

PROGRAMIRANJE INDUSTRIJSKIH KONTROLERA

- materijal za pripremu ispita -
(smer: EKIS, VII semestar)

SADRŽAJ

1	UVOD.....	4
1.1	Istorijat PLC-ova.....	4
1.2	Prednosti i nedostaci PLC-ova.....	6
1.2.1	Prednosti	6
1.2.2	Nedostaci.....	8
1.3	PLC sistem.....	9
1.3.1	Konstrukcija PLC-a	9
1.3.2	CPU i PM.....	10
1.3.3	PLC ulazni i izlazni moduli	12
1.4	Uređaji sa kojima se PLC povezuje.....	12
1.4.1	Ulazni ON/OFF uređaji.....	12
1.4.2	Ulazni analogni uređaji	13
1.4.3	Izlazni ON/OFF uređaji	15
1.4.4	Izlazni analogni uređaji.....	16
2	CPU jedinica PLC kontrolera	17
2.1	Sken ciklus.....	17
2.2	Osnovne karakteristike CPU jedinice	18
2.2.1	CPU jedinica kontrolera SLC 500	19
2.3	Organizacija memorije.....	19
2.3.1	Programske datoteke.....	20
2.3.2	Datoteke podataka.....	20
3	Diskretni U/I moduli	25
3.1	Diskretni ulazni moduli.....	26
3.1.1	Povezivanje digitalnih senzora	28
3.2	Diskretni izlazni moduli.....	29
3.3	SLC 500 diskretni U/I moduli.....	31
3.4	Sprezanje diskretnih U/I modula sa kontrolerom	32
4	Leder programiranje.....	34
4.1	Rang	34
4.2	Bit naredbe.....	35
4.2.1	Bit naredbe za definisanje uslova	35
4.2.2	Bit naredbe za postavljanje vrednosti izlaza.....	36
4.2.3	Bit trigger naredba	36
4.3	Kontakti.....	37
4.4	Start/stop kolo	40
4.5	Primeri.....	42
4.6	Metod konačnih automata.....	59
4.6.1	Realizacija konačnog automata u leder jeziku.....	62
4.6.2	ASM dijagrami.....	74
5	Naredbe za merenje vremena i prebrojavanje događaja – tajmeri i brojači.....	78
5.1	Naredbe za merenje vremena.....	78
5.1.1	Parametri tajmera	79
5.1.2	Datoteka podataka o tajmerima (timer data file)	79
5.1.3	Naredbe časovnika	80
5.2	Naredbe za prebrojavanje događaja	84
5.2.1	Parametri brojača	84
5.2.2	Datoteka podataka o brojaču (counter data file).....	84
5.2.3	Naredbe brojača	85
5.3	Primeri korišćenja tajmera i brojača	86

6	Naredbe za operacije nad podacima	104
6.1	Naredbe za poredenje.....	104
6.2	Matematičke naredbe.....	106
6.2.1	Aritmetičke i logičke binarne operacije.....	106
6.2.2	Unarne operacije.....	106
6.2.3	Složene matematičke naredbe.....	107
6.2.4	Naredbe za manipulaciju sa numeričkim podacima	108
6.3	Primeri.....	109
7	Analogni PLC moduli.....	112
7.1	Analogni ulazni moduli.....	112
7.2	Analogni izlazni moduli.....	112
7.3	SLC 500 analogni U/I moduli.....	112
7.4	Sprezanje analognih modula sa kontrolerom.....	113
7.5	Primeri.....	114
8	Sekvencijalno upravljanje.....	131
8.1	Naredbe za sekvencijalni rad sa datotekama	132
8.1.1	Korišćenje SQL naredbe.....	135
8.1.2	Korišćenje SQO naredbe.....	136
8.1.3	Korišćenje SQC naredbe.....	136
8.2	Primeri.....	137
9	PLC moduli specijalne namene	141
9.1	Izdvojeni ulazno-izlazni moduli	141
9.2	Komunikacioni moduli	141
9.3	PLC moduli za kontrolu pozicije.....	142
9.4	Moduli za vizuelnu kontrolu.....	143
9.5	Bar-kod moduli	144
9.6	Moduli za PID upravljanje.....	144
9.7	Radio-frekvencijski moduli	147
9.8	Operatorski terminali	148

1 UVOD

Industrijski kontroler ili **PLC** od *Programmable Logic Controller* - programabilni logički kontroler.

PLC - digitalni elektronski uređaj koji poseduje programabilnu memoriju za smeštanje instrukcija kojima se realizuju specifične funkcije, kao što su logičke operacije i sekvenciranje, tajming, brojanje i aritmetičke operacije, a u cilju upravljanja različitim mašinama i procesima putem digitalnih ili analognih ulazno/izlaznih modula.

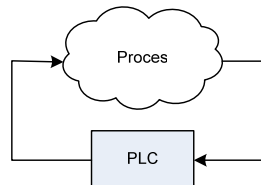
PLC - namenski elektronski **uređaj**, zasnovan na mikroprocesoru, koji je u mogućnosti da obavlja brojne tipove upravljačkih funkcija različitog nivoa složenosti.

PLC - **industrijski računar** čiji su hardver i softver posebno prilagođeni radu u industrijskim uslovima, a koji se može lako programirati i ugrađivati u nove i postojeće industrijske sisteme.

PLC - (*Programmable*) - označava mogućnost programiranja. Program rada se unapred priprema i puni u permanentu memoriju PLC kontrolera. PLC program se razvija u jeziku lestvičastih (**ladder**) dijagrama, koji je nastao po ugledu na tzv. relejne šeme.

PLC - (*Logic*) - jedna od glavnih osobina PLC kontrolera je mogućnost obavljanja logičkih (bulovih) funkcija. PLC generiše diskretne (digitalne) izlazne signale u funkciji (logičkoj) diskretnih ulaznih signala - karakteristično za prvobitne tipove PLC kontrolera. Savremeni PLC-ovi, pored logičkih mogu dodatno da obavljaju aritmetičke operacije, odmeravaju vremenske intervale, prebrojavaju događaje, a prihvataju i generišu, pored diskretnih, i analogne signale.

PLC (*Controller*)- Glavna primena PLC-ova je u industriji (proizvodnoj) gde se koriste za automatsko upravljanje procesima - prati ključne parametre procesa (posredstvom priključenih senzora i davača, i shodno memorisanom programu, generiše pobudu kojom deluje na proces (posredstvom aktuatora) - Sl. 1-1.



Sl. 1-1 PLC - upravljačka jedinica procesa.

PLC vs. PC. PLC kontroler se razlikuje od računarskog sistema opšte namene po tome što nema spoljnu memoriju (diskove), kao i niz standardne ulazno/izlazne opreme. Pored toga, njegov operativni sistem je jednostavniji i pruža komparativno manje mogućnosti od računara opšte namene. Zapravo, PLC je koncipiran i projektovan za jedan relativno uzak i jasno definisan obim poslova vezanih za nadzor i upravljanje pojedinim uređajima, mašinama i procesima, što je rezultovalo u njegovoj izuzetnoj efikasnosti i jednostavnosti. U izvesnom smislu, područje primene PLC kontrolera isto je kao i za specijalizovane mikroručunarske kontrolere ili signal procesore. Ključna razlika leži u činjenici da korišćenje PLC-a ne zahteva od korisnika gotovo nikakvo predznanje o arhitekturi mikroručunarskih sistema i programiranju. Drugim rečima, korisnik PLC kontrolera je u najvećoj mogućoj meri oslobođen rešavanja različitih problema vezanih za čisto računarski aspekt, i može da se u punoj meri skoncentriše na projektovanje same aplikacije.

1.1 Istorijat PLC-ova

Nastanak PLC-ova se vezuje za kasne 60' i rane 70' godine prošlog veka, a nastali su na bazi konvencionalnih računara tog vremena. Njihova prvobitna primena bila je u automobilske industriji, a sa ciljem da se skрати vreme zastoja u proizvodnji usled promene proizvodnog procesa. Priprema pogona za proizvodnju novog modela automobila trajala je mesecima, a podrazumevala je

prepovezivanje panela i ormara prepunih žica, relea, tajmera, sklopki i drugih, uglavnom elektromehaničkih komponenti, pomoću kojih su se, u to vreme, realizovale upravljačke jedinice. Uvođenjem PLC-a, omogućilo je da se reprogramiranje obavi "preko tastature" i da se, uz minimalna dodatna prepovezivanja, vreme zastoja proizvodnje skрати do tek nekoliko dana.

Glavni problem sa računarima/PLC-ovima iz 70' godina odnosio se na njihovo reprogramiranje. Programi su bili pisani na niskom nivou (assembler), a time i komplikovani, a njih su mogli da sastavljaju samo visoko-stručni i iskusni programeri. Kasnih 70' godina učinjen je izvestan napredak u pogledu pojednostavljenja procedure programiranja. 1972 godine pojavlju se mikroprocesori, koji, zahvaljujući performansama koje su nudili i relativno niskoj ceni, brzo nalaze široku primenu u mnogim oblastima elektronike, pa i industrijske automatizacije. Od tog vremena, PLC-ovi se neprestano usavršavaju, kako u pogledu performansi, moći obrade, mogućnosti sprežavanja sa najrazličitijim uređajima, tako u pogledu načina i procedura programiranja, kao i jezika koji se koriste za njihovo programiranje. PLC su postali "razumljiviji" široj populaciji industrijskih inženjera.

80' dolazi do eksponencijalnog rasta tržišta PLC-ova, a primena PLC-a se širi van industrijskih pogona, na sisteme kao što su elektro-energetska postrojenja, sistemi automatizacije zgrada, sistemi obezbeđenja i nadzora i td. Danas, PLC-ovi postaju prisutni i u oblastima kao što je medicinska oprema i uređaji, kućni aparati i sl.

Prvi PLC kontroleri su bili jednostavni uređaji za on/off upravljanje i koristili su se za zamenu zastarele relejne tehnike. Međutim, takvi PLC kontroleri nisu mogli da obezbede složenije upravljanje, kao što je upravljanje temperaturom, pritiskom, pozicijom. U međuvremenu, proizvođači PLC kontrolera razvili su i ugradili u PLC kontrolere brojna poboljšanja i funkcionalna unapređenja. Savremeni PLC kontroleri imaju mogućnost obavljanje izuzetno složenih zadataka kako što je upravljanje preciznim pozicioniranjem i upravljanje složenim tehnološkim procesima. Takođe, brzina rada PLC kontrolera je značajno povećana, kao i lakoća programiranja. Razvijeni su brojni moduli specijalne namene za primene kao što je radio komunikacija, vizija ili čak prepoznavanje govornih komandi.

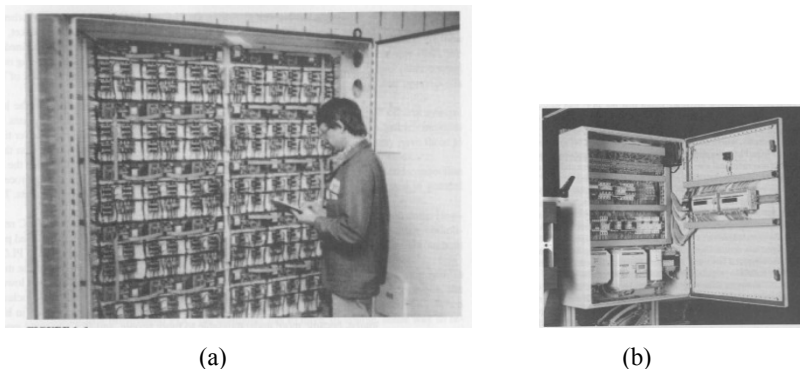
Smatra se da inženjer sa znanjem relejne logike može ovladati osnovnim PLC funkcijama u roku od nekoliko sati. Slično važi i za inženjere sa znanjem digitalne elektronike. Za ovladavanje složenijim funkcijama i mogućnostima PLC-a dovoljno je nekoliko nedelja ili jedan kurs.

Danas postoji veliki broj različitih tipova PLC kontrolera koji se razlikuju po veličini, izgledu i moći obrade, počev od malih jedinica sa malim i ograničenim brojem ulaza i izlaza do velikih, modularnih sistema koje se mogu konfigurisati za rad sa više stotina ili čak hiljada ulaza/izlaza. Na Sl. 1-2 je prikazan izgled PLC kontrolera iz familije **Allen Bradley SLC 500 Modular Controllers**. PLC se sastoji iz šasije (*rack*) koja ima određeni broj *slotova* u koje se stavljaju pojedini *moduli*. Prvi dva slotova u šasiji zauzimaju uređaj za napajanje i procesorski modul, dok je raspored modula u preostalim slotovima proizvoljan. U zavisnosti od broja modula, PLC može imati i više od jedne šasije. Svaka šasija ima sopstveno napajanje, dok se procesorski modul nalazi samo u prvoj šasiji. Programabilni logički kontroleri iz familije **Allen Bradley SLC 500 Modular Controllers** mogu imati najviše **tri** šasije sa najviše **30 slotova**. Pri tome, postoje šasije sa 4, 7, 10 i 13 slotova.



Sl. 1-2 Izgled komponenti modularnog PLC sistema.

Na Sl. 1-3(a) je prikazan kontrolni panel starijeg datuma koji se koristi u nekom proizvodnom pogonu za upravljanje procesom proizvodnje. Panel je velikih dimenzija i sadrži veliki broj žica, konektora, relae, što može stvoriti velike probleme u održavanju i servisiranju. Na Sl. 1-3(b) je prikazan PLC sistem koji u potpunosti zamenjuje relejni sistem sa Sl. 1-3(a). PLC sistem je manjih dimenzija i pouzdaniji. U slučaju promene načina rada, relejni sistem se mora prepovezati, dok se PLC može veoma brzo reprogramirati.



