pradedantys veikti praėjus nustatytam laikui po to, kai buvo suaktyvintas atitinkamas etapas. Etapai praradus aktyvumą, procesas išjungiamas.



a

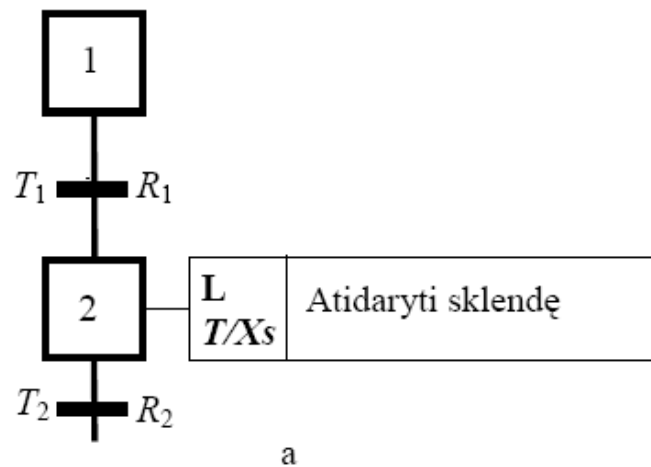


b

Nustatytos trukmės poveikiai (L – limited)

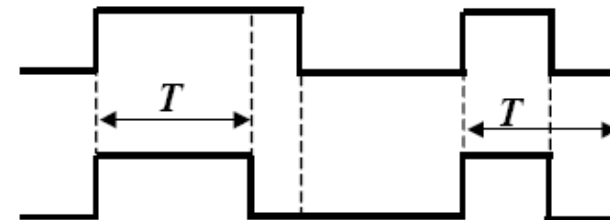
Nustatytos trukmės poveikiai – tai nustatytą laiką veikiantys poveikiai. Tokie poveikiai prasideda suaktyvėjus atitinkamam etapui ir baigiasi praėjus nustatytam laiko intervalui arba, jei etapas išjungiamas anksčiau, išjungiami drauge su etapu

Įveikus pereigą, R_1 sklendė bus atidaryta X sekundžių arba tol, kol bus įveikta pereiga R_2 ; taigi, jei nustatytasis laikas yra ilgesnis už atitinkamo etapo aktyvumo periodą, poveikis formuojamas taip pat, kaip neįsimintas poveikis, t.y. tol, kol atitinkamas etapas yra aktyvus.



Etapo būseną

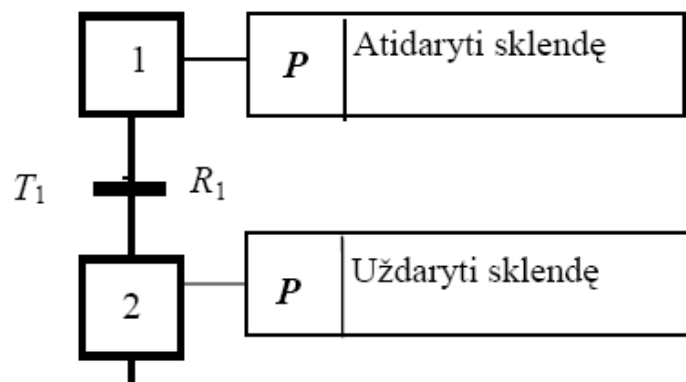
Poveikis sistemai



b

Impulsiniai poveikiai (P – pulse)

Impulsiniai poveikiai – tai labai trumpi, vienetiniai poveikio impulsai, formuojami atitinkamo etapo suaktyvinimo momentu (įveikiant pereigas).



