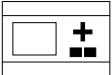
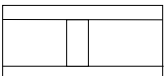
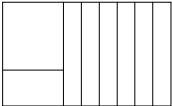
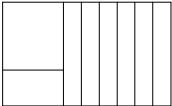


9 Programmeeritavate kontrolleri tarkvara ja käsustik

Sõltuvalt automaatikasüsteemi vajadustest võib SIMATIC kontrolleri programmeerimiseks kasutada eri tarkvara pakette. Enamasti kasutatakse eri jõudlusega kontrolleri programmeerimiseks eri tarkvarapakette, kuna kontrolleri kasutamise valdkonnad, vajadused ja võimalused on erisugused. Lisaks põhitarkvarale saab kontrolleri programmeerimiseks kasutada lisatarkvarapakette (joonis 9.1), mis on mõeldud spetsiaalselt otstarbeks nagu reguleerimiseks, diagnostikaks, simuleerimiseks, programmeerimiseks, häälestamiseks, optimeerimiseks jne.

Kontrolleri tüüp	Vajalik tarkvarapakett	Lisatav tarkvarapakett
LOGO! 	LOGO!Soft Comfort	Puudub
SIMATIC S7-200 	STEP7- Micro/DOS STEP7- Micro/WIN	Puudub
SIMATIC S7-300 	STEP7-Mini	Puudub
SIMATIC S7-300 	STEP7	S7-SCL CFC S7-GRAPH S7-HiGraph FuzzyControl Standard PID Control Modular PID Control DOCPRO TeleService S7-PDIAG S7-PLCSIM
SIMATIC S7-400	STEP7	PRODAVE MPI PC-DDE-Server

Joonis 9.1. Vajalikud ja lisatavad tarkvarapaketid

Lisatavate tarkvarapakettide kohta saab täpsustavat informatsiooni seadmete kataloogidest.

9.1 Programmeerimiskeeled

Maailmaturul tuntud firmade kontrollrite tarkvara struktuuri ja programmeerimise põhimõtted on määratud rahvusvahelise standardiga IEC 61131-3. Standardi järgi on kontrolleri programmeerimisel lubatud kasutada nt. järgmisi programmeerimiskeeli:

- kontaktaseskeemi *Ladder diagram (LD)* või *Ladder Logic (LAD)*
- algoritmi plokk skeemi ehk sammprogrammi *Sequential Function Charts (SFC)*
- loogikaskemi *Function Block Diagram (FBD)*
- kõrgkeelt *Structured Text (ST)* (C++ või **Pascal**’iga sarnane)
- käsulisti *Instruction List (IL)* või *Statement List (STL)*

Kontaktaseskeemina programmeerimine kujutab endast programmi graafilist esitust ja on mõeldud rele-kontaktor-juhtimislülituste hõlpsaks teisendamiseks kontrolleri programmiks. Kontaktaseskeem sarnaneb tavalise elektriskeemiga. Kui tavalises elektriskeemis kontakte, lüliteid ja täitureid ühendavad juhtmed paiknevad vertikaalsihis, siis kontaktaseskeemi puhul paiknevad nad horisontaalsihis. Kontaktaseskeemina programmeerimise eelisteks on *kasutajasõbralikkus* ja lihtsus. Tavaliselt saab kontaktaseskeemina programmeerimisel kasutada kõiki juhtimisoperatsioone ja käskte nagu teisteski esitusviisides (joonis 9.2 a).

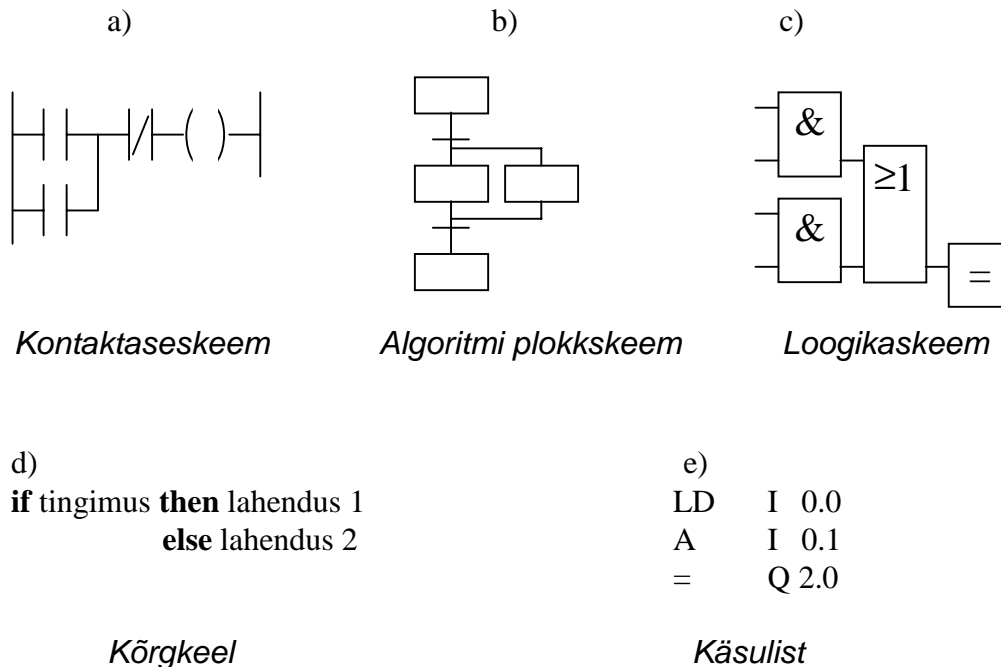
Programmeerimine algoritmi plokk skeemina kujutab endast graafilist programmeerimisviisi. Kaks põhikomponenti on seejuures operatsiooni- ja tingimusplokkid. Tingimusplokkid määravad ära järgneva operatsiooniploki. Iga operatsiooniplokk sisaldab üht või mitut jadamsi täidetavat operatsiooni. Tingimused määravad algoritmi kulgemise ja seepärast võib tingimusplokket nimetada ka ülekande- ehk siirdeplokkiks. Kui üldine juhtalgoritm on operatsiooni- ja tingimusplokkidena määratud, tuleb neid plokket programmeerida kas kontaktaseskeemina, käsulistina või kõrgkeeles (joonis 9.2 b).

Loogikaskemina programmeerimine on samuti graafiline programmeerimine (joonis 9.2 c). Loogikaskemina programmeerimisel on põhielemendiks *loogikaelement*. Sisendid paiknevad elementide vasakul, väljundid paremal küljel. Loogikaskemide puhul eristatakse põhilooikafunktsioone ja kasutaja loogikafunktsioone. Põhifunktsioonid on tarkvara tootja poolt evitatud enamlevinud loogikafunktsioonid. Kasutaja loogikafunktsioonid on programmeerija poolt mingi konkreetse ülesande täitmiseks defineeritud ja programmeeritud plokkid. Sellist programmeerimisviisi kasutatakse enamasti keerukate süsteemide ja juhtsüsteemide sageli esinevate juhtahelate programmeerimiseks.

Üldlevinud **kõrgkeeles programmeerimine** (joonis 9.2 d) sisaldab kõiki tänapäeva kõrgkeeltes esinevaid põhikäskte nagu valikukäskte (IF-THEN-ELSE and CASE OF) ja tsüklike käskte (FOR, WHILE and REPEAT) jne. Tänu selle võimaluse lülitamisele tööstuskontrolleritesse on inimesel, kes kasutab teiste automaatika süsteemide programmeerimisel kõrgkeeli C++, Pascal või Basic, suhteliselt lihtne programmeerida ilma ümberõppimiseta ka tööstuskontrollereid. Tänu kasutatavale kõrgkeelele muutuvad tööstuskontrollerid teiste automaatjuhtimisseadmetega ühtseks tervikuks.

Käsulistina programmeerimine (joonis 9.2 e) kujutab madala nivoo keeles programmeerimist. Käsulistis esitatakse programm, vastupidi kontaktaseskeemile ja

loogikaskeemile, tekstina. Käsulistina programmeerides lahendatakse kogu juhtimisülesanne üksikute käskude jadana. See võimaldab teha lühemaid ja kiiremaid programme, kuna saab paremini kasutada mälu mahtu. Võrreldes teiste keeltega on ta raskemini õpitav, kuna teistes keeltes (nagu kontaktaskeem või loogikaskeem) hõlmab üks plokk sageli mitut konkreetseks operatsiooniks vajalikku käsulisti käsku.



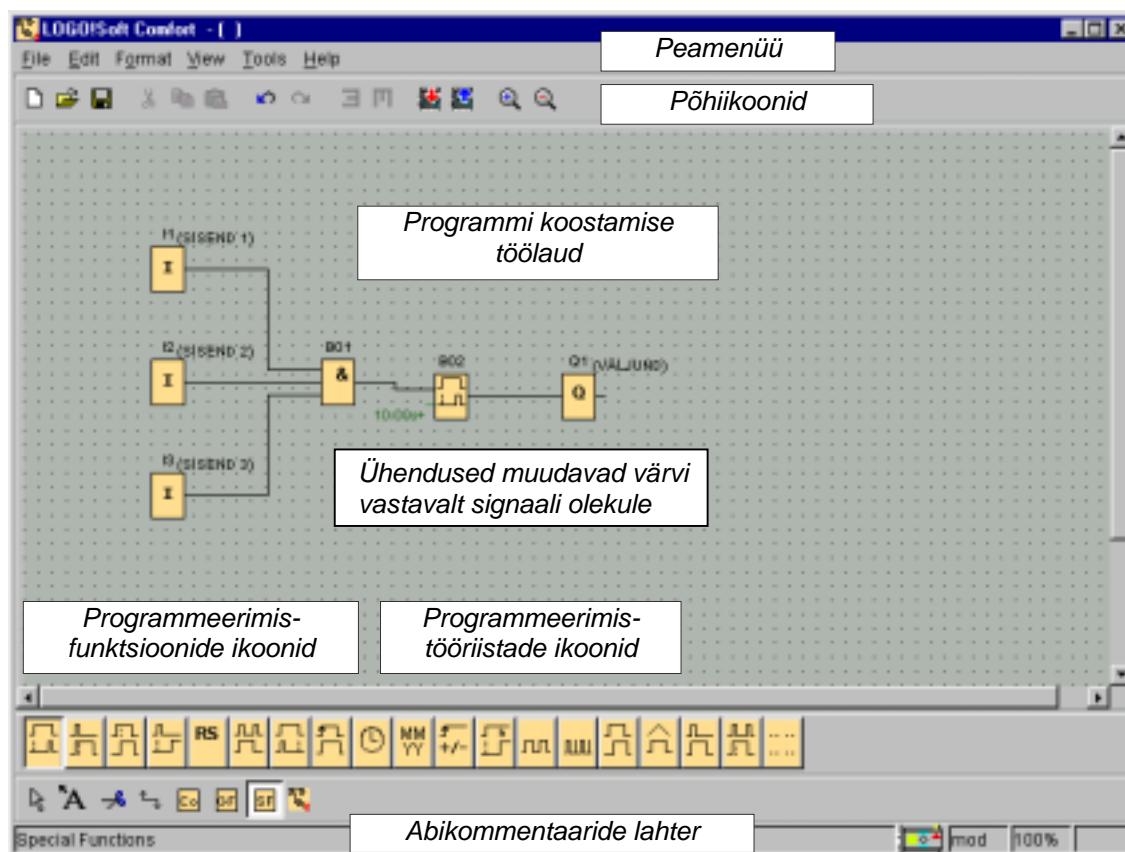
Joonis 9.2 Standardiseeritud programmeerimiskeeled

Rahvusvahelise standardi IEC 61131 eelisteks on see, et ta lubab ühe ja sama kontrolleri puhul kasutada mitmeid eri programmeerimisviise. See tähendab, et igauks leiab enda võimetele vastava programmeerimiskeele.