Sl. 1-3 (a) relejni panel; (b) PLC sistem

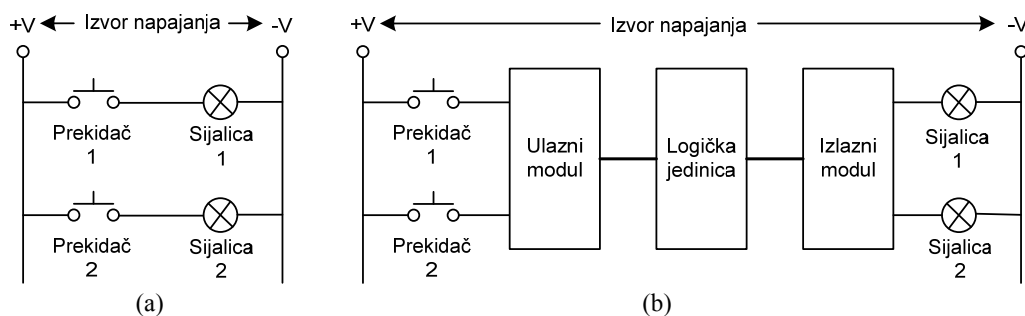
1.2 Prednosti i nedostaci PLC-ova

1.2.1 Prednosti

Fleksibilnost. U prošlosti, svaka električno-upravljana mašina za proizvodnju zahtevala je svoju sopstvenu upravljačku jedinicu; u pogonu sa 15 mašina, postojalo je 15 različitih, namenski projektovanih, upravljačkih jedinica. Danas je moguće isti model PLC-a koristiti za upravljanje bilo kojom od 15 mašina. Uz to, verovatno neće biti potrebe za 15 PLC-ova, jer jedan PLC lako može da opsluži više različitih mašina, tako što će, konkurentno, za svaku priključenu mašinu izvršavati poseban, namenski program.

Laka promena programa i korekcija grešaka. Kod tradicionalnih, relejnih panela, svaka promena programa zahtevala je značajno vreme za prepovezivanje panela i uređaja. Sa druge strane, kod PLC-a, promena programa je laka i brza. Novi program se preko tastature ili na neki drugi način učitava u PLC, a prepovezivanje obično nije potrebno, tako da celokupna aktivnost ne traje duže od nekoliko minuta. Takođe, uočene nepravilnosti u radu sistema, koje su posledica greške u programu se mogu lako i brzo ispraviti.

Relejni vs. PLC sistem. Pre pojave PLC kontrolera, za upravljanje mašinama i industrijskim postrojenjima korišćena su mehanički uređaji, kao što su prekidači i relei povezani žicama. Na Sl. 1-4(a) prikazano je jedno veoma jednostavno žičano (relejno) kolo. Prikazano kolo realizuje trivijalnu logiku: kada je prekidač zatvoren, sijalica svetli, a kada je prekidač otvoren, sijalica ne svetli. Dva para prekidač-sijalica su nezavisni zato što su vezani paralelno i tako ne utiču jedan na drugog. Ukoliko sada želimo da promenimo logiku rada kola, tako da, na primer, uključenje sijalice 2 isključi sijalicu 1, moramo, najpre, nacrtati novu šemu veze, a zatim, raskinuti postojeće veze između prekidača i sijalica i po novoj šemi iznova povezati prekidače i sijalice. Ako se nakon toga javi potreba da se logika rada ponovo promeni, tako da se, na primer, sijalica 1 uključuje samo ako su oba prekidača zatvorena, dok se sijalica 2 uključuje bilo kojim od dva prekidača, kolo se mora još jednom rastaviti i ponovo povezati. Industrijske mašine i postrojenja su mnogo složeniji od ovog jednostavnog primera. Osim, obično, velikog broja priključenih ulaznih i izlaznih uređaja, upravljanje ovakvim sistemima treba da obezbedi obavljanje složenih operacija u tačno određenom redosledu, kako i više različitih režima rada (manuelni, korak-po-korak, automatski) i slično. Takav jedan tipičan upravljački sistem bi zahtevao na stotine releja, povezanih kilometrima žice smeštenih u glomaznim ormarima. Jasno je da pod ovakvim uslovima svaka promena upravljačke logike predstavlja dugotrajnu, mukotrpnju i skupu aktivnost.



Sl. 1-4 Relejni sistem v.s. PLC sistem. (a) relejni sistem; (b) PLC sistem

Pojavom PLC kontrolera postalo je moguće zameniti celokupnu upravljačku logiku, sličnosti i više stotina relea, jednim kompaktnim elektronskim uređajem. Na Sl. 1-4(b) je prikazano kako bi se PLC kontroler iskoristio u prethodno razmatranom primeru. Može se uočiti nekoliko razlika u odnosu na prvobitno rešenje. Prvo, prekidači nisu direktno povezani sa sijalicama. Umesto toga, prekidači su povezani na ulazni, a sijalice na izlazni modul PLC kontrolera. Drugo, ulazni i izlazni moduli nisu direktno povezani, već posredno preko logičke jedinice koja shodno programu upravljanja uključuje/isključuje sijalice u zavisnosti od tekućeg stanja prekidača. Na primer, da bi se realizovala funkcija sa Sl. 1-4(a), logička jedinica treba biti tako programirana da generiše pobudu za sijalicu 1 za sve vreme dok je prekidač 1 zatvoren, i u isto vreme, generiše pobudu za sijalicu 2 za sve vreme dok je prekidač 2 zatvoren. Program po kome logička jedinica radi se unosi u memoriju logičke jedinice preko tastature ili na neki drugi način, gde ostaje zapamćen sve dok se ne unese novi program. Očigledno, promena logike rada sistema, ne zahteva nikakvo ponovno povezivanje i ožičavanje.

Veliki broj kontakta. Funkcija relejne upravljačke jedinice se realizuje povezivanjem relea (kontakta) koji imaju ulogu elektro-mehaničkih logičkih kola ili flip-flopova. Broj relea u jednom relejnom panelu je ograničen (npr. fizičkim dimenzijama panela), što ograničava i složenost funkcije koja se može realizovati. Može se desiti da u slučaju promene "programa" rada upravljačke jedinice, broj raspoloživih relea nije dovoljan da bi se realizovala nova funkcija. U takvim slučajevima, neophodno je ugraditi dodatne relee ili relejne blokove, što može biti dugotrajna operacija skopčana s neizbežnim mehaničkim intervencijama. Sa druge strane, PLC može realizovati funkcije čija je složenost ograničena jedino raspoloživom memorijom - može se zamisliti da svaki bit interne memorije PLC-a predstavlja jedan kontakt (rele); za realizaciju jednog tajmera ili brojača koriste se tri 16-bitne reči memorije i td. Tipična memorija PLC je veličine više KB.

Niska cena. Sa napretkom tehnologije, implementacione mogućnosti PLC-ova neprestano rastu. Danas je moguće, po ceni ispod 100\$, nabaviti PLC sa ogromnim brojem interno-raspoloživih "virtuelnih" relea, tajmera, brojača, sekvencera i drugih funkcija (od kojih neke nemaju pandam u "relejnog" svetu).

Mogućnost probnog rada. Rad PLC-a se može ispitati u laboratoriji, pre ugradnje u proizvodni pogon. Program se piše, testira, analizira i, ako je neophodno, modifikuje sve do trenutka kada se proceni da su sve zahtevane funkcije korektno realizovane. Tek tada se program prenosi u PLC koji se instalira (ili je već instaliran) u proizvodni pogon. Na ovaj način se čini velika ušteda skupog "fabričkog" vremena (nema zastoja u proizvodnji). Nasuprot tome, testiranje konvencionalnih relejnih sistema se može obaviti samo u fabričkoj hali, što može biti veoma vremenski neracionalno.

Mogućnost vizuelnog praćenja rada. Rad PLC-a se može direktno pratiti na ekranu monitora - na pogodan način se u grafičkom obliku prikazuju stanja ulaza i izlaza PLC-a uz "osvetljeno" prikazivanje logičkih putanja koje su trenutno aktivne i ispisivanje obavешtenja o eventualnom neispravnom radu sistema ili o nastanku nekih izuzetnih situacija. Na taj način, svaka kritična situacija se može uočiti istog trenutka kada se i desila, što olakšava pronalazene nastalih kvarova. Naime, kod naprednih PLC sistema, postoji mogućnost programiranja poruka za svako moguće neispravno ponašanje, koje će se prikazati onog trenutka kada se takva situacija detektuje od strane PLC-a, (kao npr. "MOTOR #7 je preopterećen").

Brzina rada. U poređenju sa releima koji su, s obzirom na svoju elektro-mehaničku prirodu, spori, brzina izvršenja PLC programa je veoma velika. Brzina rada PLC-a određena je periodom tzv. *sken*

ciklusa, a ona je reda milisekundi (više o sken ciklusu, kasnije). Drugim rečima, vreme koje protekne od trenutka kada se promeni stanje ulaza PLC-a do trenutka kada PLC-a reaguje postavljajući svoje izlaze tipično nije duže od nekoliko do maksimalno nekoliko desetina milisekundi.

Leder programiranje. Za programiranje PLC kontrolera koristi se jezik *lestvičastih logičkih dijagrama* (ili leder dijagrama - *ladder diagram*), koji je već dugi niz godina u upotrebi u industriji pri projektovanju logičkih i sekvencijalnih relejnih uređaja. Ovaj jezik koristi grafičku notaciju koja je po vizuelnom izgledu i logici rada slična dijagramima relejnih šema (kao što je šema sa Sl. 1-4(a)) i zbog toga je lako razumljiv industrijskim inženjerima. Drugim rečima, industrijski inženjeri ne moraju biti eksperti za programiranje da bi u svojim sistemima koristili PLC-ove.

Pouzdanost i lakoća održavanja. Poluprovodničke komponente, od kojih je PLC sačinjen, su, generalno, pouzdanije od mehaničkih sistema ili relea i tajmera. Uz to, za realizaciju PLC-ovi se koriste komponente sa veoma visokim faktorom pouzdanosti. Iz tog razloga, troškovi održavanja upravljačkih sistema zasnovanih na PLC-u su niži, a vreme zastoja kraće.

Jednostavnost naručivanja komponenti upravljačkog sistema. PLC je jedan uradaj. Kada naručeni PLC stigne u industrijski pogon, svi brojači, relei, i druge "virtuelne" komponente "sadržane" u PLC-u su takođe stigle. Situacija je potpuno drugačija kada se projektuje relejni panel. Na primer, potrebno je 20 različitih tipova relea i tajmera koji se nabavljaju od 12 različitih dobavljača, sa različitim rokovima isporuke i potencijalnim problemima sa dostupnošću traženih komponenti. Ako projektant propusti da naruči samo jednu komponentu, celokupan projekat se odlaže za vreme isporuke te komponente. Kod PLC-ova uvek postoji "jedan rele više" - pod uslovom da je naručen PLC sa izvesnom rezervom u moći izračunavanja.

Dokumentacija. Leder dijagrami, kako grafički prikazi, su u toj meri samo-deskriptivni da obično nije neophodna neka dodatna dokumentacija koja bi upotpunjavala opis rada PLC-a i načina na koji su realizovane njegove funkcije. Leder dijagram se uvek može odštampati, a pošto se isti dijagram koristi i kao program, ne postoji opasnost da dokumentacija bude neažurna, što je često slučaj sa dijagramima i šemama relejnih panela (kada inženjer nakon učinjene intervencije ne unese izmenu u relejni šemu).

Bezbednost. Program PLC-a se ne može promeniti pre nego što je PLC "otključan" (svaki PLC ima prekidač s ključem - bez ključa nije moguće menjati program).

Mogućnost reprogramiranja. Osobina brzog reprogramiranja PLC-a otvara mogućnost za postizanje neke vrste adaptivnog proizvodnog procesa, gde se program rada menja shodno karakteristikama svakog pojedinačnog prizvoda ili čestim varijacijama u procesu proizvodnje.

1.2.2 Nedostaci

Nova tehnologija. Pokazalo se da je teško promeniti način razmišljanja industrijskih inženjera sa relejne logike na PLC koncept. Međutim, zahvaljujući sve široj primeni, ne samo u domovima i kancelarijama već sve više i u fabričkim halama, računari postaju i sredstvo za postizanje veće produktivnosti proizvodnje.

Aplikacije sa fiksim programom. Pojedine aplikacije zasnovane su na samo jednoj funkciji koja se veoma retko ili nikada ne menja. U ovakvim slučajevima, zamena postojeće opreme PLC-om ne donosi veliki dobitak, jer se njihova glavna osobenost – mogućnost reprogramiranja – praktično ne koristi. PLC je najbolje rešenje kada su neophodne periodične promene u načinu rada.

Uslovi rada. Ekstremnim uslovi rada, kao što su: visoka temperatura, vlažnost, vibracije, električne smetnje, a koji su karakteristični za pojedine proizvodne procese, mogu uticati na rad PLC-a i ograničiti njegovu primenu i/ili životni vek.

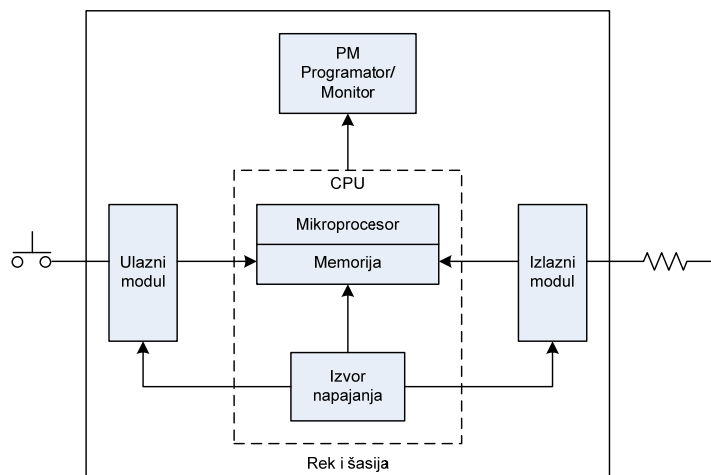
Bezbednost u radu. Kod relejnih sistema uvek postoji tzv. Stop prekidač, kojim se u bilo kom momentu može trenutno prekinuti rad sistema (isključenjem napajanja). Pri tome, relejni sistem se automatski ne resetuje kada se napajanje uključi, već zadržava stanje u kome je bio kada je napajanje isključeno. Ovakvo ponašanje se svakako može programski ostvariti i kod PLC-a, tako što će se Stop prekidač povezati na jedan od ulaza PLC-a. Međutim, ovakvo rešenje nije bezbedno (ako PLC otkaže,

prekidač Stop gubi funkciju!) Ovaj nedostatak se može prevazići ugradnjom bezbedonosnih relea na izlazima PLC-a (više detalja kasnije).