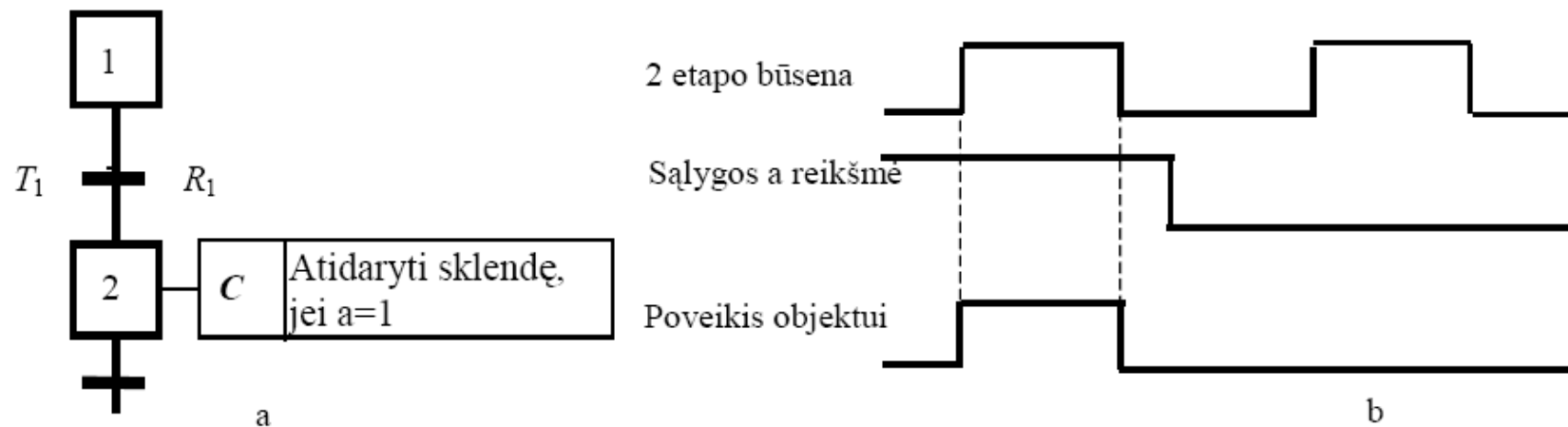
a



b

Sąlyginiai poveikiai (C – Conditional)

Sąlyginiai poveikiai – tai poveikiai, kurie formuojami esant aktyviam etapui ir patenkinus papildomas sąlygas.



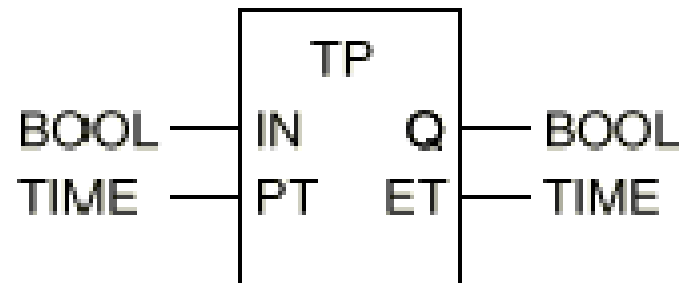
Laiko relės (Taimeriai)

IEC 1131-3 standartas numato trijų tipų taimerių funkcinius blokus:

- TP - impulsų trukmės formuotuvai;
- TON - įjungimo vėlintuvai;
- TOF - išjungimo vėlintuvai.

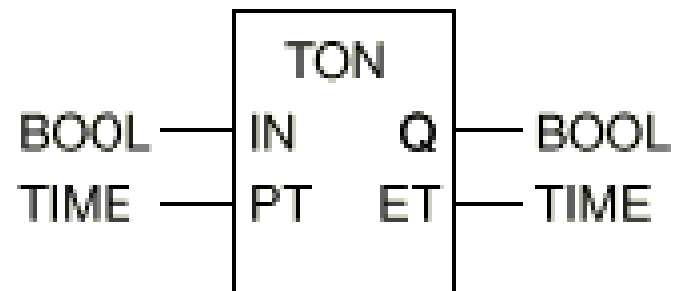
Impulso trukmės formuotuvas (TP taimeris)

Funkciniai TP (*timer pulse*) blokai - impulsų trukmės formuotuvai - įjungiami ilgesniu ar trumpesniu įėjime IN veikiančiu “1” signalu. Išėjime “1” signalas veikia PT (*preset time*) įėjime nustatytą laiko trukmę. Taigi išėjime suformuojamas tam tikros trukmės signalas. TP negali būti pakartotinai įjungtas, kol jis yra aktyvus. Konkreti laiko reikšmė gali būti stebima išėjime ET (*estimated time*).