*SIEMENS*i programmeeritavate kontrollrite SIMATIC S7/M7/C7 puhul saab kõiki eelmainitud esitusviise kasutada vastava tarkvarapaketi omamise korral. Ka teiste firmade kontrollrite tarkvara võimaldab kasutada vähemalt kolme erinevat programmeerimisviisi (kontaktaskeemi, loogikaskeemi ja käsulisti).

9.2 Kompaktkontrollerite tarkvara

LOGO!Soft Comfort erineb oma eelkäijast *LOGO!Soft* selle poolest, et ta võimaldab graafilisel kujul programmeerimist. Programmeerimisviisina kasutatakse loogikaskeemi. Tarkvaraga *LOGO!Soft* programmeerimine sarnanes *LOGO!* esipaneelil paikneva kasutajaliidese kaudu programmeerimisega, mis ei võimaldanud piisavat ülevaadet kogu programmist.



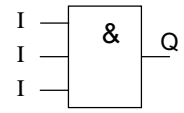
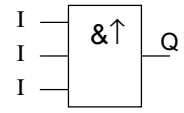
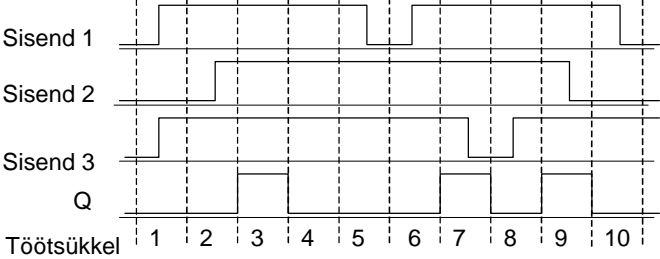
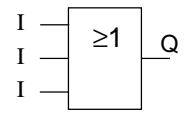
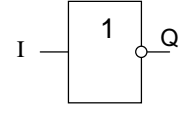
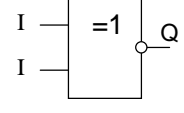
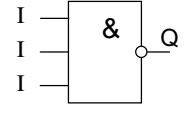
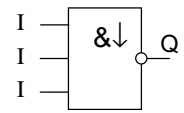
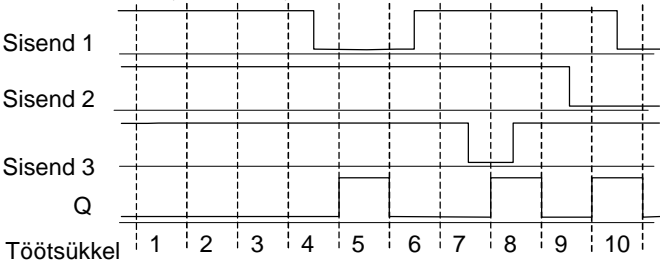
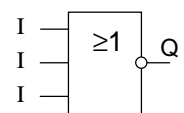
Joonis 9.3. Tarkvarapaketi LOGO!Soft Comfort tööaken

LOGO!Soft Comfort tööakna (joonis 9.3) vasakul all servas paiknevad tööriistade ikoonid, mille abil saab redigeerida joonist ja valida vastav funktsioonide kataloog nagu

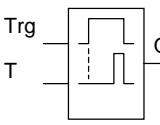
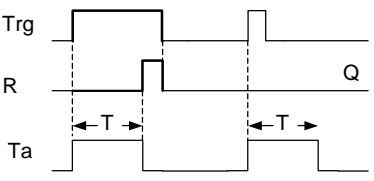
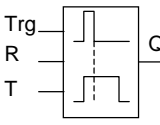
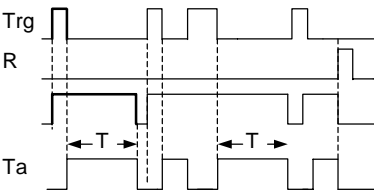
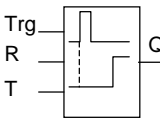
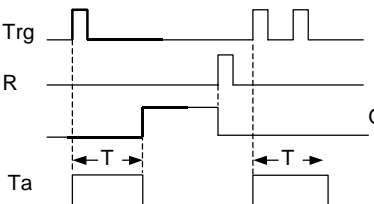
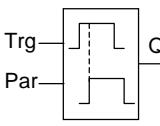
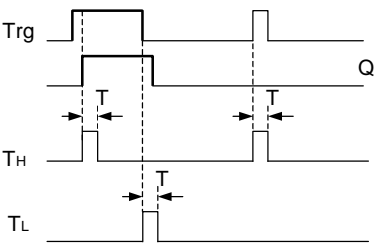
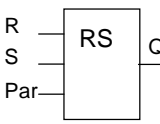
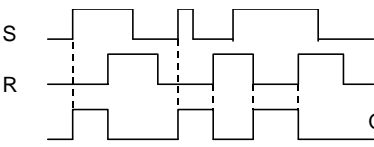
- Co (kontaktid), mis sisaldab elemente sisendite ja väljundite (I-sisend, Ia – ASI sisend, Q - väljund, Qa – ASI väljund, M - lipp, B - plokk, AI - analoogsisend) tähistamiseks skeemil
- GF (põhifunktsioonid), mis sisaldab lihtloogikalülitusi NING, VÕI, EI, NING-EI, VÕI-EI
- SF (erifunktsioonid), mille abil saab valida erifunktsioone nagu rakendusviide, tagastusviide, triger, taktrelee, salvestav rakendusviide, asünkroonne taktrelee, nädala ja aasta kell, impulsiloendur, tunniloendur, impulsi tõusvale frondile reageeriv lülitus, impulssrelee jne. Erifunktsioone lisatakse LOGO!-le pidevalt juurde. Täpsemaid andmeid nende kohta leiab käsiraamatutest.

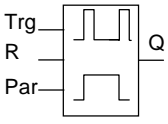
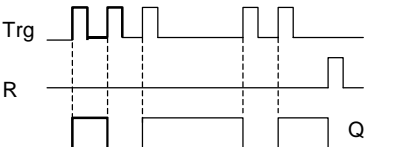
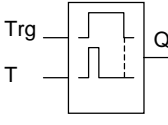
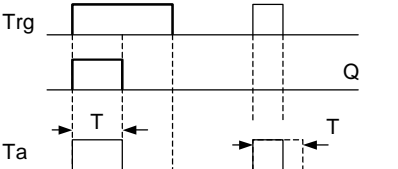
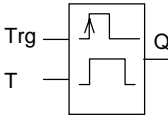
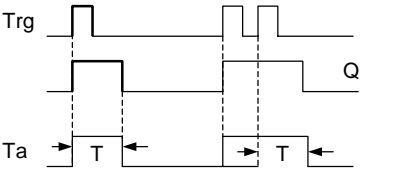
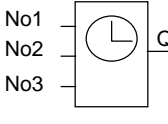
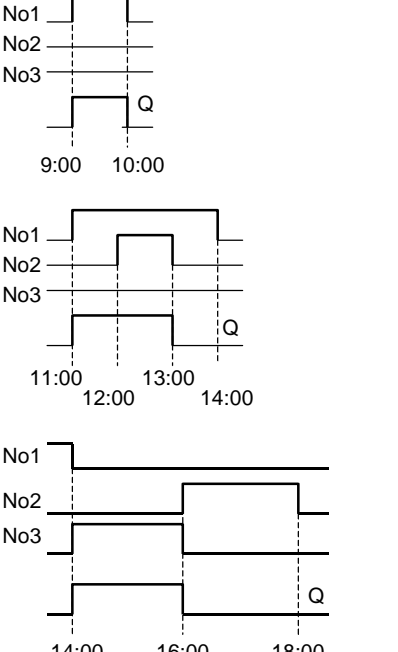
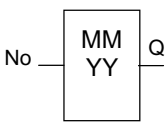
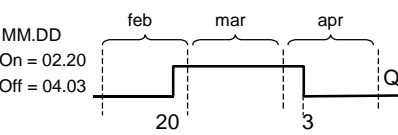
Programmi saab testida ka LOGO! olemasoluta menüüst *Tools* ➤ *Simulation* (joonis 9.4). Pärast simuleerimist, olles veendunud programmi õigsuses, saab programmi vastava kaabli olemasolul laadida LOGO!-sse, valides menüüst *Tools* ➤ *Transfer* ➤ *PC?Logo*. Tarkvara võimaldab ka üsna lihtsalt kindlaks määrata, millist LOGO! antud programmi jaoks valida; selleks tuleb valida vaid menüüst *Tools* ➤ *Determine Logo !* ning vastav teade väljastatakse akna alla serva.

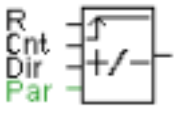
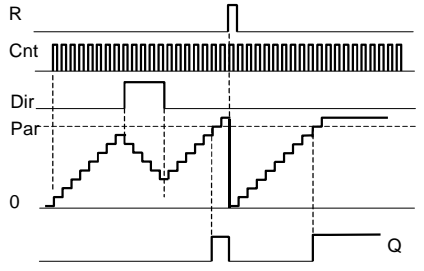
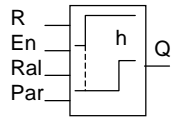
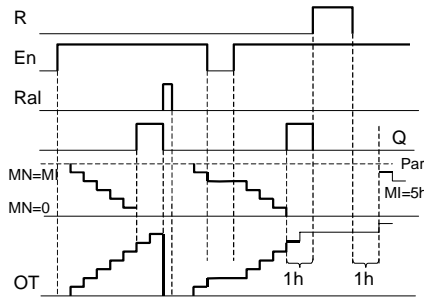
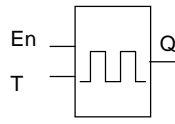
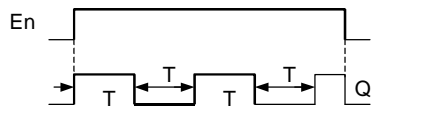
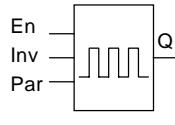
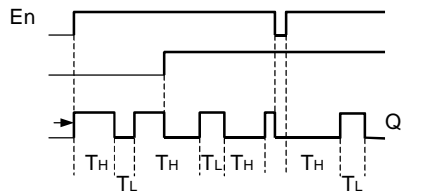
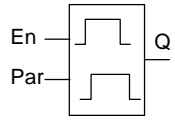
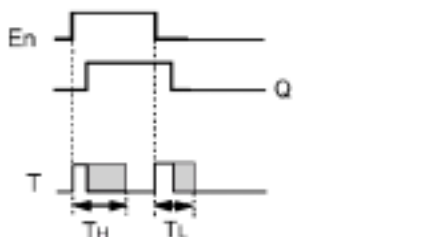
Põhifunktsioonid