1.3 PLC sistem

Na Sl. 1-5 su, u obliku blok dijagrama, prikazane četiri osnovne jedinice svakog PLC sistema kao i načina na koji su međusobno povezane:

1. **Centralna procesorska jedinica (CPU) ili logička jedinica.** Predstavlja "mozak" sistema, a sastoji se iz sledeće tri podjedinice:
 - a. *Mikroprocesor.*
 - b. *Memorija* – za čuvanje sistemskog softvera i korisničkog programa
 - c. *Izvor napajanja* - obezbeđuje napajanje mikroprocesora, memorije, ulaznog i izlaznog modula.
2. **Programator/Monitor (PM).** PM je uređaj koji se koristi za komunikaciju sa PLC-om. Primeri PM-ova su: ručni terminali, industrijski terminali i personalni računari. Sa jedne strane, prema operateru, ovi uređaji poseduju ekran i tastaturu, dok sa druge, prema PLC-u, odgovarajući komunikacioni interfejs za prenos programa, podataka, statusnih informacija ka/iz PLC-a.
3. **U/I moduli.** Ulazni modul poseduje terminale (priključne tačke) na koje se dovode električni signali koje generišu senzori ili prevarači. Izlazni modul poseduje terminale preko koji PLC šalje izlazne signale kojima pobuđuje rele, solenoide, motore, displeje i druge izlazne uređaje ili aktuatorne.
4. **Rekovi i šasije.** Delovi PLC sistema CPU, PM i U/I moduli smeštaju se u metalne ormare – tzv. rekove.



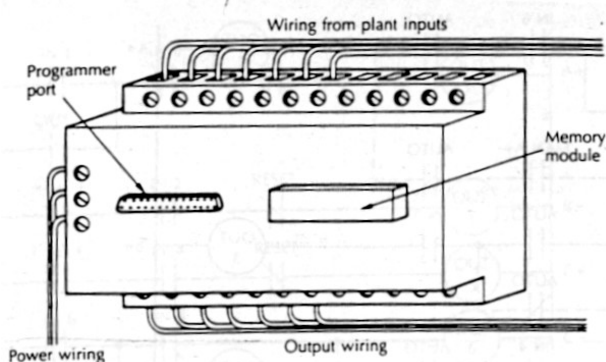
Sl. 1-5 PLC sistem.

1.3.1 Konstrukcija PLC-a

Razlikuju se dva osnovna načina konstrukcije PLC kontrolera:

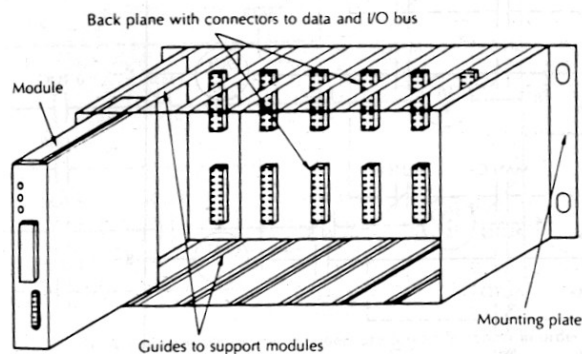
- (a) kompaktni PLC kontroleri i
- (b) modularni PLC sistemi.

Kompaktni PLC kontroleri su nezavisni, zatvoreni uređaji sa fiksnim brojem ulaza/izlaza, bez mogućnosti proširenja (Sl. 1-6). U jednom kućištu, obično manjih dimenzija, smešteni su: izvor za napajanje, procesorska jedinica i ulazni i izlazni modul. Kompaktni PLC kontroleri predstavljaju ekonomično rešenje, predviđeno za upravljanje sistemima i procesima male složenosti. Tipično, poseduju do 16 ulaza i 16 izlaza i memoriju od nekoliko KB.



Sl. 1-6 Kompaktni PLC. (Uočavaju se priključci za ulaze, izlaze i napajanje, konektor za vezu sa programatorom i slot za memorijski modul)

Modularni PLC sistemi se sastoje od većeg broja modula koji su smešteni unutar mehaničkog okvira, tj. šasije, koji se zove rek (*rack*) (Sl. 1-7). Rek poseduje veći broj slotova za smeštanje modula. Svaki slot čini par vođica duž gornje i donje stranice reka, koje služe za mehaničko učvršćenje modula kao i konektor na zadnjoj ploči reka preko koga se modulu priključuje na zajedničku magistralu izvedenu na štampanoj ploči zadnje stranice reka. Po pravilu, prvi slot je namenjen modulu izvora napajanja, koji se priključuje na mrežni napon (220Vac) i generiše jednosmerne napone potrebne za rad ostatka sistema. Sledeći, drugi slot se koristi za modul logičke jedinice, tj. procesorski modul koji izvršava korisnički program i upravlja radom ostalih modula. Preostali slotovi se koriste za module specijalne namene, kao što su U/I moduli, memorijski moduli i sl. Ovakav način konstrukcije omogućava lako proširenje sistema. Na primer, ako je potrebno povećati broj ulaza/izlaza dovoljno je ugraditi dodatni U/I modul. Ili, ako zbog povećanih zahteva obrade, postojeći procesorski modul više nije odgovarajući, procesorski modul se može zameniti novim, moćnijim, a da pri tome ostali moduli ne moraju biti zamenjeni. Proj slotova u jednom PLC reku je, tipično, od 4 do 16. Mogućnost proširenja PLC sistema nije ograničena samo na jedan rek. Uz pomoć posebnih modula za proširenje moguće je povezati dva ili više reka, što omogućuje da se jednim procesorskim modulom upravlja velikim brojem dodatnih modula.



Sl. 1-7 Modularni PLC.

1.3.2 CPU i PM

CPU je "srce" PLC sistema. Na Sl. 1-8 je prikazana tipična CPU jedinica u spoju sa programtorom/monitorom (PM) oblika ručnog terminala. CPU jedinica može biti veća ili manja od one prikazane na Sl. 1-8, zavisno od veličine i složenosti procesa kojim upravlja.

Jedan od glavnih parametara o kome se mora voditi računa prilikom izbora CPU jedinice jeste veličina interne memorije. Za upravljanje jednostavnim procesom dovoljan je mali PLC sa ograničenom memorijom; za upravljanje većim sistemom, neophodan je već PLC sa više memorije koji podržava naprednije funkcije. Kod nekih PLC-ova postoji mogućnost da se naknadno ugradi dodatna memorija; kod drugih ta mogućnost ne postoji.



Sl. 1-8 CPU i PM.

Na samoj CPU jedinici obično se mogu naći različiti konektori za kablove koji služe za povezivanje sa drugim PLC-ovima.

Mnoge CPU jedinice poseduju ugrađenu tzv. *backup* bateriju, koja omogućavaju da se u slučaju nestanka eksternog napajanja sačuva učitani korisnički lider program. Za razliku od operativnog sistem PLC-a koji čuva se u permanentnoj memoriji (ROM), korisnički program se čuva u RAM-u, koji, kao što znamo, gubi upisani sadržaj onog momenta kada se napajanje isključi. Kada PLC detektuje prestanak eksternog napajanja, *backup* baterija se automatski uključuje da bi se obezbedio napon napajanja za RAM i tako sačuva njegov trenutni sadržaj do dolaska eksternog napajanja.

CPU jedinica se postavlja u odgovarajući režim rada pomoću jednog posebnog više-položajnog prekidač. Da bi se sprečio neovlašćen pristup PLC-u, za okretanje ovog prekidača neophodan je ključ. Tipično, pozicije prekidača su:

OFF - sistem nije operativan niti se može programirati

RUN - sistem je operativan (radi), ali se ne može programirati.

DISABLE - Isključuje sve izlaze, tj. postavlja ih u neaktivno stanje.

MONITOR - uključuje ekran na kome će se prikazivati trenutne operativne informacije

RUN/PROGRAM - Sistem je operativan, a modifikacije programa su dozvoljene. Ovaj način rada se mora oprezno koristiti, jer se svaka modifikacija programa trenutno manifestuje u radu sistema.

PROGRAM - sistem nije operativan, ali se može programirati ili reprogramirati.

Na Sl. 1-9(a) je prikazan tipičan programator/monitor (PM) sa CRT ekranom. Tipičan ručni terminal prikazan je na Sl. 1-9(b). PM sa Sl. 1-9(a) može pružiti potpuniji uvid u rad i trenutni status PLC-a, ali zato ručni terminal prenosiv.



(a)



(b)

Sl. 1-9 (a) PM sa CRT ekranom; (b) PM oblika ručnog terminala.

PM se povezuje sa CPU-om pomoću kabla. Nakon što je CPU programiran, PM nije više neophodan i veza sa CPU-om se može raskinuti, isti PM se može koristiti za programiranje svih CPU-u u jednom proizvodnom pogonu.

1.3.3 PLC ulazni i izlazni moduli

PLC pribavlja informacije iz okruženja posredstvom ulaznih, a predaje informacije okruženju putem izlaznih modula. Pristupne tačke (terminali) ulaznog modula primaju signale senzora i pretvarača. Na pristupnim tačkama izlaznog modula generiše se napon za pobudu aktuatora i uređaja za indikaciju (svetiljke, displeji, ...)

Modul može imati 4, 8, 12 ili 16 terminala. Modul može biti ulazni, izlazni ili kombinovani (U/I) sa podjednakim ili različitim brojem ulaznih i izlaznih terminala (npr. 12 ulaza i 8 izlaza).

Kod manjih sistema, CPU, ulazni i izlazni moduli se smeštaju u isti rek. Kod većih PLC sistema, ulazni i izlazni moduli su smešteni u posebne rekove koji su sa CPU-om povezuju pomoću odgovarajućeg višezičnog kabla.

Ako je u jednom reku smešteno više modula, tada je neophodno podesiti (konfigurisati) adresu (redni broj) svakog od njih. Način na koji se postavlja adresa modula zavisi od tipa PLC sistema. Kod nekih PLC sistema, to se postiže pomoću niza mikroprekidača na samom modulu. Kod drugih, adrese se postavljaju programski, pomoću PM-a. Kod manjih PLC sistema, adresa modula se ne postavlja, već je unapred određena pozicijom modula u reku.

Najvažnija karakteristika U/I modula je opseg napona ili struje koji modul prihvata, odnosno generiše. Naponski i strujni opseg modula mora biti usaglašen sa električnim karakteristikama sistema ili uređaja sa kojima se PLC modul povezuje. Na primer, ulazni modul deklarisan za 24V DC, neće raditi ako uređaj s kojim je povezan daje 120V AC - šta više, biće oštećen. Ili, aktuator koji se pobuđuje strujom od 4.5A, ne može se povezati za izlazni modul koji može da da samo 2A. Da bi se zadovoljili najrazličitiji zahtevi u pogledu naponskih i strujnih opsega, proizvođači PLC sistema nude palete ulaznih i izlaznih modula deklariranih za različite naponske/strujne opsege.

Pored diskretnih ulaznih i izlaznih modula (prihvataju i generišu diskretne - digitalne, tj. ON/OFF signala), u upotrebi su i i analogni ulazni i izlazni moduli koji prihvataju i generišu analogne signale. Takvi moduli poseduju ugrađene A/D, odnosno D/A konvertore.

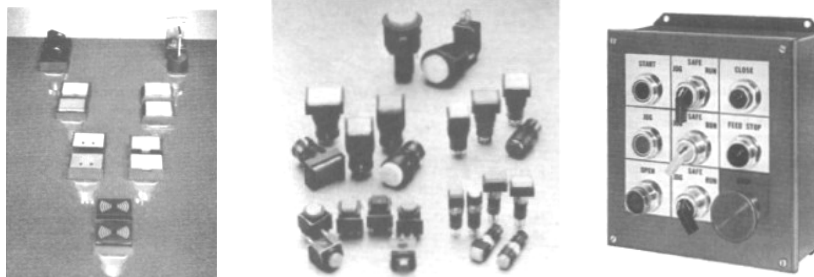
Pored diskretnih i analognih ulaznih i izlaznih PLC modula, proizvođači PLC sistema, takođe, nude i najrazličitije PLC module specijalne namene, kao što su: modul za PID kontrolu, radio-frekvencijski modul, modul za pozicioniranje, komunikacioni moduli, moduli za vizuelnu kontrolu i td..

U nekim slučajevima, proces koji se upravlja PLC-om može se nalaziti na velikom rastojanju od CPU jedinice. Usled prenosa duž vodova velike dužine, normalni ulazni i izlazni signali bili bi oslabljeni do nivoa da postaju neprepoznatljivi za ulazni PLC modul, odnosno za uređaj koji se pobuđuje izlaznim PLC modulom. U takvim situacijama, postupa se tako što se rek sa ulaznim/izlaznim modulima postavlja u blizini samog procesa, a sa CPU-om povezuje komunikacionim interfejsima za prenos podataka, pri čemu se kao prenosni medijum koriste žičani ili fiber optički kablovi.

1.4 Uređaji sa kojima se PLC povezuje

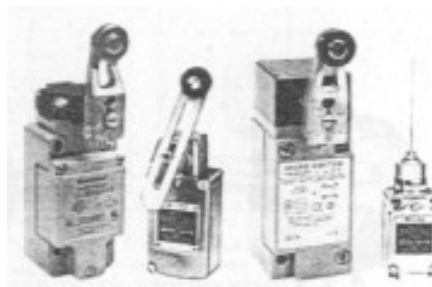
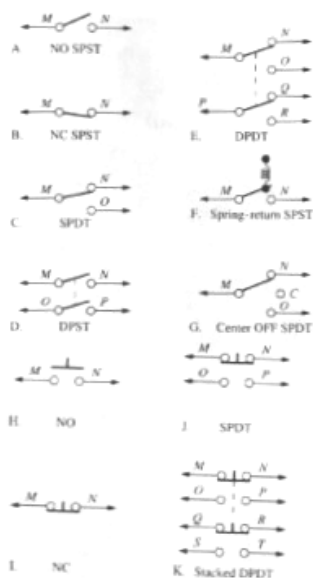
1.4.1 Ulazni ON/OFF uređaji

Na Sl. 1-10 su prikazani neki od brojnih tipova ulaznih ON/OFF uređaja (prekidača) koji se mogu povezati na ulazni PLC modul.