Ijungimo uždelsiklis (TON taimeris)

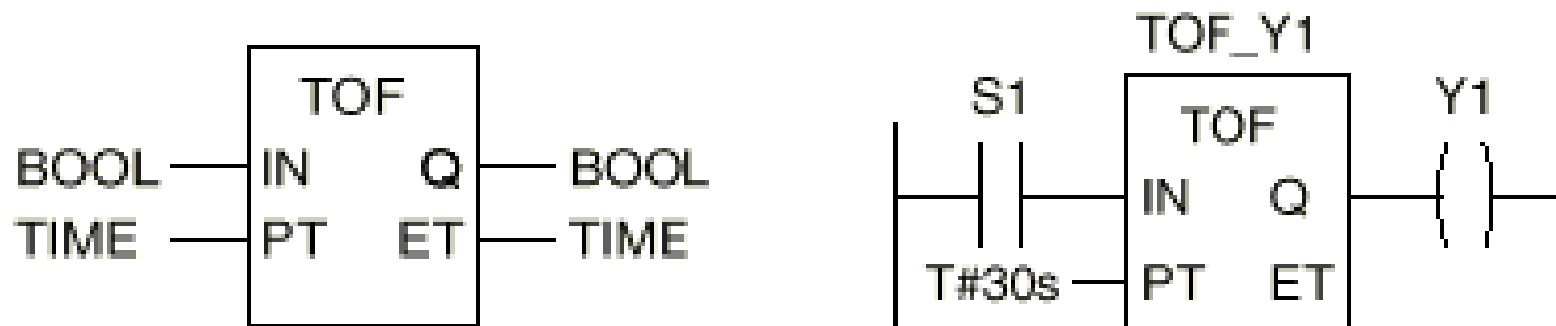
Funkcinis TON blokas (laiko uždelsimo taimeris) naudojamas norint uždelsti signalo įjungimą. Suformavus starto signalą IN įėjime, išėjimas Q įjungiamas tik kai praeina nustatytas tam tikras laiko tarpas PT ir išlieka, kol nepanaikinamas įėjimo signalas. Jei įėjimo signalo trukmė yra mažesnė už delsos laiką, išėjimo Q reikšmė išlieka lygi nuliui (išėjimas neįjungiamas).



Išjungimo uždelsiklis (TOF taimeris)

TOF - tai funkcinis blokas, uždelsiantis signalo išjungimą.

Suformavus signalą IN įėjime, išėjimas Q įjungiamas iš karto. Kai įėjimo signalas panaikinamas, išėjimas Q išjungiamas tik praėjus nustatytam laiko tarpui PT.



Skaitikliai

Skaitikliai reikalingi vienetiniams gaminiams ir įvykiams skaičiuoti.
IEC 1131-3 standarte numatyti trys skirtingi skaitiklių moduliai:

- CTU – sudėties skaitiklis;
- CTD – atimties skaitiklis;
- CTUD – sudėties ir atimties skaitiklis.

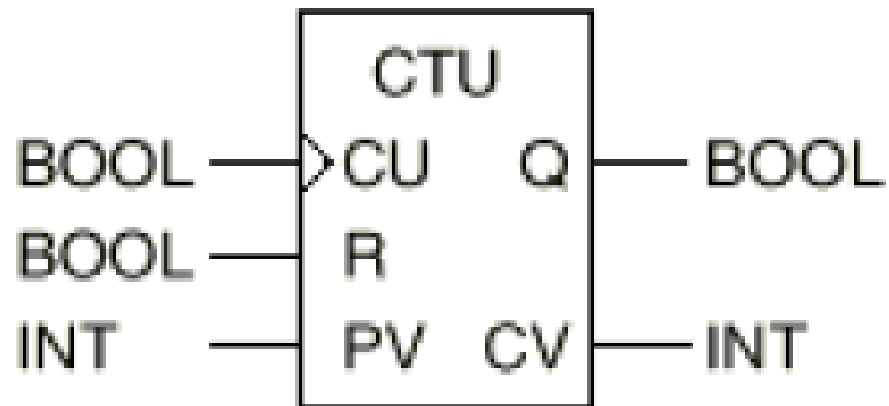
Skaitikliai

Šie standartiniai funkciniai moduliai naudojami laiko požiūriu normaliai vykstantiems įvykiams skaičiuoti. Tačiau daugelyje valdymo uždavinių reikia vadinamųjų didelio greičio skaitiklių. “Didelis greitis” paprastai reiškia, kad skaitiklio skaičiavimo greitis viršija 50 Hz, t. y. per sekundę suskaičiuojama daugiau kaip 50 įvykių. Šio tipo uždavinių negali spręsti standartiniai PLV skaitiklių funkciniai moduliai.