Element	Kirjeldus
	<p>Kolme sisendiga NING (AND)</p> <p>Väljund Q on sisselülitatud (olek "1") vaid juhul kui kõik sisendid on sisselülitatud (olek "1").</p>
	<p>Kolme sisendiga ja pulsi tõusule reageeriva impulssvälundiga NING ↑ (AND↑)</p>  <p>Töötssükkel 1 2 3 4 5 6 7 8 9 10</p>
	<p>Kolme sisendiga VÕI (OR)</p> <p>Väljund Q on väljalülitatud (olek "0") vaid juhul kui kõik sisendid on väljalülitatud (olek "0").</p>
	<p>EITUS (NOT)</p> <p>Väljundi Q olek vastupidine sisendi olekule. Kui sisend väljalülitatud (olek "0"), siis on väljund sisselülitatud (olek "1") või vastupidi.</p>
	<p>Kahe sisendiga VÄLISTAV-VÕI (XOR)</p> <p>Väljund Q on sisselülitatud vaid siis (olek "1") kui üks sisend on sisselülitatud (olek "1").</p>
	<p>Kolme sisendiga NING-EI (NAND)</p> <p>Väljund Q on väljalülitatud (olek "0") vaid juhul kui kõik sisendid on sisselülitatud (olek "1").</p>
	<p>Kolme sisendi ja pulsi langule reageeriva impulssvälundiga NING-EI ↓ (NAND↓)</p>  <p>Töötssükkel 1 2 3 4 5 6 7 8 9 10</p>
	<p>Kolme sisendiga VÕI-EI (NOR)</p> <p>Väljund Q lülitatakse sisse (olek "1") vaid juhul kui kõik sisendid on välja lülitatud (olek "0").</p>

Erifunktsioonid

Element	Ajadiagramm	Kirjeldus
		<p>Sisselülitusviivitus (On-Delay)</p> <p><i>Trg</i> sisendis tuvastatav pulsitõus käivitab sisendis <i>T</i> etteantud aja <i>Ta</i> lugemise. Väljund <i>Q</i> lülitub sisse aja <i>T</i> möödumisel kui sisendi impulsi olek on aja möödudes "1". Väljund lülitub välja kui tuvastatakse sisendipulsi pulsilang.</p>
		<p>Väljalülitusviivitus (Off-Delay)</p> <p>Sisendis <i>Trg</i> tuvastatav pulsitõus viib väljundi <i>Q</i> olekusse "1". Väljundi <i>Q</i> olek muutub "0"-ks, kui sisendis <i>Trg</i> tuvastatud pulsilangu poolt käivitatud taimer annab vastavalt kasutaja poolt sisendis <i>T</i> etteantud ajale <i>Ta</i> selleks loa. Sisendi <i>R</i> abil on võimalik suvalisel ajahetkel viia väljundi olek "0".</p>
		<p>Salvestav sisselülitusviivitus (Retentive On Delay)</p> <p>Sisendisse <i>Trg</i> antava impulsi tõusuga käivitatakse sisendis <i>T</i> etteantud aja <i>Ta</i> lugemine. Aja <i>Ta</i> möödudes viiakse väljund <i>Q</i> olekusse "1". Aja <i>Ta</i> sisse mahtuva mitme järjestikuse impulsi korral käivitatakse aja lugemine esimese impulsi tõusuga. Sisendi <i>R</i> abil on võimalik suvalisel ajahetkel viia väljundi <i>Q</i> olek "0".</p>
		<p>Sisse-/väljalülituse viivitus (On/Off Delay)</p> <p>Kui sisendis <i>Trg</i> tuvastatakse pulsi tõus hakatakse lugema sisendis <i>Par</i> etteantud aega T_H, aja möödumisel lülitatakse väljund <i>Q</i> sisse (olek "1"). Kui sisendis <i>Trg</i> tuvastatakse pulsi lang hakatakse lugema sisendis <i>Par</i> etteantud aega T_L, aja möödudes lülitatakse väljund <i>Q</i> välja (olek "0").</p>
		<p>SR-triger (Latching Relay)</p> <p>Sisendisse <i>S</i> antava impulsi tõusuga lülitatakse väljund <i>Q</i> sisse (olek "1") sõltumatult väljundi eelnevast olekust. Sisendisse <i>R</i> antava impulsi tõusuga lülitatakse väljund <i>Q</i> välja (olek "0") sõltumatult väljundi eelnevast olekust. Samaaegse impulsi tõusu tuvastamisel sisendites <i>S</i> ja <i>R</i> lülitatakse väljund <i>Q</i> välja sõltumatult väljundi eelnevast olekust. Sisendiga <i>Par</i> määratakse väljundi <i>Q</i> käitumine pingekatkestuse korral, kas säilitatakse peale pingekatkestust väljundi eelmine olek või ei.</p>

		<p>Impulsslülitus (Current Impulse Relay) Väljundi Q olekut muudetakse sisendisse Trg antava impulsi tõusuga. Sisendi R abil on võimalik suvalisel ajahetkel lülitada väljund Q välja (olek "0"). Sisendiga Par määratakse pingekatkestusele eelnenud oleku talletus.</p>
		<p>Impulssviivitus (Interval Time-Delay Relay/Pulse Output) Sisendisse Trg antava impulsi tõusuga lülitatakse väljund Q sisse (olek "1") ja käivitatakse sisendiga T etteantud aja Ta lugemine. Kui sisendisse Trg antava impulsi pikkus on lühem etteantud ajast Ta, lülitatakse väljund Q välja vastavalt sisendimpulsi Trg languga. Kui sisendimpulss Trg on pikem etteantud ajast Ta, siis lülitatakse väljund Q välja aja Ta möödumisel.</p>
		<p>Pikendatud ehk salvestav impulssviivitus (Edge Triggered Interval Time-Delay Relay) Sisendisse Trg antava impulsi tõusuga lülitatakse väljund Q sisse (olek "1") ja käivitatakse sisendiga T etteantud aja Ta lugemine. Väljund Q lülitatakse välja (olek "0") aja Ta möödumisel.</p>
		<p>Nädalakell (Seven-Day Time Switch) Sisendite $No1$, $No2$ ja $No3$ abil saab kombineerida 3 eri lülitusvarianti väljundi Q sisse- ja väljalülitamiseks kellaajaliselt vastavalt nädalapäevadele. Kellaajad on sätestatavad vahemikus 00.00...23.59 ja nädalapäevad vastavalt esmas-päevast pühapäevani. Päevi tähistatakse järgmiste lühendite abil.</p> <p>M – esmaspäev, T – teisipäev, W – kolmapäev, T – neljapäev, F – reede, S – laupäev, S – pühapäev</p>
		<p>Aastakell (Twelve-Month Time Switch) Sisendiga No sätestatakse kuupäevaliselt väljundi Q sisse- ja väljalülitamine</p>

		<p>Loendur (Up/Down Counter)</p> <p>Sisendisse R antava impulsi tõusuga viiakse nullitakse loendussuurus ja lülitatakse väljund Q välja (olek "0"). Sisendiga Cnt loendatakse kuni 5 kHz sagedusega impulsse ja iga impulsi tõusu korral suurendatakse või kahandatakse loenduri sisu ühe võrra. Sisendisse Dir antav impulsi olek määrab loendamise suuna ehk loendussuure suurendamise või kahandamise. Sisendiga Par määratakse loendusväärtuse piir, mida ületades lülitatakse väljund Q sisse ja piirist allapoole langedes välja.</p>
		<p>Tunniloendur (Hours Counter)</p> <p>Sisendiga Par antakse ette loendatavate tundide arv, millest alates lülitatakse väljund Q sisse. Sisendi R abil peatatakse vastavalt impulsi pikkusele tundide loendamine, kuid ei nullita tundide loendurit ja lülitatakse väljund Q välja. Sisendi Ral abil nullitakse tundide loendur ja lülitatakse väljund Q välja. Tundide loendamine käivitatakse sisendisse En antava impulsi tõusuga ja peatatakse impulsi languga.</p>
		<p>Sümmeetriline impulssgeneraator (Symmetrical Clock Pulse Generator)</p> <p>Sisendimpulsi En tõusuga (olek "1") käivitatakse impulsside genereerimine, sisendimpulsi languga (olek "0") lõpetatakse impulsside genereerimine. Aeg T on genereeritavate impulsside poolperioodi aeg.</p>
		<p>Asünkroonne impulssgeneraator (Asynchronous Pulse Generator)</p> <p>Sisendisse En antava impulsi tõusuga käivitatakse ja impulsi languga seisatakse impulssgeneraator. Sisendiga Par antakse ette ülemine ja alumine poolperioodide aeg TH ja TL. Sisendiga Inv vahetatakse ülemise ja alumise poolperioodi ajad omavahel.</p>
		<p>Juhuslik sisse-/väljalülituse viivitus (Random Generator)</p> <p>Sisendis En tuvastatud impulsi tõus käivitab sisselülitusviivituse mille aeg leitakse juhuslikult vahemikus $0...T_H$ sekundit ja mille möödumisel väljund Q lülitatakse sisse. Sisendis En tuvastatud impulsi lang käivitab väljalülitusviivituse mille aeg leitakse juhuslikult vahemikus $0...T_L$ sekundit ja mille möödumisel väljund Q lülitatakse välja. Aeg TH ja TL määratakse sisendi Par kaudu.</p>

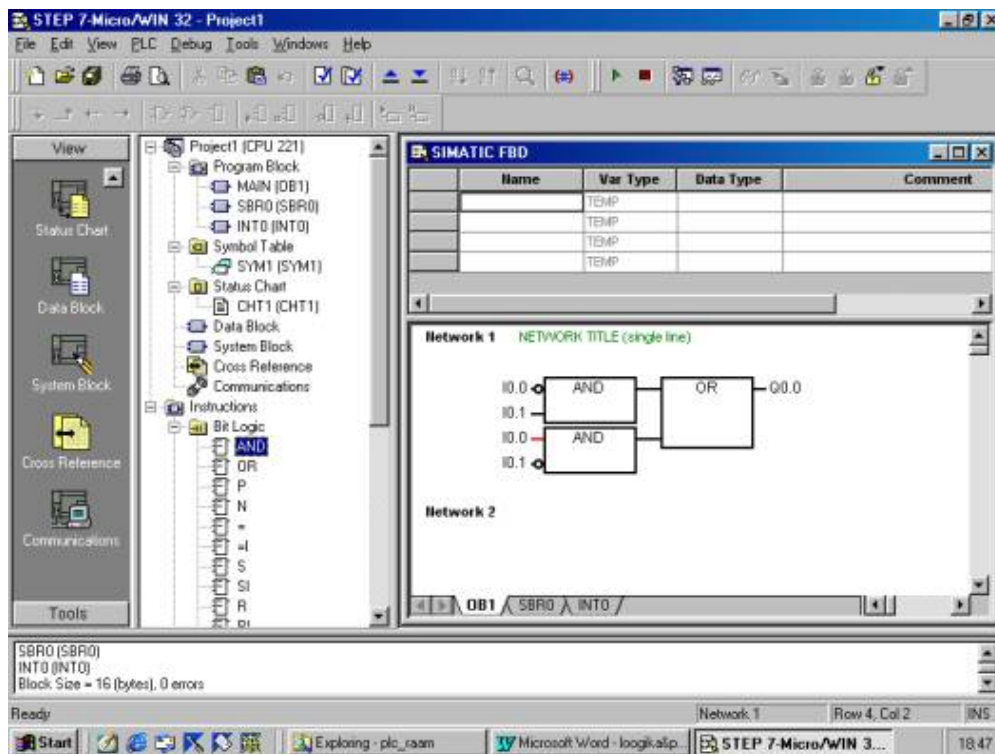
		<p>Sageduskäiviti (<i>Frequency Trigger</i>) Väljund Q lülitatakse sisse või välja vastavalt sisendis Fre ajas G_T mõõdetud sagedusest ja sisendiga Par etteantud sagedusvahemikust $SW\uparrow$ ja $SW\downarrow$. Kui G_T sisendis Fre mõõdetud sagedus on suurem või võrdne sagedusega $SW\uparrow$, siis lülitatakse väljund Q sisse. Aga, kui ajas G_T sisendis Fre mõõdetud sagedus on väiksem või võrdne sagedusega $SW\downarrow$, siis lülitatakse väljund Q välja.</p>
		<p>Analoog-käiviti (<i>Analog Trigger</i>) Väljund Q lülitatakse sisse kui sisendis Ax mõõdetud analoogsuuruse väärtus ületab sisendiga Par etteantud väärtust $SW\uparrow$ ja Väljund Q lülitatakse välja kui sisendis Ax mõõdetud analoogsuuruse väärtus langeb alla poole sisendiga Par etteantud väärtust $SW\downarrow$.</p>
		<p>Analoogsignaali komparaator (<i>Analog Comparator</i>) Väljund Q lülitatakse sisse kui kahe analoogsuuruse Ax (AI1-sisend I7) ja Ay (AI2-sisend I8) väärtused erinevad üksteisest rohkem kui sisendiga Par etteantud Δ väärtus.</p>
		<p>Stairwell'i valgustuse lüüti (<i>Stairwell Light Switch</i>) Sisendimpulsi Trg tõusuga käivitatakse aeg Ta ja lülitatakse väljund Q sisse. 15 sekundit enne aja Ta lõppu lülitatakse väljund Q üheks sekundiks välja ja seejärel uuesti sisse. Kui aeg Ta lõppeb lülitatakse väljund välja. Kasutatakse trepikodade valgustuseks.</p>
		<p>Multifunktsionaalne lüüti (<i>Dual-Function Switch</i>) Eristatakse kaht lülitusvarianti. 1) Väljundit Q saab sisse ja välja lülitada sisendisse Trg antavate impulsside tõusudega. Kui väljalülitamist sisendisse Trg antava impulsi tõusuga ei tehta, siis lülitub väljund Q automaatselt aja T_H möödudes välja. 2) Väljundit Q saab lülitada sisse ajast sõltumatult kui anda sisendisse Trg impulss pikkusega T_L, sellisel juhul saab väljundit Q uuesti välja lülitada vaid sisendisse Trg antava impulsi tõusuga.</p>
		<p>Teated ekraanil (<i>Message Texts</i>) Väljund Q lülitatakse sisse ja välja samaselt sisendi En olekule. Sisendiga P määratakse tekstide esitamise prioriteet ja sisendiga Par ekraanile esitatav tekst.</p>