Sl. 1-10 Različiti tipovi prekidača i tastera.

Na Sl. 1-11(a) su, u obliku šematskih simbola, prikazane različite konfiguracije prekidača. Varijante A – G prikazuju preklopnike, dok su varijante H – K tasteri. Preklopnici, imaju dva položaja (kao prekidač za svetlo u sobi), a okretanjem ili pritiskom se prebacuju iz jednog u drugi položaj. Tasteri (kao taster za zvono) imaju trenutno dejstvo koje traje samo dok su pritisnuti. Kada se sprežu sa PLC-om, jedan kraj prekidača je povezan na masu, a drugi na terminal (tačku) ulaznog modula. Na primer, za preklopnike tip A i B, kraj M treba povezati na masu, a kraj N na terminal ulaznog modula. Jedan položaj prekidača se uvek smatra normalnim (podrazumevanim). U normalnom položaju prekidač može biti otvoren (kontakti prekidača nisu spojeni), na šta ukazuje oznaka **NO** (*Normally Open* - normalno otvoren) ili zatvoren (kontakti prekidača su spojeni), na šta ukazuje oznaka **NC** (*Normally Closed* - normalno zatvoren).



(a)

(b)

Sl. 1-11 Prekidači i tasteri: (a) Električni prekidači i tasteri; (b) granični prekidači.

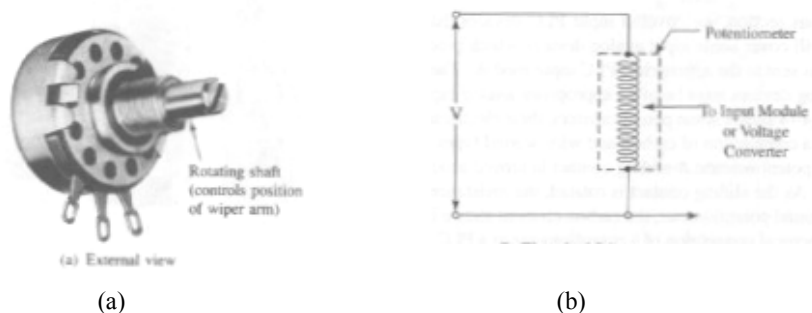
Druga vrsta ulaznih ON/OFF uređaja su granični prekidači (Sl. 1-11(b)). Granični prekidači nisu predviđeni da se ručno uključuju/isključuju, već poseduju specifične mehaničke aktuatore koji se u fizičkom kontaktu sa nekim predmetom ili objektom pomeraju i zatvaraju/otvaraju kontakte prekidača.

U grupu ulaznih ON/OFF uređaja spadaju i brojni tipovi diskretnih senzora, koje možemo zamisliti kao prekidače koji se zatvaraju/otvaraju pod dejstvom toplote, svetlosti, magnetskog polja, vlažnosti, odnosno kada temperatura, intenzitet svetlosti, jačina magnetnog polja ili nivo vlažnosti dostigne unapred podešenu graničnu vrednost.

1.4.2 Ulazni analogni uređaji

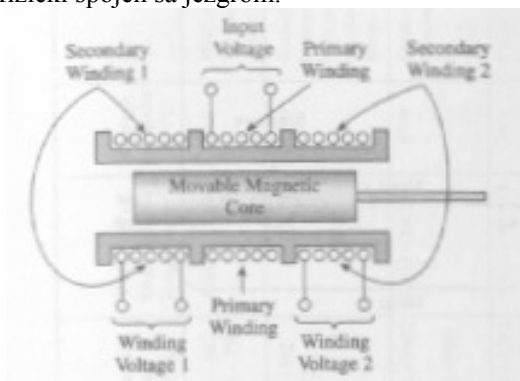
Za razliku od ulaznih ON/OFF uređaja, koji mogu biti samo uključeni (ON) ili isključeni (OFF), ulazni analogni uređaji generišu promeljive električne signale koji se vode na odgovarajuće (analogne) PLC ulazne module.

Na Sl. 1-12(a) je prikazan *ugljeni*, a na Sl. 1-12(a) žičani potencijometar. U oba slučaja, otpornosti između srednjeg i krajnjih izvoda se menja okretanjem klizača, a shodno tome i izlazni analogni, jednosmerni (DC) napon.



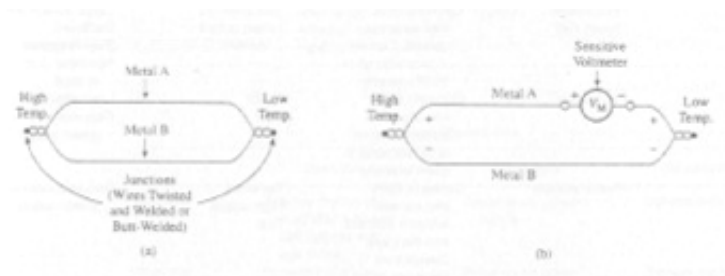
Sl. 1-12 Potenciomatar: (a) ugljeni, (b) žičani.

Na Sl. 1-13 je prikazan linearni varijabilni diferencijalni transformator (LVDT). Naizmenični (AC) izlazni napon LVDT-a varira, kako se jezgro pomera (uvlači/izvlači). Zavisnost između izlaznog napona i pozicije jezgra je linearna, tako da se LVDT može koristiti za indikaciju pozicije nekog mehaničkog sklopa koji je fizički spojen sa jezgrom.



Sl. 1-13 Linearni varijabilni diferencijalni transformator - LVDT

Za indikaciju (merenje) temperature, najčešće se koriste termoparovi (Sl. 1-14). Termopar je spoj dva metala. Između spojeva se generiše napon koji zavisi od razlike temperatura spojeva. Ovaj napon je obično veoma mali (reda nekoliko desetina $\mu\text{V}/^\circ\text{C}$), a uz to zavisnost napona od temperature nije linearna.



Sl. 1-14 Princip termopara.

Osim navedenih, u industriji su u upotrebi i brojni drugi analogni ulazni uređaji, kao što su temperaturni, optički, magnetski, elektro-mehanički pretvarači, senzori protoka, pritiska nivoa tečnosti i sl.

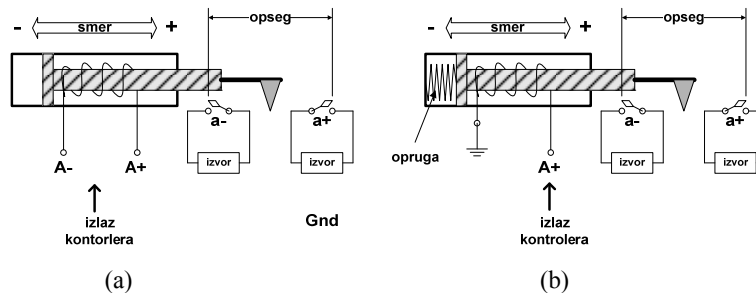
Da bi se omogućila sprega najrazličitijim tipovima anlognih ulaznih uređaja, proizvođači PLC-ova nude serije tipiziranih ulaznih PLC modula, za jednosmeran/naizmenični ulazni napon ili struju, za različite opsege napona/struje, za spregu sa specifičnim anlognim ulaznim uređajima (kao što su termoparovi) i sl.

1.4.3 Izlazni ON/OFF uređaji

Izlazni PLC moduli se mogu povezati sa mnoštvom najrazličitih tipova izlaznih ON/OFF uređaja (ili ON/OFF aktuatora). Jedan od najvažnijih takvih uređaja je solenoid. *Solenoid* je elektromehanički aktuator, čijim radom se upravlja pomoću elektromagnetne sile proizvedene u namotaju. U principu, rad solenoida zasniva se na struji koja postoji u namotaju i koja proizvodi magnetno polje. U zavisnosti od smera struje, menja se i smer sile magnetnog polja koja privlači gvozdeno jezgro ka centru namotaja ili ga odbija od centra. Postoje dva tipa solenoidnih aktuatora:

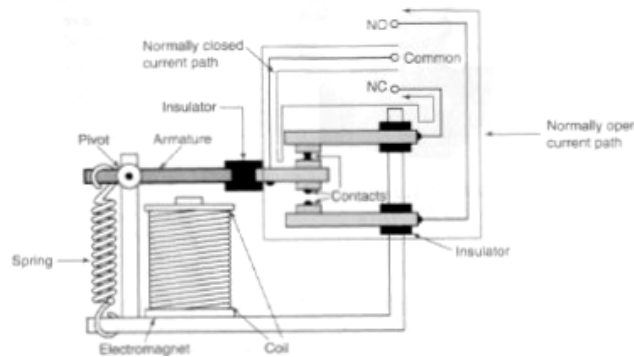
- *Jednosmerni solenoid* – kod koga postoji samo jedan izvod za napajanje, tako da struja ima uvek isti smer, što znači da se i jezgro pod dejstvom magnetne sile može pomerati samo u jednom smeru. U odsustvu napajanja solenoida, mehanička opruga vraća jezgro u početni položaj.
- *Dvosmerni solenoid* – kod koga postoje dva izvoda za napajanje, tako da smer struje, odnosno odgovarajuće magnetne sile zavisi od toga na koji izvod je priključeno napajanje. U skladu sa time i jezgro se kreće u jednom od dva moguća smera. Ukoliko se napajanje dovede na oba izvoda, jezgro se neće pomerati, Isto tako, ukoliko ni na jednom kraju nema napajanja, jezgro će ostati u zatečenom položaju, uz uslov da ne postoji neka mehanička sila (npr. sila zemljine teže, ako je solenoid u vertikalnom položaju) koja bi izazvala njegovo kretanje. Drugim rečima, u odsustvu napajanja, solenoid se nalazi u slobodnom stanju.

Šematski dijagram oba tipa solenoida prikazan je na Sl. 1-15. Potrebno je da se istakne da solenoid po pravilu ima i dva granična prekidača koji omogućavaju da se detektuje kada jezgro dođe u krajnji desni ili krajnji levi položaj. Solenoidi koriste kao mehanički aktuatori za upaljanje drugih (mehaničkih) uređaja, kao što je ventil.



Sl. 1-15 Solenoid: (a) jednosmerni; (b) dvosmerni.

Na Sl. 1-16 je prikazano elektro-mehaničko rele, još jedan uobičajeni tip ON/OFF uređaja kojim se može upravljati putem izlaznog PLC modula. Izlazni PLC modul se povezuje sa kalemom relea. Struja koju daje PLC modul, a koja protiče kroz kalem stvara magnetno polje koje prilači kotvu i zatvara (ili otvara) kontakt, koji, dalje upravlja nekim drugim urađajem. Prednost korišćenja relea umesto direktnog korišćenja kontakta izlaznog modula su: (1) rele obično ima, ne samo jedan, više kontakta od kojih su neki NO, a drugi NC tipa; (2) kontakti relea su električno izolovani od PLC modula i (3) kroz kontakte relea može proticati veća struja, a kontakt može prekidati više napone nego kontakt izlaznog modula. Releji za veoma velike struje se nazivaju kontaktorima.



Sl. 1-16 Standardno rele.

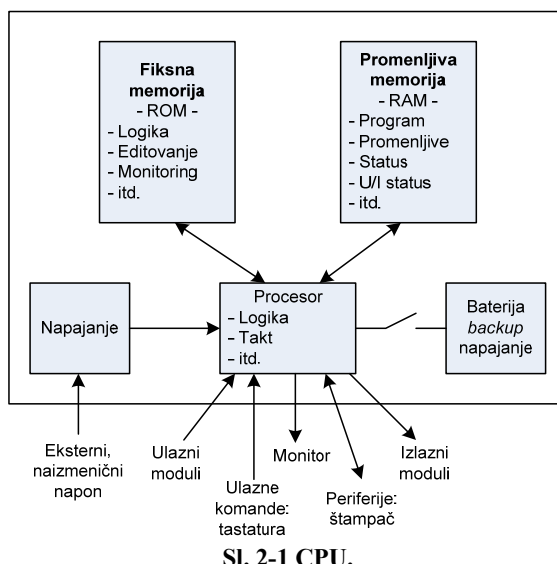
Starter motora je varijanta relea/kontaktora. Kontakti startera motora su konstruisani na način da mogu da izdrže velike povratne struje koje se javljaju kada se sa induktivnih namotaji motora ukloni napon (tj. kada se motor isključi). Obično, starter motora poseduje ugrađena električna kola za detekciju preopterećenja motora koja isključuju napajanje motora kada struja prekorači dozvoljenu maksimalnu vrednost. Takođe, postoje dualni starteri motora koji se koriste za "napred-nazad" upravljanje motorom putem kojih je moguće ne samo uključiti/isključiti motor, već i promeniti smer okretanja motora (promenom smera struje kroz namotaje motora). Osim startera za jednosmerne i naizmenične motore, postoje i starteri za sinhronne (trofazne) motore.

1.4.4 Izlazni analogni uređaji

Osim izlaznih ON/OFF uređaja kod kojih se upravljanje svodi na prosto uključivanje/isključivanje, u upotrebi su i analogni izlazni uređaji (ili analogni aktuatori) kod koji ma se upravlja na kontinualan način, pomoću analognog (promenljivog) napona ili struja. Primer takvog aktuatora je servo motor čija se brzina rotiranja reguliše kontinualno, pomoću posebne vrste analognog izlaznog PLC modula. Još jedan primer analognog aktuatora je hidraulični servo ventil. Za razliku od ON/OFF ventila koji se može samo da se otvori ili zatvori, pozicija (tj. otvorenost) servo ventila se može fino, kontinualno regulistati. Po pravilu, upravljanje analognim izlaznim uređajem podrazumeva postojanje povratne informacije o trenutnoj poziciji aktuatora. Za indikaciju pozicije koriste se uređaji koji se nazivaju enkoderima.