Skaičiavimo dažnis skaitiklių funkcinuose blokuose ribojamas dėl išėjimo signalo uždelsimo. Kiekvienas įėjimo signalas, taip pat ir skaitiklio signalas, prieš jį apdorojant valdiklyje yra šiek tiek uždelsiamas. Taip išvengiama signalų sutapimo. Kitas ribojimas – tai PLV ciklo trukmė. Todėl prireikus skaičiuoti dideliu greičiu reikalingi papildomi skaitiklių moduliai. Didelio greičio skaitikliai naudojami, pavyzdžiui, pozicionavimo sistemose.

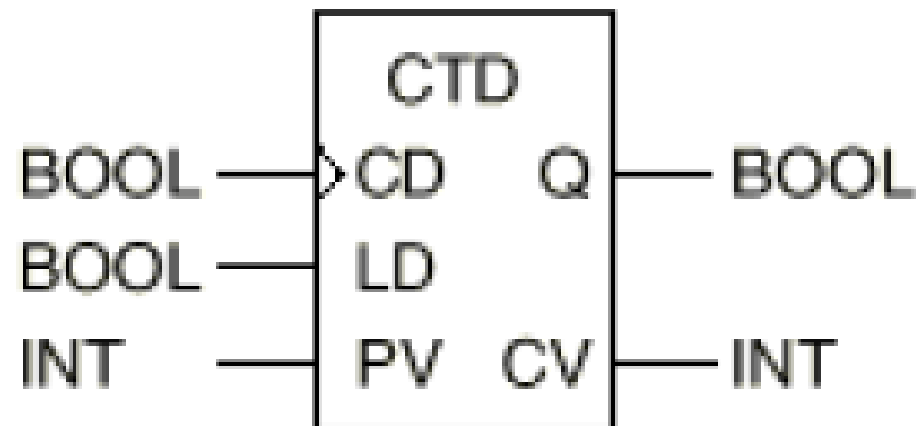
Sudėties skaitiklis

CTU (*count up*) skaitiklis. Skaitiklis nustatomas pradinei 0 reikšmei loginiu signalu veikiančiu išjungimo įėjime R. Skaitiklio būseną (jo turinys) yra prieinama CV išėjime. Su kiekvienu teigiamu skaitiklio loginio įėjimo CU signalo frontu, skaitiklio turinys padidėja 1. Tuo pat metu funkciname bloke skaitiklio turinys yra lyginamas su nustatyta PV reikšme. Kai tik skaitiklio turinys CV susilygina (ar viršija) su nustatyta reikšme, išėjimo signalas Q įgyja loginio "1" reikšmę. Iki tol skaitiklio išėjimo Q reikšmė buvo loginis "0".



Atimties skaitiklis

CTD (*count down*) skaitiklis. Atimties skaitiklis pageidaujamai PV reikšmei nustatomas padavus į LD įėjimą loginio “1” signalą. Su kiekvienu CD įėjime veikiančio loginio signalo teigiamu frontu skaitiklio turinys sumažėja vienetu. Konkreti skaitiklio turinio reikšmė prieinama CV išėjime. Funkcinio bloko išėjimo Q reikšmė tampa loginiu “1”, kai CV reikšmė ≤ 0 .



Sudėties ir atimties skaitiklis

CTUD skaitiklis yra sudėties ir atimties skaitiklių kombinacija. Išėjimo QU reikšmė nustatoma pagal sąlygą $CV \geq PV$, išėjimo QD reikšmė nustatoma pagal sąlygą $CV \leq 0$. Įsidėmėkite, kad skaitiklio atimties funkcija gali būti realizuota tik įkrovus skaitiklį paduodant loginį signalą į LD įėjimą.