9.3 Moodulkontrollerite tarkvara

Uue projekti alustamisel *MicroWIN* tarkvaraga tekib viie jaotusega tööaken (joonis 9.5):

- programmeerimise aken (*SIMATIC Ladder*, *FBD* või *STL*),
- programmeerimise akna kohal muutujate defineerimise aken
- süsteemsete operatsioonide menüü ekraani vasakus servas
- projekti haldamise kataloog ekraani keskel
- teadete aken all

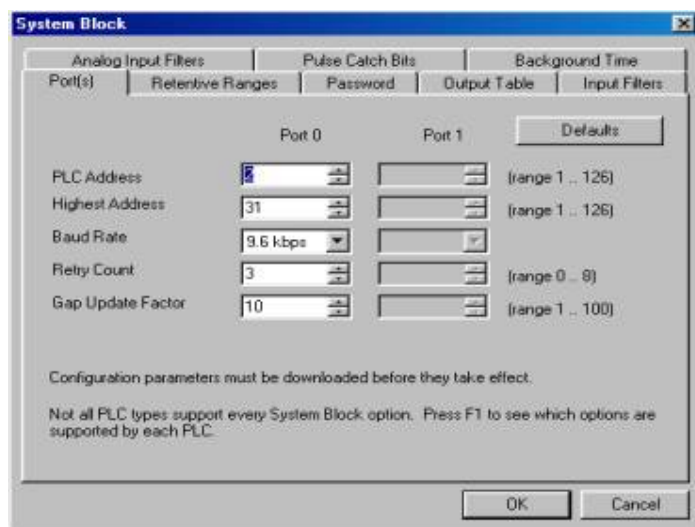
Programmeerimise aknas ekraani paremal servas saab kirjutada programmi ja määrata lokaalseid mälumuutujaid. Süsteemsete operatsioonide menüüst on võimalik jälgida sidustalitluses kontrolleri sisend-, väljund- jt. muutujate olekudiagrammi (*Status Chart*), redigeerida andmeplokki (*Data Block*), sätestada kontrolleri süsteemseid parameetreid (*System Block*), andmesideparameetreid sätestada ja andmeside olekut kuvada (*Communications*) jne. Projekti haldamise aken on programmeerija pidev abivahend kogu projekti haldamisel, programmiplokkide programmeerimisel, sümboltabeli koostamisel jne. Projekti haldamise omab osalielt samasid valikuvõimalusi mis süsteemsete operatsioonide menüü. *MicroWIN* tarkvara võimaldab kasutada kolme programmi esitusviisi, milleks on kontaktskeem ja käsulist, mida saab valida menüüst vastavalt *View* ➤ *STL*, *View* ➤ *Ladder* jne. *S7-200* seeria kontrolleri programmeerimise käsud erinevad teataval määral *S7-300/400* programmeerimiskäskudest, kuid vastavad rahvusvahelisele standardile IEC 61131 ning on kiiresti omandatavad.



Joonis 9.5. Tarkvara *MicroWIN* tööaken

Pärast programmi koostamist ja enne kontrollerrisse laadimist tuleb ta kompileerida menüüst *PLC* ➤ *Compile*, mille tulemusel väljastatakse teade, et programmis on või ei ole vigu. Pärast kompileerimist võib programmi kontrollerrisse laadida *File* ➤ *Download*, olles eelnevalt viinud kontrolleri talitlusse *STOP*. Tähelepanu tuleb pöörata sellele, et kontrolleri küljes paiknev talitlusviisi lüliti oleks õiges asendis. Programmi töö kontrollimiseks pärast kontrollerrisse laadimist tuleb valida menüüst talitus *PLC* ➤ *Run*. Nüüd saab arvuti ekraanil ja kontrolleri valgusdioodindikaatorite abil jälgida ja testida programmi tööd vastavate sisendite ja väljundite eri olekute korral.

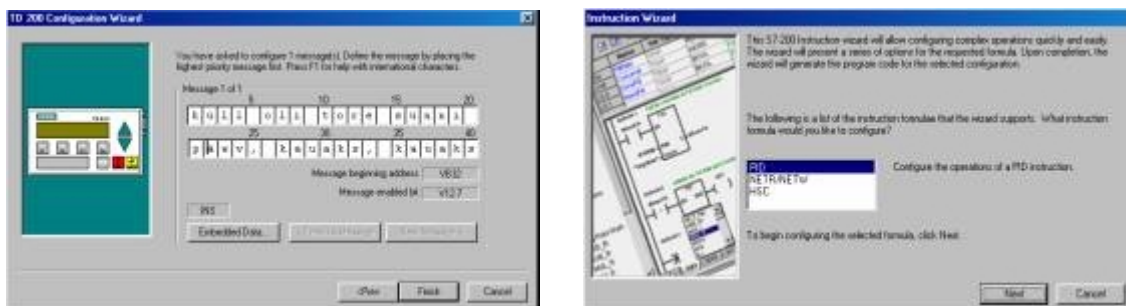
MicroWIN tarkvara võimaldab ka konfigurereida ja sätestada kontrollerrile mitmesuguseid parameetreid nagu andmesidevõrgu aadressid, juurdepääsu privileegid programmile koos salasõnadega jne (joonis 9.6).



Joonis 9.6. Parameetrite määramine

Samuti võimaldab *MicroWIN* kasutajal *online*-talitluses lugeda eri informatsiooni protsessoris toimuva kohta (nt. talitusviis, protsessori talitusviis, programmi tsükli aeg jne.).

MicroWIN versiooni 3.2 on juba integreeritud tekstipaneeli *TD200* konfigureerimiskeskond *TD 200 Configuration Wizard*, mille abil on lihtne kirjeldada paneelile edastatavaid teateid ja mis seeläbi lihtsustab tekstipaneeli programmeerimist (joonis 9.7).



Joonis 9.7. Tekstipaneeli teadete ja keerukate operatsioonide (PID, network, kõrgsagedusloenduri) kirjeldamise aken

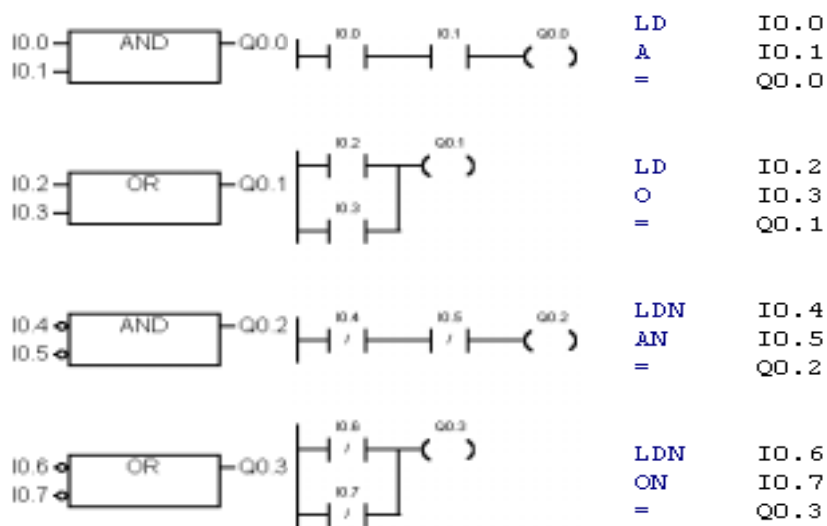
9.4 STEP 7 käsustik

Iga programmi koostamise aluseks on ülesande kirjeldus, milleks võib olla tehnoloogiaskeem, juhtimise graafiskeem või tekstina kirjeldatud algoritm. Ülesande kirjeldus tuleb viia seega kontrolleriile mõistetavale kujule ehk *programmeerida*. Programmeerimiseks saab kasutada mitmeid kontrolleriile arusaadavaid esitusviise (programmeerimiskeeli). Käesolevas peatükis käsitletakse kolme enamkasutatavat programmi esitusviisi ehk keelt – loogikaskeemi, kontaktaseskeemi ja käsulist.

- bittoperatsioonide käsud,
- siirdamiskäsud,
- võrdluskäsud,
- loenduskäsud
- viivituskäsud,
- aritmeetikakäsud,
- teisenduskäsud,
- nihkekäsud sh. ringnihkekäsud,
- programmi struktuurikäsud,
- jne.

Bitioperatsioonide käsud

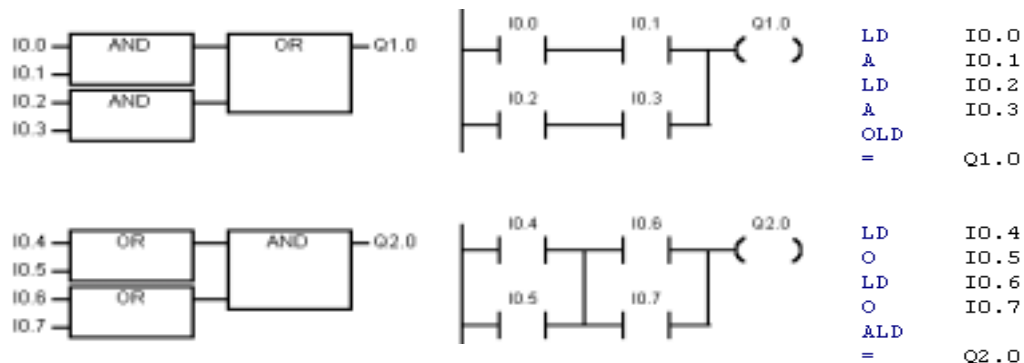
Loogikakäske kasutatakse lihtsate binaarfunktsioonide programmeerimiseks (nt. NING, VÕI, EI ja nende kombinatsioonid). Programmeerimistarkvara STEP 7 MicroWIN võimaldab programmeerida kolmel erineval viisil. Järgnevas peatükikes vaadeldakse põhiliselt kolme erinevat programmeerimisviisi – loogikaskeem, kontaktaseskeem ja käsulist.