2 CPU jedinica PLC kontrolera

Bez obzira na veličinu PLC-a, centralna procesorska jedinica, tj. CPU, ili samo *procesorska jedinica*, uvek objedinjuje mikroprocesor i memoriju. Kod većih PLC-ova, CPU sadrži samo mikroprocesor i memoriju, dok kod manjih, dodatno, sadrži U/I interfejs i izvor za napajanje. Takođe, moguće je da CPU sadrži mikroprocesor, memoriju i izvor napajanja, a da je U/I interfejs realizovan eksternim U/I modulima. Jedna takva organizacija, prikazana je blok dijagramom sa Sl. 2-1. Fiksna memorija sadrži program (ili programe) koje je u postavio proizvođač. Ovo su programi *operativnog sistema* PLC-a, koji imaju sličnu namenu, mada sa neuporedivo manjim mogućnostima, kao npr. DOS operativni sistem kod starijih PC računara, Windows kod novijih ili UNIX kod radnih stanica. Programi operativnog sistema su smešteni u posebnom memorijskom integrisanom kolu tipa ROM. Kao što znamo, sadržaj ROM-a se ne može brisati niti menjati, a ostaje nepromenjen i nakon isključenja napajanja. Promenljiva memorija, realizovana pomoću RAM memorijskih čipova, podeljena je na veći broj sekcija koje sadrže korisnički program, programske promenljive, trenutne susne informacije kontrolera i td. (o organizaciji memorije PLC kontrolera, nešto kasnije u ovom poglavlju).



Sl. 2-1 CPU.

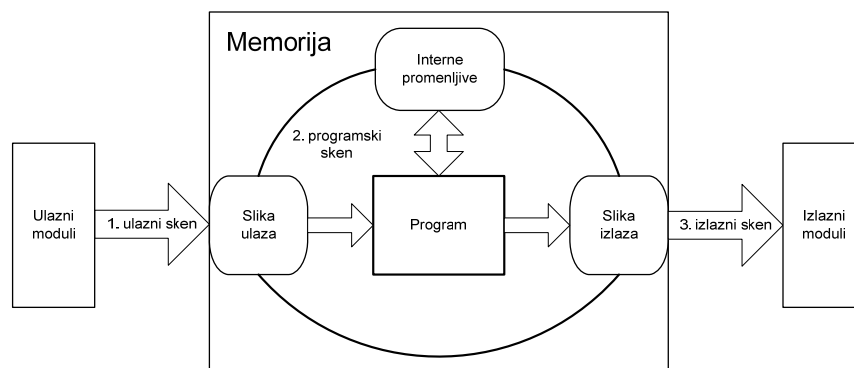
2.1 Sken ciklus

Operativni sistem PLC kontrolera je projektovan tačno za određenu vrstu primene. Naime, predpostavlja se da će u svojoj osnovnoj formi, PLC biti korišćen za realizaciju izvesnih logičkih funkcija koje preslikavaju signale sa senzora u signale koji se prenose na aktuatora. Otuda se od PLC-a očekuje da *periodično očitava* (unos) signale sa senzora, *izvršava* određen broj aritmetičko-logičkih operacija (u skladu sa zadatom funkcijom) čiji rezultati se *prenose* na izvršne organe ili neke druge indikatorske uređaje. Pored toga, sa istom ili nekom drugom učestanošću, PLC treba da održava *komunikaciju* (razmenjuje podatke) sa nekim drugim računarskim sistemima u mreži. Polazeći od ovog zahteva, operativni sistem PLC kontrolera projektovan je tako da, u toku rada sistema, automatski obezbedi ciklično ponavljanje navedenih aktivnosti (*Sken ciklus*) kao što je to ilustrovano na Sl. 2-2.



Sl. 2-2 Sken ciklus PLC kontrolera.

Sken ciklus započinje sa **ulaznim skenom** u okviru koga PLC očitava sadržaj ulaznih linija (registara ulaznih modula). Očitani podaci se prenose u određeno područje memorije – **slika ulaza**. Zatim se aktivira **programski sken** u okviru koga procesor izvršava programske naredbe kojima su definisane odgovarajuće aritmetičko-logičke funkcije. Podaci (operandi) koji se koriste u programskim naredbama uzimaju se iz memorije i to iz područja označenog kao **slika ulaza** (ako su operandi ulazni podaci) ili iz područja gde se smeštaju interne promenljive. Rezultati obrade se smeštaju u posebno područje memorije – **slika izlaza**. Važno je istaći da se pri izvršavanju programskih naredbi podaci ne uzimaju direktno sa ulaznih modula, niti se rezultati direktno postavljaju na izlazne module, već program razmenjuje podatke isključivo sa memorijom (Sl. 2-3). Po završetku programskog skena, operativni sistem PLC kontrolera aktivira **izlazni sken** u okviru koga se podaci iz **slike izlaza** prenose na izlazne linije (registre izlaznih modula). Na ovaj način stvara se utisak da je PLC sve operacije definisane programom obavio u isto vreme. Četvrti deo sken ciklusa – **komunikacija** - namenjen je realizaciji razmene podataka sa uređajima koji su povezani sa PLC-om. Nakon toga, operativni sistem dovodi PLC u fazu **održavanja** u okviru koje se ažuriraju interni tajmeri i registri, obavlja upravljanje memorijom kao i niz drugih poslova vezanih za održavanje sistema, o kojima korisnik i ne mora da bude informisan. U zavisnosti od tipa ugrađenog mikroprocesora ulazni i izlazni sken ciklus izvršavaju se u vremenu reda milisekundi (od 0,25ms do 2,56ms). Trajanje programskog skena, svakako zavisi od veličine programa.



Sl. 2-3 Razmena podataka za vreme sken ciklusa.

2.2 Osnovne karakteristike CPU jedinice

Kao što je već rečeno CPU modul sadrži mikroprocesor i memoriju. Mikroprocesor obuhvata aritmetičko-logičku jedinicu (*ALU*), registre i upravljačku jedinicu. U funkcionalnom smislu mikroprocesor PLC-a se bitno ne razlikuje od mikroprocesora bilo kog mikroračunara opšte namene. Međutim, razlika postoji u programskom jeziku; dok se za programiranje mikroračunara koristi asemblerski jezik ili neki viši programski jezik (npr. C), za programiranje PLC-ova se koristi, kao što je ranije rečeno, jezik lader dijagrama. U osnovi, osnovna razlika se ogleda u skupu naredbi koji je odabran tako da se zadovolje osnovni zahtevi u pogledu korišćenja PLC-a.

Osnovne karakteristike procesorskog modula izražavaju se preko sledećih elemenata:

Memorija(RAM) – karakteriše se svojom veličinom, mogućnošću proširenja i konfigurisanja za smeštanja programa ili podataka.

U/I tačke – karakteriše se najvećim brojem lokalnih U/I adresa koje podržava procesor u toku ulaznog i izlaznog skena, kao i mogućnošću proširenja preko **udaljenih U/I**. (Pod udaljenim U/I podrazumeva se posebna šasija koja sadrži U/I module koji razmenjuju podatke sa PLC-om).

Komunikacione opcije - odnose se na raznovrsnost uređaja za spregu (komunikacionog interfejsa) koji podržavaju različite topologije mreža i različite komunikacione protokole.

Opcije trajnog pamćenja - odnose se na raspoloživost različitih tipova memorijskih EPROM modula koji obezbeđuju trajno pamćenje podataka.

Performansa - specificira se preko *vremena programskog skeniranja* potrebnog za izvršenje 1Kbajt programa, preko *vremena potrebnog za ulazni i izlazni sken*, kao i *vremena izvršavanja jedne bit naredbe*.

Programiranje - specificira se u odnosu na broj podržanih različitih naredbi ledar jezika.

2.2.1 CPU jedinica kontrolera SLC 500

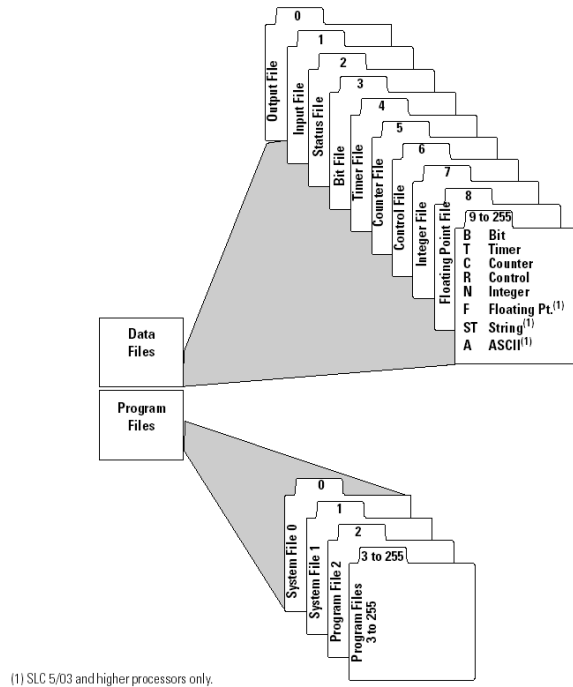
U tabeli T. 1 dat je pregled osnovnih karakteristika procesora SLC 5. Oznaka “UP” odnosi se na “naredbe korisničkog programa”, dok je sa “DW” označena “podatak dužine jedne reči”.

T. 1 Osnovne karakteristike procesorskog modula SLC 5.

Specifikacija	SLC 5/01	SLC 5/02	SLC 5/03	SLC 5/04
Memorija	1K UI ili 4K DW 4K UI ili 16K DW	4K UI ili 16K DW	12K UI i 4K DW	12K UI i 4K DW 28K UI i 4K DW 60K UI i 4K DW
Lokalni U/I	256 digitalnih	480 digitalnih	960 digitalnih	960 digitalnih
Udaljeni U/I	nema	<i>Kapacitet zavisi od vrste napajanja i veličine programske memorije - najviše može biti 4096 ulaza i 4096 izlaza -</i>		
Trajno pamćenje	EEPROM ili UVPRM	EEPROM ili UVPRM	Fleš EPROM	Fleš EPROM
Tipično vreme programskog skeniranja	8 ms/K	4.8 ms/K	1 ms/K	0.9 ms/K
Izvršavanje bit naredbe	4 μs	2.4 μs	.44 μs	.37 μs
Broj mašinskih naredbi	52	71	99	99

2.3 Organizacija memorije

Operativni sistem PLC kontrolera, koji realizuje sken cikluse, upravlja i zauzećem RAM memorije, koja je organizovana na poseban način. U principu, RAM memorija se deli na *program files* (*programske datoteke*) i *data files* (*datoteke podataka*) Sl. 2-4. Skup programa i datoteka podataka koje su formirane za jednu aplikaciju čini *processor file* (*procesorsku datoteku*). Ona sadrži sve naredbe, podatke i specifikaciju modula koji su relevantni za datu aplikaciju, odnosno korisnički program. Procesorska datoteka čini jednu celinu koja se može prenositi sa jednog procesorskog modula na drugi. To zapravo znači da se jedna aplikacija može razviti na jednom sistemu i zatim u celini preneti i koristiti na drugom sistemu.



Sl. 2-4 Organizacija memorije PLC kontrolera.

2.3.1 Programske datoteke

Programske datoteke sadrže informacije o samom kontroleru, glavni korisnički program i potprograme. Svaka aplikacija (procesorska datoteka) mora da ima sledeće tri programske datoteke:

System Program – sistemski program (file 0) - sadrži različite informacije o samom sistemu kao što su tip procesora, konfiguracija U/I modula, ime procesorske datoteke, lozinku i niz drugih relevantnih podataka.

Reserved – datoteka rezervisna za potrebe operativnog sistema (file 1)

Main Ladder Program – glavni leder program (file 2) – program koji formira sam korisnik i u okviru koga se definiše niz operacija koje SLC treba da izvede.

Subroutine Ladder Program - potprogrami (file 3 - 255) – korisnički potprogrami koji se aktiviraju u skladu sa naredbama za njihovo pozivanje koje se nalaze u glavnom programu.

2.3.2 Datoteke podataka

Datoteke podataka sadrže podatke koji se obrađuju pomoću naredbi leder programa. Pri tome se pod pojmom *podaci* podrazumevaju konvertovane (numeričke) vrednosti signala koji se preko ulazno/izlaznih modula unose u kontroler, ili se iz kontrolera prenose na izlazne uređaje, kao i interne promenljive koje se koriste kao operandi u različitim operacijama.

Datoteke podataka organizovane su u skladu sa tipom promenljivih koje sadrže. To zapravo znači da jedna datoteka sadrži samo jedan tip (vrstu) podataka. Jedna procesorska datoteka može da ima najviše 256 datoteka podataka.

2.3.2.1 Tipovi promenljivih i datoteka

Osnovna karakteristika datoteke podataka je njen *tip*. Kao što je već istaknuto tip datoteke, zapravo ukazuje na vrstu promenljivih koje se u njoj pamte. To nadalje podrazumeva da tip datoteke ujedno određuju i njenu organizaciju, koja zavisi od vrste podatka i usvojenog načina za njegovo prikazivanje u računaru.

Datoteka se označava pomoću *rednog broja*, koji jednoznačno određuje mesto te datoteke u nizu datoteka podataka koje se nalaze u procesorskoj datoteci i *slova* kojim se identifikuje tip datoteke. Prvih devet datoteka imaju unapred definisan tip koji ne može da se menja. Tipove preostalih datoteke korisnik sam odabira i definiše u skladu sa aplikacijom koju razvija.

File 0 – Tip O - output (izlaz) – sadrži *sliku izlaza*; sadržaj datoteke se prenosi na izlazne linije za vreme izlaznog skena.

File 1 – Tip I - input (ulaz) - sadrži *sliku ulaza*; u ovu datoteku se za vreme ulaznog skena smeštaju vrednosti sa ulaznih linija.

File 2 – Tip S - status - sadrži podatke vezane za rad kontrolera.

File 3 – Tip B - bit – sadrži interne promenljive *bit* tipa.

File 4 – Tip T - timer (časovnik) - sadrži podatke koji se koriste za interne *časovnike*.

File 5 – Tip C - counter (brojač) - sadrži podatke koji se koriste za interne *brojače*.

File 6 – Tip R - control (upravljanje) – sadrži dužinu, položaj pokazivača i bitove statusa za određene naredbe kao što su naredbe za pomeranje sadržaja registara i sekvenci.

File 7 – Tip N - integer (celobrojna) – sadrži podatke celobrojnog tipa.

File 8 – Tip F - floating point (realna) - sadrži podatke predstavljene u formatu pokretnog zareza kao 32-bit brojeve u opsegu ($\pm 1.1754944e-38$ to $\pm 3.40282347e+38$).