Joonis 9.8. Lihtloogikakäsud

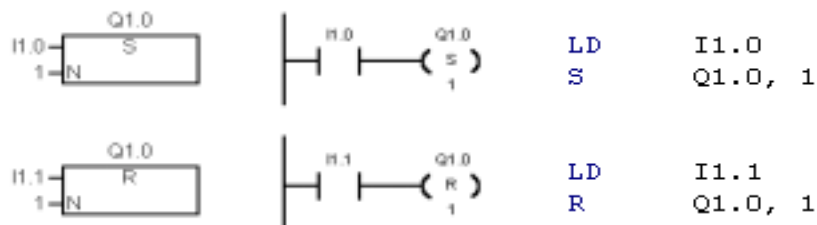
Joonisel 9.8 on esitatud loogikaskeemina, kontaktaseskeemina ja käsulistina NING-, VÕI-, inverteeritud sisenditega NING- ja VÕI-lüli. NING-lüli tööd iseloomustab see, et väljundis on olek "1" ainult siis, kui kõigis sisendites on olek "1". VÕI-lüli tööd iseloomustab see, et

väljundis on olek "1", kui kasvõi ainult ühes sisendis on olek "1". Sisend ja väljundahelate kohale kirjutatakse operandide koodid. Seega esimesele sisendile vastab kontrolleri sisend I aadressiga 0.0, teisele vastavalt sisend I aadressiga 0.1 ning väljundile kontrolleri väljund Q aadressiga 0.0.

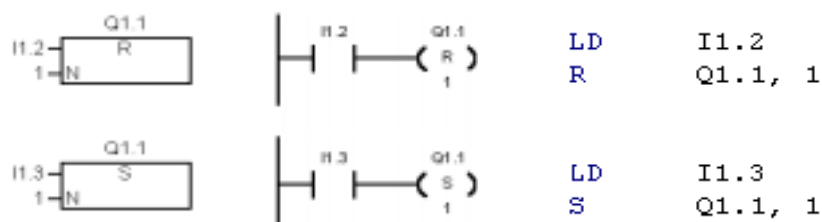


Joonis 9.9. NING-VÕI ja VÕI-NING loogikühend

Programmeerimise puhul eristatakse termineid *NING-VÕI-loogikühend (NV)* ja *VÕI-NING-loogikühend (VN)*. NV kujutab endast rööbiti (VÕI-lülige) ühendatud NING-lülisid, kus kõigepealt teostatakse NING-tehe ja seejärel VÕI-tehe, nagu on näha joonisel 9.9 kolmes erinevas programmeerimiskeeles. VN kujutab endast jadamisi (NING-lülige) ühendatud VÕI-lülisid, kus kõigepealt teostatakse VÕI-tehe ja seejärel NING-tehe, nagu on näha joonisel 4.9 kolmes erinevas programmeerimiskeeles. Bitioperatsioonide hulga kuuluvad ka olekut salvestava funktsiooniga käsud ehk mälu funktsioonid. Tuntuim mälu funktsiooniga loogikaseade on triger. Programmeerimiskeeltes STEP 5 ja STEP 7 kasutatakse RS- ja SR-trigerit. RS-triger on ülilusliku sisselülitamise, SR-triger on ülilusliku väljalülitamise (joonis 9.10 ja 9.11).



Joonis 9.10. Ülilusliku väljalülimisega mälu funktsioon

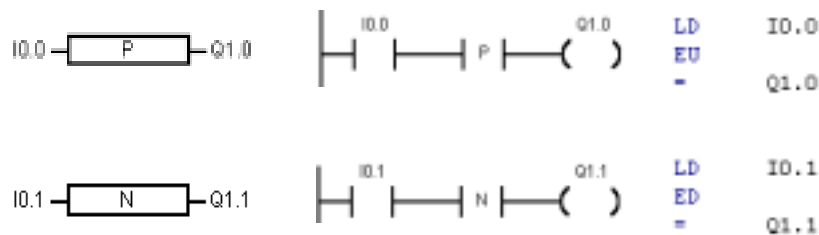


Joonis 9.11. Ülilusliku sisselülimisega mälu funktsioon

Mõlemal trigeril seatakse väljund olekusse "1", kui sisendisse S (*set*) antakse signaal olekuga "1"; väljund seatakse olekusse "0", kui sisendisse R (*reset*) antakse signaal olekuga "1". SR-

trigerit iseloomustab see, et kui samaaegselt anda trigeri R- ja S- sisendile signaal olekuga “1”, siis trigeri väljund viiakse olekusse “0”. RS-trigeri puhul põhjustab signaal olekuga “1” mõlemas sisendis trigeri väljundi siirdumise olekusse “1”.

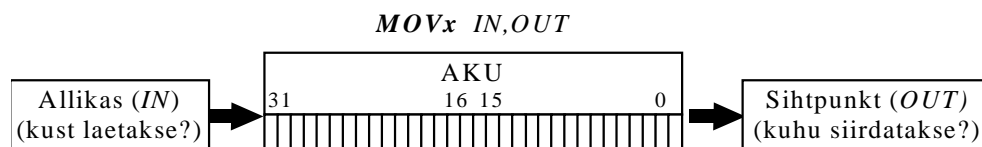
Samuti kuuluvad bittoperatsioonide hulka impulsi tõusule ja langule reageerivad lülid. Näiteks alltoodud joonise 9.12 toodud programmi järgi väljastatakse väljundisse Q1.0 kontrolleri ühe tsükli pikkune impulss kui sisendis IO.0 tuvastatakse impulsi tõus ja väljundisse Q1.1 väljastatakse kontrolleri ühe tsükli pikkune impulss kui sisendis IO.1 tuvastatakse impulsi lang.



Joonis 9.12 Impulsi tõusu ja langu tuvastavad käsud

Siirdamiskäsud

Programmeerimiskeel STEP 7 võimaldab baidi, sõna ja topeltsõna pikkust infovahetust, sisend- ja väljundkaartide, sisend- ja väljundprotsessi kuva, loendurite, taimerite, mälu, andmeplokkide jms. vahel. Selleks kasutatakse siirdamiskäsku (**MOV**). Selline infovahetus ei toimu otse, vaid alati vahemälu (AKU) kaudu. Vahemälu kujutab endast ühte protsessori registrit. Laadimis- ja siirdamiskäsu toimet selgitab joonis 9.13.

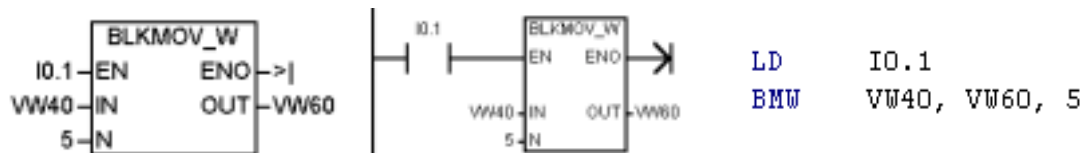


Joonis 9.13. Siirdamistoiming

Seega mingi info laetakse teatud allikast (nt. sisendmoodulist) vahemällu ning siiratakse sealt sihtpunkti (nt. väljundmoodulisse). Infoallikaks või sihtpunktiks võivad olla sisendid, väljundid, mälad, loendurid, taimerid, andmeplokkid jne. Siirdamiskäsud on *absoluutkäsud* ning neid töödeldakse kontrolleri iga töötsoikli jooksul. Neid käskude tähistus käsulistis on MOV, millele järgneb operandi tunnus ehk otsene või kaudne aadress, st. kust laetakse ja kuhu saadetakse.



Joonis 9.14. Baidi siirdekäsk

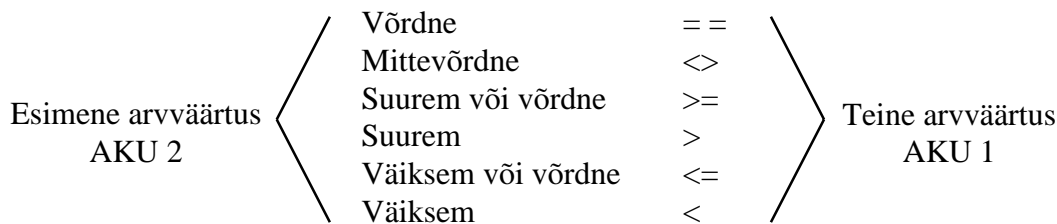


Joonis 9.15. Ploki siirdekäsk

Joonisel 9.14 esitatud käsu MOVB tulemusel laetakse sisendbaidis IB10 olev väärtus vahemällu ja seejärel saadetakse ta mällu aadressile VB50. Joonisel 9.15 esitatud käsk BMW erineb joonisel 9.13 toodud käsust selle poolest, et ta võimaldab suurema hulga andmeid siirata ühest mälu piirkonnast teise. Näiteks laetakse eeltoodud joonisel 5 sõna alates aadressist VW40 (kuni VW48) ja siiratakse mällu aadressile VW60 (kuni VW68).

Võrdluskäsud

STEP 7 keeles saab võrrelda operande (arve), mille tunnuseks on sisend, väljund, konstant, mälu ja andmeplokk ja mis on esitatud kas sõna- (**I**) ja topeltsõnapikkuse täisarvu (**D**) või topeltsõnapikkuse reaalarvu (**R**) kujul. Kõigi nende arvutüüpide võrdlemiseks saab kasutada kuut erinevat võrdluskäsku (joonis 9.16), kusjuures võrreldakse esimest arvvaartust teise arvvaartuse suhtes.



Joonis 9.16. Võrdluskäsud

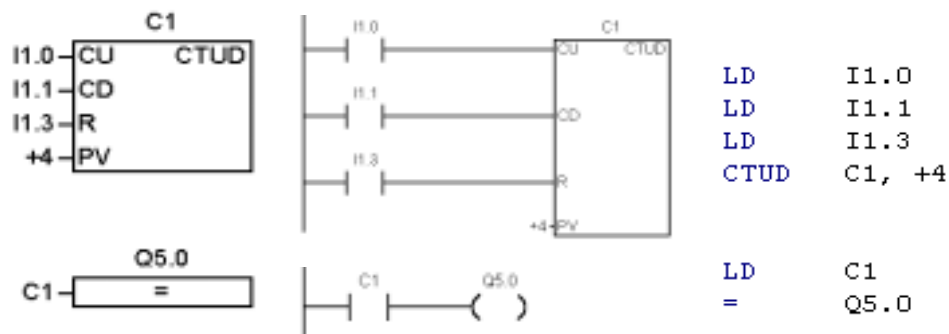
Võrdluskäsu abil võrreldakse kahte operandi omavahel. Joonisel 9.17 toodud näites võrreldakse kas mälus aadressil VW10 paiknev suurus on vördne konstandiga 10. Kui tingimus on tõene siis väljund Q2.0 lülitatakse sisse (signaali olek "1"). Vastupidisel juhul, kui tingimus on väär, lülitatakse väljund välja (signaali olek "0")



Joonis 9.17. Võrdlustehe

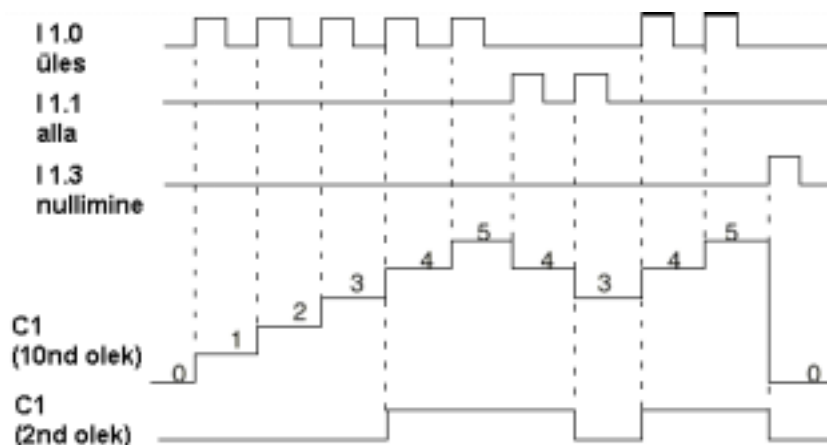
9.4.1 Loenduskäsud

Protsessides on sageli vaja loendada sündmusi või esemeid, nt. ajaimpulse, tooteid, detaile vms. Sedalaadi loendamiseks kasutatakse programmeerimisekeeles STEP 7 loenduskäskude ehk graafiliste esitusviiside puhul loendureid. Loenduritega (joonis 9.18) saab loendada arve alates -32768 kuni 32767.



Joonis 9.18. Loenduskäsk

Loenduri tööpõhimõtet selgitab alljärgnev joonis 9.19. Sisendisse I1.0 (loenduri CU sisend) antava impulsi kasvatatakse loendusväärtust ühe võrra ja sisendisse I1.1 (loenduri CD sisend) antava impulsi kahandatakse loendusväärtust ühe võrra. Sisendisse I1.3 (loenduri R sisend) antava impulsi nullitakse loendusväärtus. Loenduri sisendisse PV antav väärtus määrab piiri mille ületamisel loenduri väljund lülitatakse sisse ja millest all pool välja. Loendusväärtust on võimalik lugeda siirde käsu ja kirjutada ta mälupiirkonda, näiteks *MOVW C1, VW100*.



Joonis 9.19. Loenduri tööd kirjeldav ajadiagram

Viivituskäskud

Juhtimisülesannete lahendamisel tuleb väga sageli ühe või teise seadme sisse-, välja- ja ümberlülitamiseks kasutada viivitusi. Kontrollerites on ajafunktsioonid ehk taimerid integreeritud protsessoriplokki. Kasutamisel antakse programmiga ette ajaintervallid, taimerid ja käivitustingimused. Programmeerimisel saab kasutada viit erinevat ajafunktsiooni: impulss-

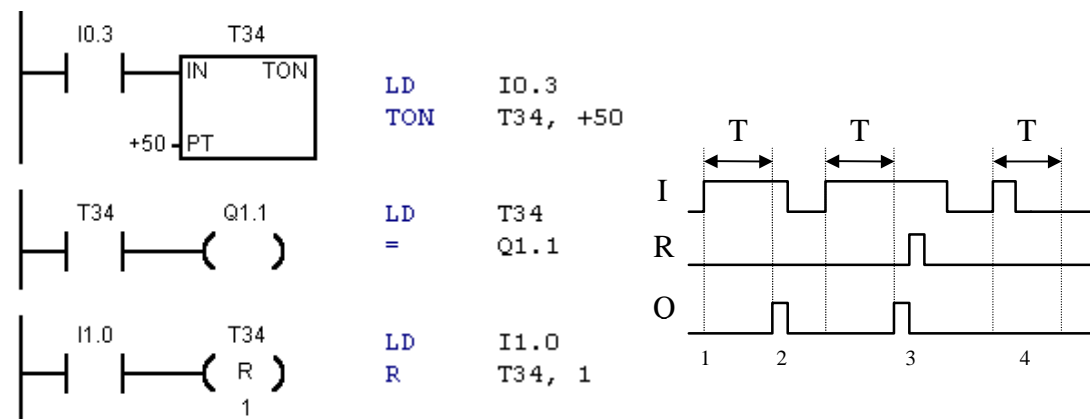
, pikendatud impulss-, viivitusega sisselülitus-, salvestavat viivitusega sisselülitus-, viivitusega väljalülitusfunktsiooni

Loetletud ajafunktsioonide realiseerimist STEP 7 keeles ja nende erisusi selgitab alljärgnev. S7-200 sarja kontrollerite puhul eristatakse 3 erineva resolutsiooniga taimereid. Sõltuvalt protsessijuhtimisele esitatud nõudmistestvajadusel saab kasutada 1, 10 ja 100 ms kaupa aja lugemist (tabel 9.1).

Tabel 9.1 Taimerid

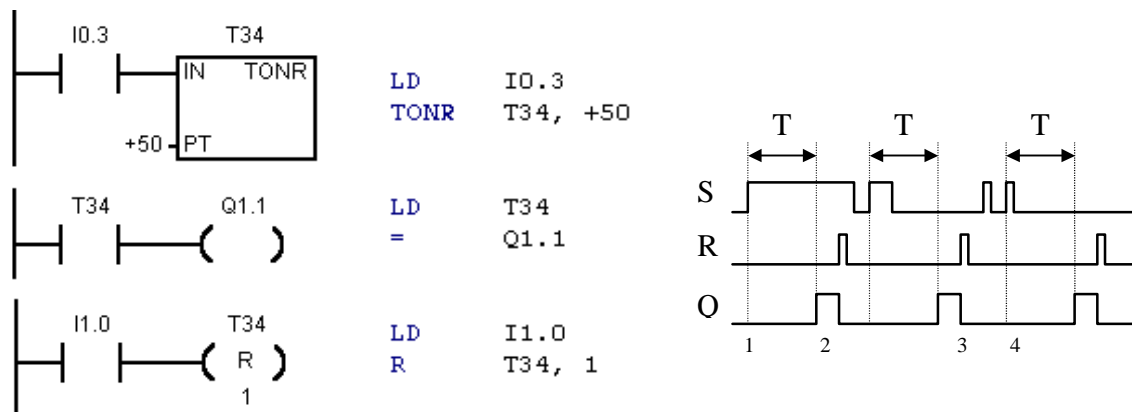
Resolutsioon, ms	Maksimaalne aeg, s	Taimeri number
1	32.767	T32, T96
10	327.67	T33-T36, T97-T100
100	3276.7	T37-T63, T101-T255

Viivitusega sisselülitusfunktsiooni (joonis 9.20) kasutamisel hakatakse ajaintervalli T loendama S-sisendimpulsi tõusva fronti korral (1). Väljund Q läheb ajaintervalli möödudes olekusse "1", kui sisendimpulsi kestus on etteantud ajaintervallist (2) suurem. Kui S-sisendsignaali olek "1" kestab etteantud ajast vähem, siis väljund Q ei lähe olekusse "1" (4). Väljund Q viiakse olekusse "0" S-sisendsignaali viimisega olekusse "0" pärast etteantud ajaintervalli möödumist (2) või R-sisendsignaali tõusva frondiga (3). Seda ajafunktsiooni tähistatakse rahvusvaheliselt sümboliga *SD*.



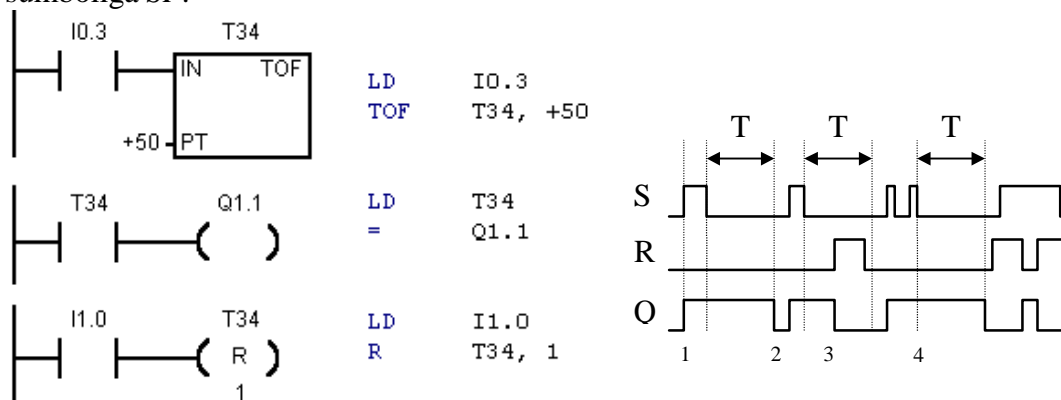
Joonis 9.20. Viivitusega sisselülitusfunktsioon

Salvestava viivitusega sisselülitusfunktsiooni (joonis 9.21) kasutamisel hakatakse ajaintervalli T loendama S-sisendimpulsi tõusva fronti korral (1) ja väljund Q läheb ajaintervalli möödudes olekusse "1" (2). Erinevus võrreldes eelmisega seisneb selles, et väljundit Q saab viia olekust "1" olekusse "0" ainult R-sisendimpulsi tõusva frondiga (3). Mitme jadamisi etteantud ajaintervalli jooksul S-sisendisse saabuva impulsi puhul hakatakse aega lugema viimase impulsi tõusvast frondist (4). Seda ajafunktsiooni tähistatakse rahvusvaheliselt sümboliga *SS*.



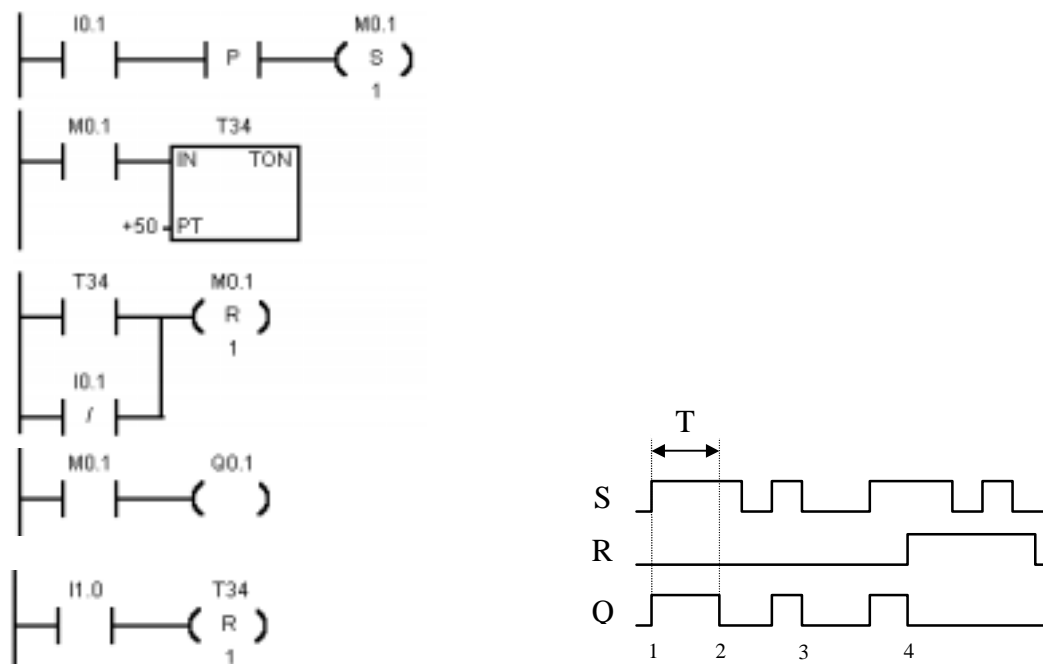
Joonis 9.21. Salvestav viivitusega sisselülitusfunktsioon

Viivitusega väljalülitusfunktsiooni (joonis 9.22) saab kasutada mingi seadme viivitusega väljalülitamiseks. S sisendimpulsi tõusva frondi korral läheb väljund Q koheselt olekusse “1” (1). Väljund Q läheb olekusse “0”, kui S-sisendsignaali langevast frondist on möödunud etteantud aeg (2) või kui R-sisendis on tuvastatud tõusva frondi (3) signaal. Ajaintervalli T jooksul mitme jadamisi S-sisendisse saanud impulsi korral alustatakse viivituse lugemist viimase impulsi langevast frondist (4). Seda ajafunktsiooni tähistatakse rahvusvaheliselt sümboliga *SF*.