File 9 do file 255 – Tip definiše korisnik - *korisničke datoteke* – ove datoteke definiše korisnik kao datoteke tipa **B, T, C, N**.

Za procesore tipa *SLC 5/03* i *SLC 5/04* korisnik može da definiše i datoteke tipa **F, St - string, A - ASCII**. Pored toga *datoteka 9* se može koristiti i kao *komunikacioni interfejs*.

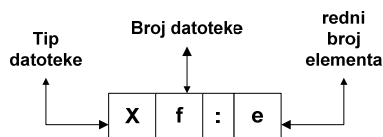
Specijalni U/I moduli imaju takođe memoriju u kojoj se, pored ostalog, nalaze i njima pridružene datoteke. One se označavaju kao **M0** i **M1 files**, i njihova organizacija zavisi od specifičnosti pojedinog modula.

2.3.2.2 Element datoteke

Osnovna jedinica datoteke je jedan *element*. Svaki elemenat se sastoji iz nekoliko *16-bitnih reči*. Broj reči koje čine jedan element zavisi od tipa datoteke, odnosno vrste podataka koji se u nju smeštaju.

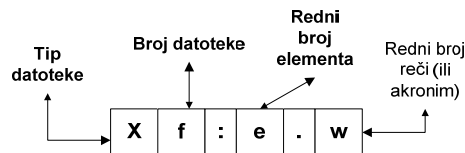
Kao što je već istaknuto, podaci koji su smešteni u datotekama predstavljaju operande (promenljive) koje se koriste u pojedinim programskim naredbama. Ove promenljive se u programu pozivaju preko svojih simboličkih imena koja predstavljaju *logičke adrese*. Pri tome, adrese omogućavaju da se pozove ne samo elemenat u celini, već i njegov deo. To znači da se mogu adresirati pojedine reči u okviru elementa ili pojedini bitovi u okviru reči. Budući da su podaci u izvesnom smislu hijerarhijski organizovani: 1 elemenat sadrži nekoliko reči, a 1 reč 16 bitova, to su i odgovarajuće adrese strukturirane po istom hijerarhijskom principu. Pojedine reči i bitovi u nekim datotekama imaju i pridružene akronime, što dodatno olakšava njihovo korišćenje.

- *Adresa elementa* – u principu, svaki element u okviru datoteke se identifikuje pomoću njegovog relativnog položaja u odnosu na početak datoteke (multi, prvi, drugi, ... element). U skladu sa time adresa elementa ima izgled kao na **Sl. 2-5**.



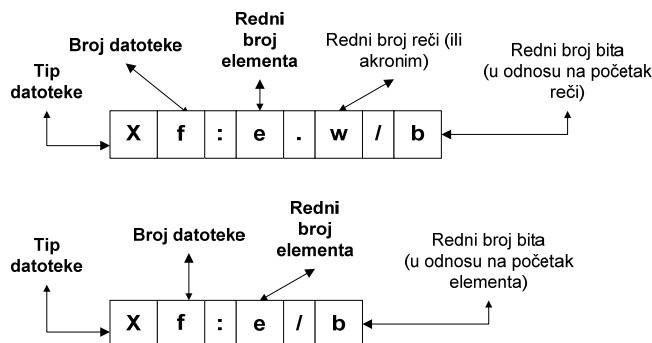
Sl. 2-5 Adresa elementa.

- *Adresa reči* – jedna reč elementa se identifikuje ili pomoću relativnog položaja te reči u okviru elementa, ili pomoću posebnog akronima (ukoliko je isti definisan). Format adrese jedne reči prikazan je na Sl. 2-6.



Sl. 2-6 Adresa reči

- *Adresa bita* – Jedan bit u okviru reči identifikuje se ili preko njegovog relativnog položaja u okviru te reči (*multi, prvi, drugi, ... bit brojano s desna u levo*) ili preko relativnog položaja u odnosu na početak odgovarajućeg elementa kome pripada reč čiji se bit adresira (Sl. 2-7).



Sl. 2-7 Adresa bita.

Dužine elemenata u pojedinim datotekama date su u tabeli T. 2. Potrebno je istaći da se teorijski u svakoj datoteci mogu adresirati i element u celini i njegove reči i bitovi. U tabeli T. 2 su, međutim, navedene samo one adrese koje sa aspekta vrste podatka i načina organizacije datoteke imaju smisla. Podrazumeva se pri tome da je:

- *adresirani bit* $0 \leq b \leq 15$
- *adresirni element* $0 \leq e \leq 255$

Može se uočiti da su iz tabeli izostavljene ulazne i izlazne datoteke (I i O). Ovo je učinjeno zato što one donekle odstupaju od navedenog pravila. Naime, kao što će se kasnije videti, kod ovih datoteka elementi mogu biti dužine od jedne ili dve reči, što zavisi od tipa U/I modula. Element ovih datoteka je zapravo određen slotom u šasiji u koji se modul postavlja.

T. 2 Veličine elemenata pojedinih datoteka

Tip datoteke	Dužina elementa	Adresira se
B, N, A	1 reč	bit "b" - Xf:e/b ;
		element "e" - Xf:e
F	2 reči	element "e" - Xf:e
T, C, R	3 reči	bit "b" u reči 0- Xf:e.0/b ; (indikator stanja su bitovi reči 0)
		reč "w" - Xf:e.w ; w = {1,2}, promenljive su u rečima 1 ili 2
St	42 reči	bit "b" u reči "w" - Xf:e.w/b
		reč "w" - " - Xf:e.w ; $0 \leq w \leq 41$
		element "e" - Xf:e

U cilju ilustracije organizacije datoteka posmatrajmo *binarnu datoteku – bit file*. Maksimalna veličina ove datoteke iznosi 256 elemenata. Svaki element je jedna 16-bitna reč, što znači da ova datoteka može imati najviše 4096 bitova. U skladu sa izloženim načinima adresiranja, jedan bit može biti adresiran pomoću rednog broja elementa (0 - 255) i rednog broja bita u okviru elementa (0 – 15) ili pomoću rednog broja bita u okviru datoteke (0 – 4095), kao što je to ilustrovano u tabeli T. 3. Bit označen sa X je jedanaesti bit u drugom elementu, odnosno četrdesetčetvrti bit u celoj datoteci. Prema tome njegova adresa je B3:2/11 ili B3/44. Iz tabele se takođe vidi da je moguće da se adresira i ceo element ove datoteke.

T. 3 - Datoteka 3 – Bit

Adresa elementa	Adrese pojedinih bitova	Data file 3 – Bit file															
		bitovi															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B3:0	B3:0/0 ... B3:0/15 B3/0 ... B3/15																
B3:1	B3:1/0 ... B3:1/15 B3/16 ... B3/31																
B3:2	B3:2/0 ... B3:2/15 B3/32 ... B3/47					X											
B3:3	B3:3/0 ... B3:3/15 B3/48 ... B3/63																
.	.																
.	.																
.	.																
B3:255	B3:255/0 ... B3:255/15 B3/4080 ... B3/4095																

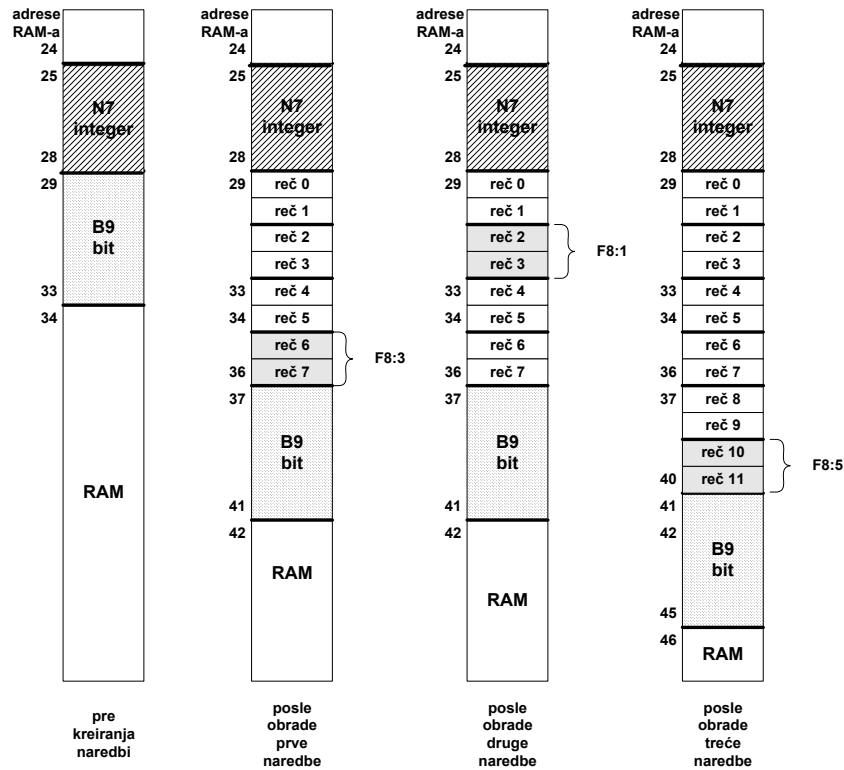
Potrebno je da se zapazi da će u svakoj datoteci tipa *bit* koju korisnik sam definiše adrese biti iste, s tim što će se umesto *broja datoteke* (3) staviti broj koji korisnik pridružuje svojoj datoteci (9 - 255).

2.3.2.3 Kreiranje datoteka i zauzeće memorije

Jedna datoteka podataka zauzima memorijski prostor koji obuhvata niz susednih reči. Broj reči koje zauzima jedna datoteka određen je najvećom adresom elementa te datoteke koji se koristi u programskim naredbama. Naime, sa izuzetkom *datoteke statusa S*, koja se kreira automatski, sve ostale datoteke podataka ne postoje a priori same po sebi, već se formiraju u toku kreiranja programa i to navođenjem odgovarajuće adrese u programu. Prvo navođenje broja datoteke inicijalizuje njeno kreiranje. Pri tome tip datoteke koji je naveden u adresi određuje broj reči koje se pridružuju jednom elementu, dok adresa elementa određuje niz konsektivnih elemenata za koje se u memoriji rezervišu prostor. Taj niz počinje od nule, a završava se sa adresom elementa koja je navedena u naredbi. Ako se kasnije pojavi veća adresa elementa iste datoteke onda se prethodno rezervisani prostor proširi tako da uključujući i tu adresu.

Predpostavimo da se u programskim naredbama pojavljuju redom adrese operanada F8:3, F8:1 i F8:5. U tom slučaju, pri kreiranju prve naredbe rezervisaće se memorijski prostor za datoteku 8 i zauzeti ukupno 8 reči (elementi 0,1,2 i 3 ; svaki element po dve reči). Kada se kasnije naiđe na adresu F8:1, ona neće prouzrokovati nikakve promene u zauzeću memorije, jer je memorijski prostor za taj element već zauzet. Međutim, adresa F8:5, dovešće do povećanja zauzetog prostora na ukupno 12 reči (6 elemenata), kao što je to ilustrovano na Sl. 2-8. Ovde je takođe predpostavljeno da su pre nailaska na pomenute tri naredbe, nekim drugim naredbama već kreirane datoteka N7 i korisnička datoteka 9, kojoj je pridružen tip B.

Potrebno je da se istakne da veličina datoteke nije određena stvarnim brojem elemenata koji se koriste, već najvećem adresom. U posmatranom primeru se tako koriste samo tri elementa datoteke 8, ali je zauzet prostor za 6 elemenata. To nadalje znači, da se pažljivim izborom adresa elemenata može ostvariti ušteda u zauzetom memorijskom prostoru.

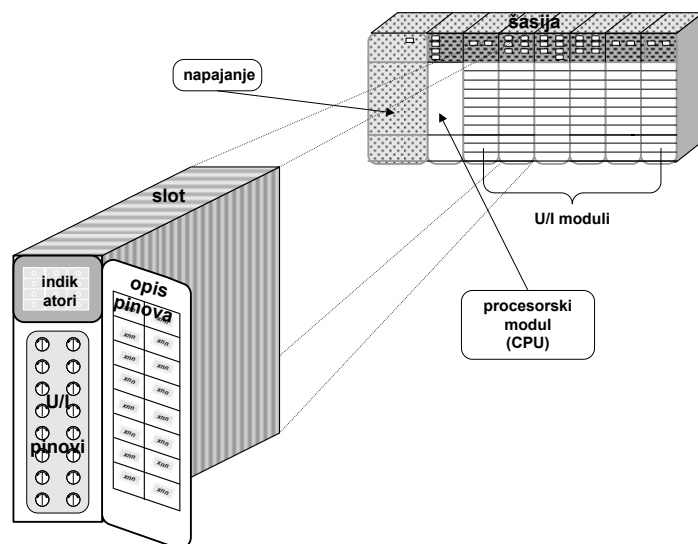


Sl. 2-8 Promena zauzeća RAM memorije pri kreiranju naredbi.

Kao što je već rečeno, dozvoljeno je kreiranje najviše 256 datoteka podataka. Samo se po sebi razume da će broj kreiranih datoteka zavistiti od promenljivih koje korisnik definiše u programu. Pri tome sve kreirane datoteke redaju se u nizu jedna iza druge. U formiranoj aplikaciji, datoteke podataka zauzimaju jedan neprekidan memorijski prostor. Redni brojevi ovih datoteka poredani su u rastućem nizu, ali ne moraju da čine kontinualni niz.

3 Diskretni U/I moduli

Uprkos činjenici da diskretni senzori i aktuatori koji se nalaze na nekom procesu ili postrojenju mogu imati veoma različite tehničke karakteristike, zahtev koji se postavlja pri njihovom vezivanju za kontroler je uvek isti. Naime, od kontrolera se očekuje da obezbedi konverziju digitalnog (binarnog) signala koji dolazi sa senzora u numeričku vrednost 0 ili 1 i da taj podatak smesti kao jedan *bit* na odgovarajuće mesto u memoriji, ili da očita numeričku vrednost (sadržaj) nekog bita u memoriji, da ga konvertuje u binarni signal koji se vodi na aktuator. Ova činjenica omogućila je projektovanje i izradu tipiziranih U/I kola koja su u stanju da obrađuju gotovo sve signale koji se sreću kod industrijske merne opreme i izvršnih organa. Pored toga, nekoliko U/I kola su grupisana zajedno i čine *Diskretni U/I modul*, čija veza sa kontrolerom se ostvaruje jednostavnim ubacivanjem u odgovarajući slot na šasiji.



Sl. 3-1 Digitalni U/I modul.

Izgled tipičnog U/I modula prikazan je na Sl. 3-1. Na prednjoj ploči U/I modula nalazi se određeni broj pinova (*terminal points*) za koje se vezuju izlazi sa mernih instrumenata, odnosno ulazi u izvršne organe. Svaki pin je zapravo ulazna ili izlazna tačka odgovarajućeg kola za spregu sa kontrolerom. U skladu sa time svaki pin se identifikuje svojim *tipom* (ulaz ili izlaz) i *brojem* koji određuje položaj U/I kola u okviru modula, i koji zapravo predstavlja *adresu* pina. Opis pinova dat je na unutrašnjoj strani vrata na modulu. Pored U/I pinova, na prednjoj ploči modula nalaze se i pinovi koji su interno povezani sa napajanjem (DC ili AC), sa zajedničkom (nultom) tačkom i sa zemljom. Način sprežanja pojedinog uređaja sa modulom zavisi od specifičnosti samog uređaja, kao i karakteristika modula, detaljna šema sprege vezivanja data je uz svaki modul.

Za vreme rada U/I modula, stanje svakog pina se prikazuje na odgovarajućem LED indikatoru. Indikator koji je povezan sa ulaznim pinom svetli ako je ulazni signal u stanju logičke jedinice. Indikator povezan sa izlaznim pinom svetli ako je, kao rezultat obrade programa, na izlazni pin postavljena logička jedinica.

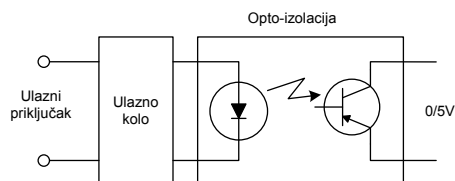
Postoje tri tipa U/I modula: *ulazni*, *izlazni* i *kombinovani ulazno/izlazni modul*. Oni se izrađuju sa različitim gustinama pinova (4, 8, 16 i 32 pina po modulu) i mogu se sprežati sa AC, DC i TTL naponskim nivoima.

Kao što se vidi na Sl. 3-1 U/I moduli se smeštaju u slotove na šasiji. Maksimalni broj modula koji se može direktno povezati sa jednim kontrolerom, zavisi od veličine šasije i broja slobodnih slotova. Budući da svaki slot ima svoju adresu unutar šasije, to znači da je samim stavljanjem modula u slot određena i njegova adresa. Konačno, kao što je već istaknuto, i svaki pin unutar jednog modula ima

svoju adresu. U skladu sa time svaki pin ima u okviru kontrolera *jedinstvenu adresu*, koja je određena *adresom slota* u koji se modul postavlja i *adresom pina* unutar modula. Potrebno je da se naglasi da je adresa pina određena automatski stavljanjem modula u šasiju kontrolera i da se ne može programski menjati.

3.1 Diskretni ulazni moduli

Ulazni modul PLC kontrolera obavljaju sledeća dva osnovna zadatka: (1) prihvataju ulazne signale iz spoljnjeg sveta i (2) štite logičku jedinicu od spoljnjeg sveta. Ulazni modul konvertuje logičke nivoe napona iz spoljnjeg sveta u logičke nivoe koje zahteva procesorska jedinica.



Sl. 3-2 Princip opto-izolacije.

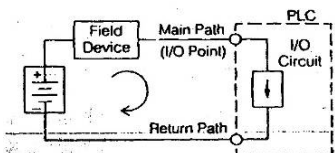
Za zaštitu logičke jedinice najčešće se koristi opto-izolacija (skraćeno od optička izolacija). Princip opto-izolacije ilustrovan je na Sl. 3-2. Kao što se može videti sa slike, između spoljnjeg priključka ulaznog modula i CPU jedinice ne postoji električna veza. Signal prisutan na ulazu modula se konvertuje u svetlost. Svetlost obasjava foto-prijenik koji se pod dejstvom svetlosti uključuje. Komponente koje se koriste za opto izolaciju, a objedinjuju u jednom zalivenom kućištu foto-predajnik (LED) i foto-prijemnik (foto-tranzistor) zovu se *opto-kapleri*.

Diskretni ulazni modul sadrže ugrađena kola za eliminaciju treperenja ulaznog signala. Naime, mnogi ulazni uređaji su mehaničke komponente i imaju kontakte. Pri otvaranju ili zatvaranju kontakta električni spoj se ne raskida, odnosno ne uspostavlja, trenutno već se javlja prelazni režim u toku koga se spoj veći broj puta upostavlja/raskida pre nego što se uspostavi konačno stanje. Ova pojava se zove treperenje kontakta i može imati neželjene posledice ukoliko se ne eliminiše, s obzirom da logička jedinica svaki treptaj može protumačiti kao novu aktivaciju ulaznog signala.

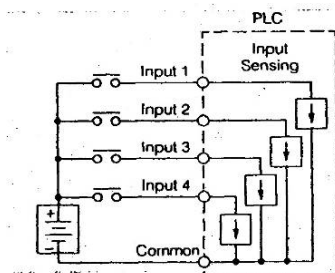
Pored jednostavnih mehaničkih prekidača, kao ulazni uređaji često se koriste i senzori. Postoji veliki broj različitih tipova senzora koji se koriste za detekciju prisustva predmeta, brojanje proizvoda, merenje temperature, pritiska, veličine i sl.

PLC kontroler može razmenjivati proste, digitalne, on/off signale i sa drugim pametnim uređajima, kao što su roboti, računari, sistemi za viziju i sl. Ovi signali se koriste za usklađivanje zajedničkog rada nezavisnih delova sistema. Na primer, robot može da pošalje signal PLC kontroleru u trenutku kada je završio neku operaciju. Po prijemu ovog signala PLC može da aktivira neki drugi uređaj koji nastavlja obradu proizvoda. Ovakav način koordinacije, kada uređaji jedan drugom daju dozvolu za obavljanje neke operacije, a sa ciljem da se postignu odgovarajuće performanse i obezbedi bezbednost rada, se zove *hendšejk (handshake)*. Komunikacija digitalnim (on/off) signalima se zove *primitivna komunikacija*.

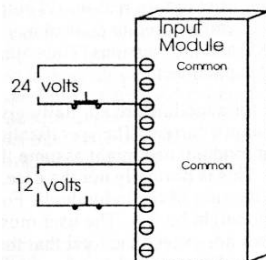
Ulazni moduli, najčešće, zahtevaju eksterno napajanje. Način povezivanja prikazan je na Sl. 3-3. U šemi veze uočava se izvor napajanja, ulazni uređaj (npr. prekidač), ulazni put (priključak preko koga u modul ulazi struja), ulazno kolo modula (ulazni deo opto-izolatora) i povratni put (priključak preko koga struja izlazi iz modula). Kada je prekidač zatvoren, strujno kolo je zatvoreno i struja teče; kada je prekidač otvoren, struja ne teče. Nažalost, ovakav način povezivanja zahteva dva priključka po ulazu. Iz tog razloga većina ulaznih modula poseduje grupu ulaza koji dele zajednički povratni put – COMMON (ili COM) (Sl. 3-4). Neki ulazni moduli imaju više od jedne grupe sa zajedničkim priključkom za napajanje (Sl. 3-5). Niz prekidača je povezan sa ulaznim modulom PLC kontrolera tako što je jedan kraj svakog prekidača povezan sa jednim od signalnih ulaza, dok su drugi krajevi svih prekidača međusobno spojeni i posredstvom eksternog izvora napajanja spojeni sa zajedničkim priključkom COM. Ako je potrebno, razdvojeni priključci za napajanje se mogu eksterno kratkospojiti (Sl. 3-6), čime se dobija konfiguracija kao na Sl. 3-4.



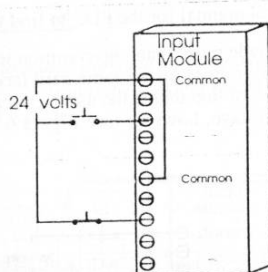
Sl. 3-3 Tipična sprega ulaznog uređaja i ulaznog PLC modula. (*Main path – ulazni put; Return path – povratni strujni put; Field Device – ulazni uređaj*)



Sl. 3-4 Ulazni PLC modul sa zajedničkim priključkom.

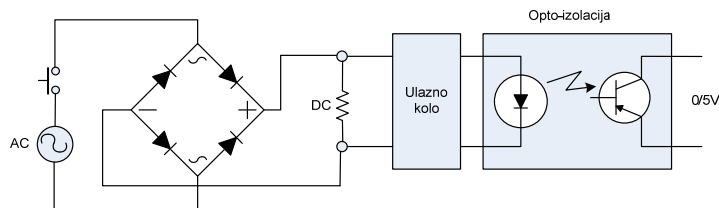


Sl. 3-5 Ulazni modul sa dva zajednička priključka.



Sl. 3-6 Spajanje zajedničkih priključaka.

Pored diskretnih ulaznih modula za jednosmerni, DC, napon, u upotrebi su i diskretni ulazni moduli za naizmenični, AC, napon, za slučaj kada se umesto jednosmernog koristi naizmenični, eksterni izvor napajanja. U ovom slučaju, na samim ulazima modula obavlja se AC-DC konverzija, npr. pomoću diodnog mosta, kao što je prikazano na slici. Na taj način, naizmenični ulazni napon se transformiše u jednosmerni i nakon eventualnog ispravljanja, nadalje obrađuje na identičan način kao kod DC ulaznih diskretnih modula (optoizolacija, filtriranje, ...). Napomenimo, da treba praviti razliku između AC diskretnih ulaznih modula i analognih ulaznih modula. Kod prvih, zadatak modula je da detektuje prisustvo/odsustvo ulaznog AC napona (što se konvertuje u binarnu vrednost 0 ili 1), dok je zadatak drugih konverzija ulaznog, promenljivog napona u numeričku (više-bitnu) vrednost.



Sl. 3-7 AC ulazni diskretni modul.

3.1.1 Povezivanje digitalnih senzora

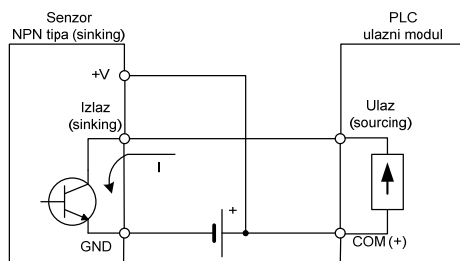
Digitalni senzori/prekidači su raspoloživi u dve varijante: sa normalno-zatvorenim i normalno-otvorenim izlaznim kontaktima. Pojam normalno se odnosi na stanje senzora/prekidača kada on ne detektuje prisustvo objekta, odnosno kada nije pritisnut. Pojmovi zatvoreno i otvoreno se odnose na stanje izlaza senzora/prekidača. Dakle, senzor/prekidač sa normalno-zatvorenim kontaktima je u stanju **on** za sve vreme dok senzor ne detektuje prisustvo objekta, odnosno dok prekidač nije pritisnut, a prelazi u stanje **off** kada senzor detektuje objekat, odnosno kada se prekidač pritisne. Za senzore/prekidače sa normalno-otvorenim kontaktima važi suprotno.

Način povezivanja senzora na ulazni modul PLC kontrolera se donekle razlikuje od načina povezivanja prekidača. Većina senzora poseduje ograničenje u pogledu maksimalne izlazne struje, tj. struje koju može da prihvati izlazni stepen senzora kada je senzor u stanju **on**. Obično, izlazna struja je ograničena na 100mA. To praktično znači da u izlaznom kolu senzora, kada je on priključen na PLC ili neki drugi uređaj mora postojati otpornost koja će ograničiti struju kroz senzor. Ako se senzor vezuje na PLC ovaj uslov je ispunjen jer sam ulaz PLC kontrolera poseduje dovoljno veliku ulaznu otpornost koja u većini slučajeva ograničava struju kroz senzor na prihvatljiv nivo. Ukoliko otpornost ulaza PLC kontrolera nije dovoljno velika, na red sa senzorom treba vezati eksterni otpornik.

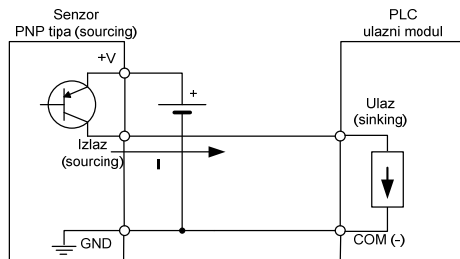
Senzori zahtevaju napajanje da bi obavljali svoju funkciju, čak i kada nisu aktivirani spoljnom pobudom. Struja koja je neophodna za rad senzora se zove *struja curenja*. S obzirom da se struja curenja protiče kroz ulazno kolo PLC modula, može se desiti da izazove aktiviranje ulaza PLC kontrolera. Struja curenja je mala i obično nije veća od 2mA i u većini slučajeva je dovoljno mala tako da ne dolazi do lažnog aktiviranja ulaza PLC kontrolera. Ako ipak postoji mogućnost da se to desi, paralelno sa senzorom treba vezati otpornik.

Pored senzora sa dva izlazna priključka, u upotrebi su i senzori sa tri priključka. Jedan od priključaka je predviđen za napajanje (+V ili GND) dok su druga dva izlazni kontakti senzora. Postoje dva tipa senzora sa tri priključka: NPN i PNP.

Senzora NPN tipa se može predstaviti kao na Sl. 3-8. Kao što se može videti, kod senzora NPN tipa, struja izlazi iz ulaznog priključka PLC modula (*sourcing* tipa) i ulazi u senzor kroz izlazni priključak senzora (zato se kaže i da je senzor *sinking* tipa). Sprega senzor PNP tipa sa PLC modulom prikazana je na Sl. 3-9. U ovom slučaju, struja izlazi iz izlaznog priključka senzora (zato se kaže i da je senzor *sourcing* tipa) i ulazi u ulazni priključak PLC modula, koji je *sinking* tipa.



Sl. 3-8 Povezivanje senzora NPN tipa na ulazni PLC modul.