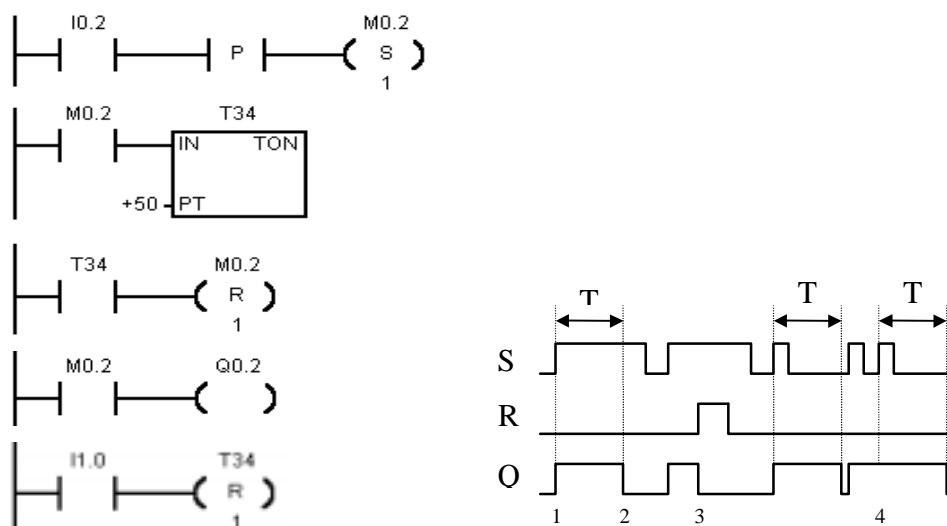
Joonis 9.22. Viivitusega väljalülitusfunktsioon

Impulssfunktsioon (joonis 9.23). Ajaintervalli T loendamine algab S sisendisse antava impulsi tõusva frondiga ja väljund Q läheb koheselt olekusse “1” (1). Kui sisendimpulss S on etteantud ajaintervallist pikem, loetakse ta lõpuni ja väljund Q läheb seejärel olekusse “0” (2). Kui sisendimpulss S on lühem (3) kui etteantud ajaintervall või sisendsignaali R (4) omab tõusvat fronti, katkestatakse ajaintervalli lugemine ja väljund läheb olekusse “0”. Seda ajafunktsiooni tähistatakse rahvusvaheliselt sümboliga *SP*.



Joonis 9.23. Impulssfunktsioon

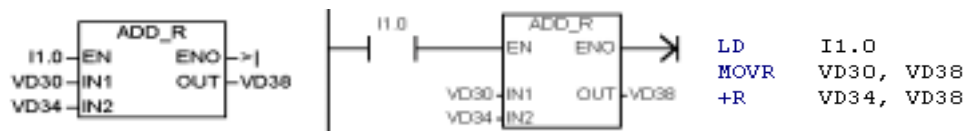
Pikendatud impulssfunktsiooni (joonis 9.24) korral hakatakse aega T lugema S-sisendisse antava impulsi tõusva frondi puhul ja väljund Q läheb olekusse “1” (1). Võrreldes lihtsa impulssfunktsiooniga ei sõltu ajaintervalli lugemine ja väljundsignaali oleku “1” kestus sisendisse S antava signaali kestusest, kuid sõltub R sisendisse antava signaali olekust. Väljundi Q signaal läheb olekusse “0”, kui etteantud ajaintervalli loendamine on lõppenud (2) või kui R-sisendis on tuvastatud tõusva frondiga (3) signaal. Mitme jadamisi etteantud ajaintervalli jooksul S-sisendisse saabunud impulsi korral hakatakse aega lugema viimase impulsi tõusvast frondist (4). Seda ajafunktsiooni tähistatakse rahvusvaheliselt sümboliga *SE*.



Joonis 9.24. Pikendatud impulssfunktsioon

Aritmeetikakäsud

Aritmeetikakäsud võimaldava matemaatilisi põhitehteid sõna- ja topeltsõnapikkuste täisarvude ning ujukomaarvudega. Näiteks liitmine $+I$ (*plus integer*), $+D$ (*plus double integer*), $+R$ (*plus real*), lahutamine $-I$, $-D$, $-R$, korrutamine ($*I$, $*D$, $*R$) ja jagamine ($/I$, $/D$, $/R$). Peale selle võimaldab STEP 7 tarkvara kasutada ka mõningaid trigonomeetria- ja erifunktsioone, nagu *SIN*, *COS*, *TAN*, *ASIN*, *ACOS*, *ATAN*, *EXP*, *LN*, *SQR* ja *SQRT*.



Joonis 9.25. Aritmeetikakäsk

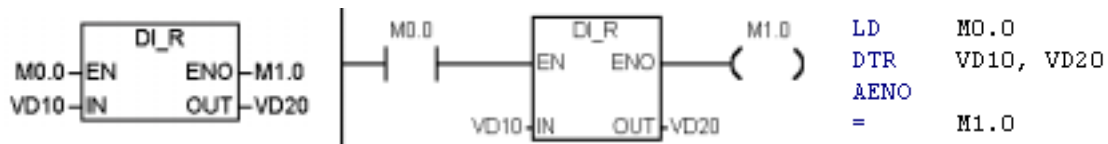
Joonisel 9.25 esitatud reaalarvude liitmiskäsu kirjeldus on alljärgnev (joonis 9.26). Kui sisend I1.0 (EN sisend) on sisselülitatud, siis siiratakse sisendis IN1 määratud aadressil VD30 olev väärtus aadressile VD38 (OUT väljund) ning seejärel liidetakse aadressil VD38 paiknevale väärtusele sisendis IN2 määratud aadressil VD 34 olev väärtus.

Liitmine	
VD30	+3.0E+00
	+
VD34	+2.0E+00
	=
VD38	+5.0E+00

Joonis 9.26. Aritmeetikakäsu tööpõhimõtte kirjeldus

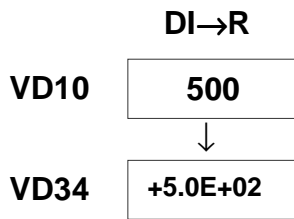
Teisenduskäsud

Teisenduskäsud võimaldavad eri pikkustega (sõna, topeltsõna) arvude teisendusi ühest arvusüsteemist või andmetüübist teise. Näiteks bait täisarvuks BTI (*byte to integer*), täisarv baidiks ITB (*integer to byte*), täisarv topeltsõna täisarvuks ITD (*integer to double integer*), topeltsõna täisarv täisarvuks DTI (*double integer to integer*), reaalarv täisarvuks ümardamisega ROUND(*real to double integer with rounding*), reaalarv märgiga täisarvuks TRUNC (*real to signed double integer*), täisarv ASCII koodi ITA(*integer to ASCII*), topeltsõna täisarv ASCII koodi DTA(*double integer to ASCII*), reaalarv ASCII koodi RTA(*real to ASCII*), ASCII kood kuuteistkümnendarvuks ATH(*ASCII to hexadecimal*), kuuteistkümnendarv ASCII koodi HTA(*hexadecimal to ASCII*) jne. Eelmainitud teisenduste programmiline esitus programmis sarnaneb järgneva näitega.



Joonis 9.27. Teisenduskäsk

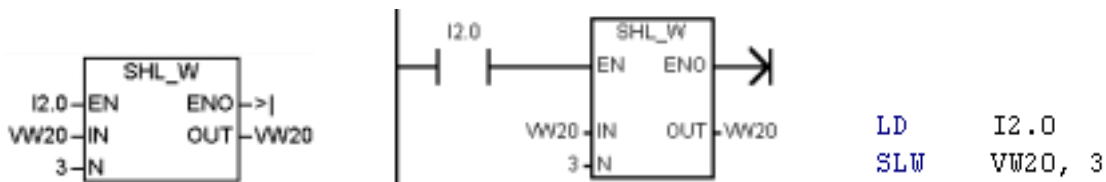
Joonisel 9.27 esitatud teisenduskäsk, mis teisendab topeltsõnapikkuse arvu reaalarvuks, on alljärgnev (joonis 9.28). Kui sisend M0.0 (EN sisend) on sisselülitatud, teisendatakse mälu aadressil VD10 paiknev topeltsõna pikkune täisarv reaalarvuks ja kirjutatakse mällu aadressile VD20. Vigadeta teisenduse korral viiakse väljund M1.0 (ENO väljund) olekusse “1”



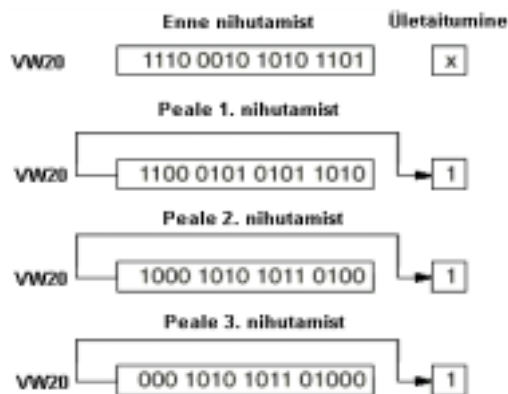
Joonis 9.28. Teisenduskäsu graafiline selgitus

Nihkekäsud

Nihkekäsud võimaldavad nihet paremale *SRB* (shift right byte), *SRW* (shift right word), *SRD* (shift right double word) ja vasakule *SLB*, *SLW* (joonis 9.29), *SLD* ning ringnihketehet paremale *RRB* (rotate right byte), *RRW* (rotate right word), *RRD* (rotate right double word) ja vasakule *RLB*, *RLW*, *RLD* vastavalt baitide, sõnade ja topeltsõnadega.



Joonis 9.29. Nihkekäsk

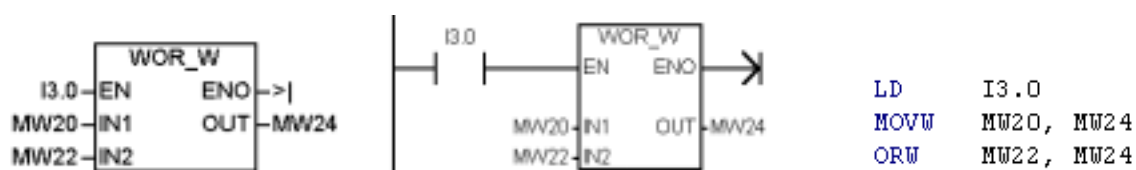


Joonis 9.30. Nihkekäsu graafiline selgitus

Kahendarvude nihutamine vasakule tähendab sisuliselt nihutatava arvu korrutamist 2^N -ga, kus N on nihutatavate kohtade arv. Kahendarvude nihutamine paremale tähendab sisuliselt nihutatava arvu jagamist 2^N -ga, kus N on nihutatavate kohtade arv (joonis 9.30).

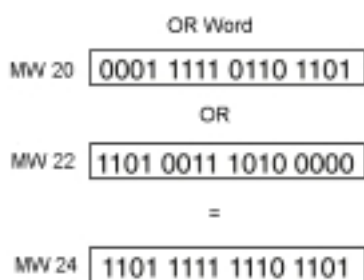
Loogikatehete käsud

Loogikatehete käsud võimaldavad loogikatehteid nagu VÕI, NING, EITUS ja VÄLISTAV-VÕI, kuid erinevus seisneb selles, et tehe teostatakse baitide, sõnade või topeltsõnade sama järku bittide vahel. VÕI-tehe *ORD* (or double word), *ORW* (or word) (joonis 9.31), *ORB* (or byte); NING-tehe *ANDD*, *ANDW*, *ANDB*; EITUS-tehe *INVD* (invert double word), *INW* (invert word), *INVB* (invert byte); VÄLISTAV-VÕI tehe *XORD* (exclusive or double word), *XORW* (exclusive or word), *XORB* (exclusive or byte).



Joonis 9.31. Loogikatehe

Joonisel 9.31 esitatud kahendarvude VÕI-tehte kirjeldus on alljärgnev (joonis 9.32). Kui sisend I3.0 (EN sisend) on sisselülitatud, siis siiratakse sisendis IN1 määratud aadressil MW20 olev kahendarvuline väärtus aadressile MW24 (OUT väljund) ning seejärel teostatakse aadressil MW22 (sisendis IN2) paikneva väärtuse ja aadressil MW24 paikneva väärtuse sama järku bittide vahel VÕI tehe ning tulemus salvestatakse aadressile MW24.



Joonis 9.32. Loogikatehte kirjeldus

Programmi juhtkäsud ja alaprogrammplokkide koostamine

Programmeerimiskeele STEP 7 struktuurikäskude all mõistetakse plokisisesid siirdekäske ja plokkidevahelisi siirdekäske ehk plokkide päringukäske (tabel 9.2).

Keeruka protsessi juhtimisülesannet on lihtsam programmeerida, kui see jaotada osaprotsessideks ehk *plokkideks*. Alati on vaja lahendada küsimus, kuidas määrata osaprotsessid, et need moodustaksid terviku. Võtkem näiteks valgusfooride juhtimine. Nagu

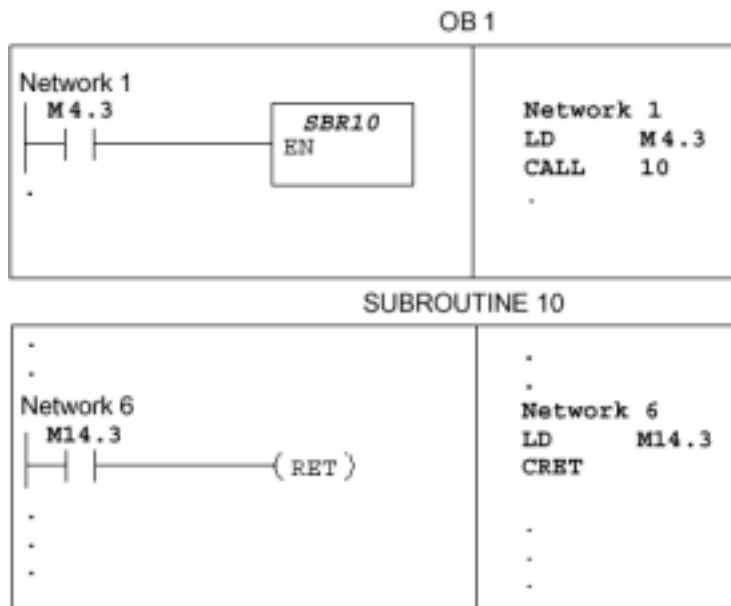
teada, on valgusfooril 3 tuld (punane, kollane, roheline). Teada on ka see, et valgusfoori tuled ei sütti korraga, s. t. iga tule süttimiseks on vastav tingimus. Seega koosneb valgusfoori juhtimine mitmest väiksemast alamprotsessist. Iga alamprotsessi käivitab omakorda mingi tingimus. Sellist programmeerimist, kus alamprotsesside kirjeldamiseks kasutatakse eraldi plokkide, nimetatakse *jaotatud* programmeerimiseks.

STEP7 keeles kasutatakse ülemisel tasandil juhtplokkina *OB*-tüüpi plokkide, s. t. esimesena pöördub süsteem nende poole ja alles seejärel vastavalt teiste programmis (juhtplokkis) määratud plokkide poole. Üks tähtsamaid juhtplokkide on *OB1*, mis on mõeldud tsükliliseks programmijuhtimiseks. Operatsioonisüsteem pöördub *OB1* poole tsükliliselt, iga teatud ajavahemiku tagant. Plokkidevahelisteks siireteks ja neis sisalduvate programmide täitmiseks on plokkidevahelised siirdekäskud ja plokisisesed siirdekäskud. Plokkidevaheliste siirete teostamiseks kirjutatakse käsu järel plokki nimi kuhu tuleb siirduda. Plokisisesete siirete teostamiseks kirjutatakse käsu järgi vastav märgendi nimetus, mis määrab ära millisele märgendiga tähistatud käsureale siire toimub.

Tabel 9.2 Siirdekäskud

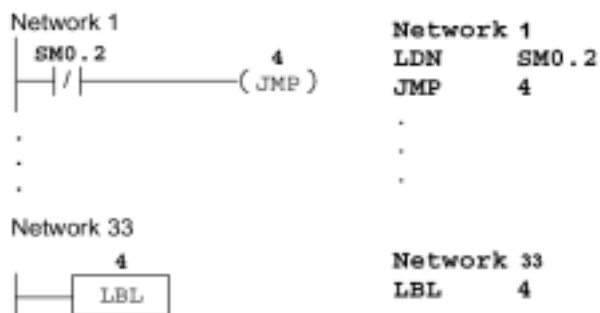
Siirdekäsk	Kirjeldus
<i>CALL</i> alamprogrammi number	Selle käsu tulemusel toimub siire alamprogrammi. Siire võib olla ka tingimuslik
<i>CRET</i> , <i>RET</i>	Tingimuslik ja tingimusetu siire tagasi juhtprogrammi ehk <i>OB1</i> -te
<i>JMP</i> märgendi tähis, <i>LBL</i> märgendi tähis	<i>JMP</i> käsu tulemusel toimub plokki sisene siire märgendi käsuga <i>LBL</i> tähistatud käsureale
<i>FOR</i> indeksmälu, algus, lõpp <i>NEXT</i>	Nende käskude abil teostatakse mingi programmi lõigu tsüklilist täitmist ehk selles lõigus tsüklilist siirdumist. Indeksmälu on mõeldud tsüklite loendamistoimingute jaoks ning algus ja lõpp määravad teostatavate tsüklite arvu. Käskude <i>FOR</i> ja <i>NEXT</i> vahelist osa seega täidetakse nii mitu korda kui on parameetrite lõpp ja algus vahe pluss üks.
<i>STOP</i>	<i>STOP</i> käsu abil katkestatakse programmi töö ja kontrolleri viiakse talitusviisi <i>STOP</i> .
<i>END</i>	Tingimuslik siire programmiploki lõppu

Juuresoleval joonisel 9.33 toodud programmi kirjeldus on alljärgnev. Kui mälobitt *M4.3* omab olekut "1", siis siirduakse juhtprogrammi *OB1* programmisegmentist 1 alamprogrammi *SBR10*, kasutades käsku *CALL*. Kui mälobitt *M14.3* omab olekut "1", siis siirduakse alamprogrammi *SBR10* programmisegmentist juhtprogrammi *OB1*, kasutades käsku *CRET*



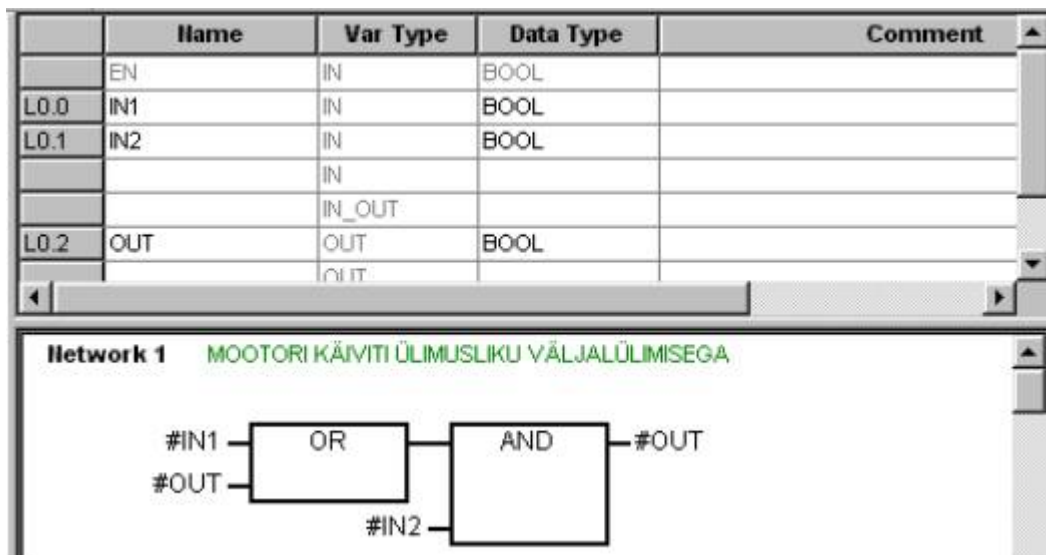
Joonis 9.33. Plokkide vaheline siire

Juuresoleval joonisel 9.34 toodud programmi kirjeldus on alljärgnev. Kui süsteemimälu bitt SM 0.2 omab olekut “0”, siis toimub samas programmiplokis siirdumine märgendiga 4 tähistatud käsureale.



Joonis 9.34. Plokisisene siire

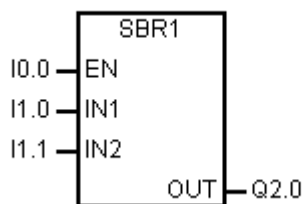
Juuresoleval joonisel 9.35 toodud programmi kirjeldus on alljärgnev. Kui sisend I2.1 on sees ehk omab oleku “1”, siis täidetakse käskude FOR ja NEXT vahelist osa ehk segmenti 10 ja 15 vahele jäävat programmi lõiku kaks korda vastavalt sisenditesse INITIAL ja FINAL salvestatud väärtustele. Jooksva tsükli number salvestatakse iga tsükli käigus mällu aadressile VW225.



Joonis 9.37. Muutujatega alamprogrammplokk

Kui muutujatega alamprogrammplokk on koostatud, võime seda korduvalt kasutada juhtprogrammplokis OB1 ja omistame muutujatele vastavad operandid (joonis 4.38).

Network 1 MUUTUJATEGA ALAPROGRAMMI ESITUS OB1-s



Joonis 9.38 Muutujatega programmploki kasutamine juhtplokis OB1