Sl. 3-9 Povezivanje senzora PNP tipa na ulazni PLC modul.

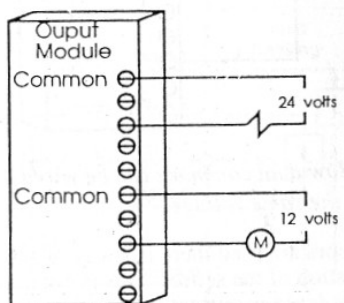
3.2 Diskretni izlazni moduli

Digitalni (ili diskretni) izlazni modul obezbeđuje spregu PLC kontrolera i izlaznih uređaja koji zahtevaju on/off upravljanje. Digitalni izlazni moduli funkcionišu kao prekidači. Izlazni uređaji mogu biti: starteri motora, svetiljke, relei, solenoidi i sl. Digitalni izlazni moduli, u zavisnosti od tipa, mogu generisati DC (jednosmeran) ili AC (naizmeničan) napon. Izlazni moduli su dostupni u konfiguracijama od po 8, 16 i 32 izlaza.

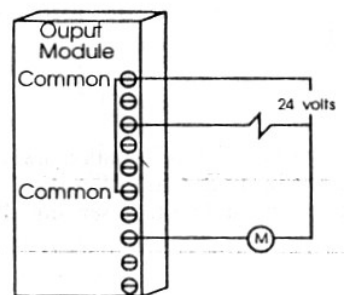
Važan parametar u specifikaciji izlaznog modula je maksimalna izlazna struja, koja se uobičajeno navodi kao maksimalna struja celokupnog modula i maksimalna struja po jednom izlazu, pri čemu je obično ukupna maksimalna struja manja od zbira maksimalnih struja pojedinačnih izlaza. Na primer, maksimalna struja pojedinačnog izlaza može biti ograničena na 1A, a maksimalna ukupna struja na 5A.

Svaki izlaz izlaznog modula je zaštićen posebnim osiguračem. Kod mnogih modula postoji svetlosna indikacija pregorelog osigurača.

Digitalni izlazni moduli obično imaju više od jednog priključka za masu. To omogućava korišćenje različitih naponskih nivoa na istom modulu (Sl. 3-10). Korisnik može krakospojiti ove priključke, ali je onda ograničen na korišćenje samo jednog eksternog izvora napajanja (Sl. 3-11).

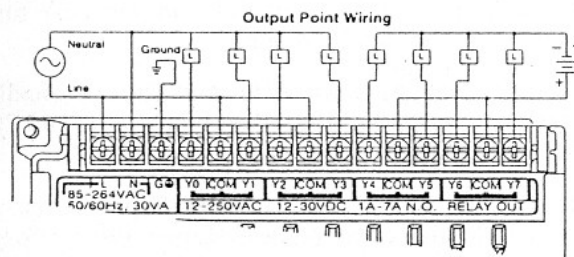


Sl. 3-10 Digitalni izlazni modul sa dva zajednička priključka.

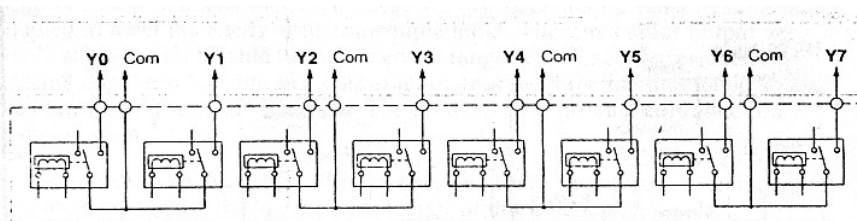


Sl. 3-11 Krakospojeni zajednički priključci.

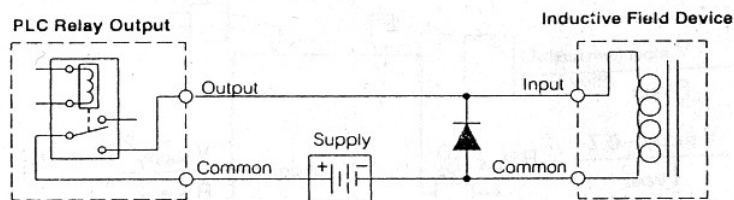
Izlazni stepen digitalnog izlaznog modula se realizuje pomoću: tranzistora, triaka ili relea. Tranzistorski izlazi se koriste za DC (jednosmerne) izlaze. Triaci se koriste za AC (naizmenične) izlaze, dok se releini izlazi mogu koristiti kako za DC tako i za AC izlaze. Šta više, kod modula sa releinim izlazima, pojedini izlazi mogu imati DC, a drugi AC pobudu (Sl. 3-12). Na (Sl. 3-13) je prikazana struktura releinog izlaznog stepena. U slučajevima kada se releini izlaz povezuje sa induktivnim opterećenjem, izlazi bi trebali biti zaštićeni diodama kako bi se produžio životni vek kontakta relea. Naime, prilikom uključivanja/isključivanja izlaza dolazi do pojave naponskih pikova na krajevima induktivnog opterećenja (kontra-elektromotorna sila) koji mogu izazvati varničenje kontakta izlaznog relea. Ugradnom diode, na način kao na (Sl. 3-14), ovi pikovi se eliminišu.



Sl. 3-12 Primena digitalnog izlaznog modula sa relejnim izlazima. (*L - opterećenje*)

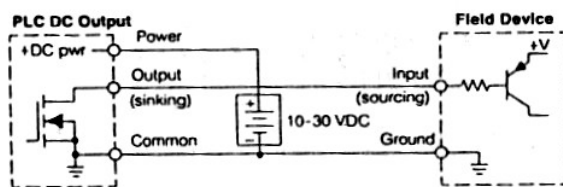


Sl. 3-13 Relejni izlazni stepen.

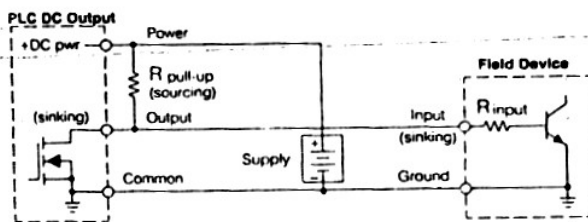


Sl. 3-14 Sprega induktivnog opterećenja na izlazni digitalni modul.

Na Sl. 3-15 i Sl. 3-16 prikazano je kako se tranzistoriski izlaz povezuje sa izlaznim uređajem čiji je ulaz PNP, odnosno NPN tipa. Kao što se vidi tranzistoriski izlaz je tipa otvoreni drejn, što znači da u slučaju pobude uređaja NPN tipa mora biti ugrađen eksterni *pull-up* otpornik kako bi se obezbedila struja pobude kada je izlazni tranzistor zakočen. Važno je pravilno dimenzionisati *pull-up* otpornik. Ako *pull-up* otpornik isuviše velike otpornosti, struja pobude može biti nedovoljna za pobudu ulaza priključenog uređaja. Sa druge strane, ako *pull-up* otpornik ima malu otpornost, disipacija na otporniku kada je izlazni tranzistor provodan biće velika.



Sl. 3-15 Sprega izlaznog PLC modula *sinking* (NPN) tipa i izlaznog uređaja *sourcing* (PNP) tipa.



Sl. 3-16 Sprega izlaznog PLC modula *sourcing* (PNP) tipa i izlaznog uređaja *sinking* (NPN) tipa.

3.3 SLC 500 diskretni U/I moduli

Seriya diskretnih ulaznih i izlaznih modula predviđenih za spregu sa PLC kontrolerima tipa SLC 500 nosi oznaku 1746 i sadrži 39 različitih jedinica. Moduli iz ove serije imaju 4, 8, 16 ili 32 ulaza/izlaza. Postoje varijante sa AC, DC ili TTL ulaznim električnim interfejsima. Moduli obezbeđuju ulazno filtriranje, optičku izolaciju. Dostupni su ulazni moduli sa ulazima *sourcing* ili *sinking* tipa. Postoje varijante izlaznih modula sa tranzistorskim AC, tranzistorskim DC i releinim izlazima. Takođe, ova serija diskretnih modula sadrži i kombinovane ulazno-izlazne module sa po 2-ulaza/2-izlaza, 4-ulaza/4-izlaza i 6-ulaza/6-izlaza. Na Sl. 3-17 dat je pregled diskretnih modula serije 1746.

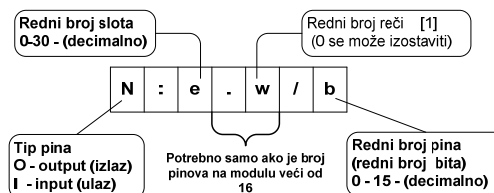
ID Code	Voltage Category	Cat. No.	Input/Output	I/O Points	Module Description
AC Modules					
100	100/120V ac	1746-IA4	Input	4	120V ac Input Module
300	100/120V ac	1746-IA8	Input	8	120V ac Input Module
500	100/120V ac	1746-IA16	Input	16	120V ac Input Module
101	200/240V ac	1746-IM4	Input	4	240V ac Input Module
301	200/240V ac	1746-IM8	Input	8	240V ac Input Module
501	200/240V ac	1746-IM16	Input	16	240V ac Input Module
2703	100/120V ac	1746-OA8	Output	8	120/240V ac Output Module
2903	100/120V ac	1746-OA16	Output	16	120/240V ac Output Module
2803	120/240V ac	1746-OAP12 ^①	Output	12	High Current 120/240V ac Output Module
DC Modules					
306	24V dc	1746-IB8	Input	8	Current Sinking DC Input Module
506	24V dc	1746-IB16	Input	16	Current Sinking DC Input Module
706	24V dc	1746-IB32 ^①	Input	32	Current Sinking DC Input Module
519	24V dc	1746-ITB16	Input	16	Fast Response DC Sinking Input Module
509	48V dc	1746-IC16	Input	16	Current Sinking DC Input Module
507	125V dc	1746-IH16	Input	16	Current Sinking DC Input Module
320	24V dc	1746-IV8	Input	8	Current Sourcing DC Input Module
520	24V dc	1746-IV16	Input	16	Current Sourcing DC Input Module
720	24V dc	1746-IV32 ^①	Input	32	Current Sourcing DC Input Module
518	24V dc	1746-ITV16	Input	16	Fast Response DC Sourcing Input Module
515	5V dc/TTL	1746-IG16 ^②	Input	16	Current Sourcing TTL Input Module
2713	24V dc	1746-OB8	Output	8	Current Sourcing DC Output Module
2913	24V dc	1746-OB16	Output	16	Current Sourcing DC Output Module
2920	24V dc	1746-OB16E ^{①③}	Output	16	Current Sourcing DC Output Module
3113	24V dc	1746-OB32 ^①	Output	32	Current Sourcing DC Output Module
2721	24V dc	1746-OBP8 ^③	Output	8	High Current Sourcing DC Output Module
2921	24V dc	1746-OBP16 ^①	Output	16	High Current Sourcing DC Output Module
2714	24V dc	1746-OV8	Output	8	Current Sinking DC Output Module
2914	24V dc	1746-OV16	Output	16	Current Sinking DC Output Module
3114	24V dc	1746-OV32 ^①	Output	32	Current Sinking DC Output Module
2922	24V dc	1746-OVP16 ^①	Output	16	High Current Sinking DC Output Module
2915	5V dc/TTL	1746-OG16 ^②	Output	16	Current Sinking TTL Output Module

ID Code	Voltage Category	Cat. No.	Input/Output	I/O Points	Module Description
AC/DC Modules					
510	24V ac/dc	1746-IN16	Input	16	24V ac/dc Input Module
2500	AC/DC Relay	1746-OW4 ⁽¹⁾	Output	4	Relay (Hard Contact) Output Module
2700	AC/DC Relay	1746-OW8 ⁽¹⁾	Output	8	Relay (Hard Contact) Output Module
2900	AC/DC Relay	1746-OW16 ⁽¹⁾	Output	16	Relay (Hard Contact) Output Module
2701	AC/DC Relay	1746-OX8 ⁽¹⁾	Output	8	Isolated Relay Output Module
800	Inputs – 120V ac Outputs – AC/DC Relay	1746-IO4 ⁽¹⁾	Input/Output	2 Inputs 2 Outputs	Combination Input/Output Module
1100	Inputs – 120V ac Outputs – AC/DC Relay	1746-IO8 ⁽¹⁾	Input/Output	4 Inputs 4 Outputs	Combination Input/Output Module
1500	Inputs – 120V ac Outputs – AC/DC Relay	1746-IO12 ⁽¹⁾	Input/Output	6 Inputs 6 Outputs	Combination Input/Output Module

Sl. 3-17 Diskretni ulazni i izlazni moduli iz serije 1746.

3.4 Sprezanje diskretnih U/I modula sa kontrolerom

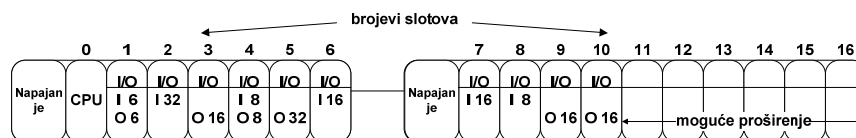
Vrednost binarnog signala koji dolazi na ulazni pin nekog U/I modula očitava se za vreme *ulaznog dela sken ciklusa*. U zavisnosti od toga da li očitana vrednost predstavlja *logičku nulu* ili *jedinicu* formira se odgovarajuća vrednost *bita* (0 ili 1) i upisuje na mesto u *datoteci 0 (Input image file)* koje odgovara adresi ulaznog pina. Isto tako, vrednost bita koji treba da se prenese na izlazni pin kontrolera kao binarni signal, nalazi se u *datoteci 1 (output image file)*. Za vreme *izlaznog dela sken ciklusa* ova vrednost se očitava, konvertuje u odgovarajući signal i prenosi na izlazni pin čija adresa odgovara mestu u datoteci na kome se nalazi posmatrani bit.



Sl. 3-18 Format adresa binarnih U/I signala.

Svakom modulu koji nema više od 16 pinova pridružuje se po jedna **16-bitna reč** u datoteci 0 odnosno 1. Ako modul ima manje od 16 pinova, onda se ne koriste svi bitovi u pridruženoj reči. Ako modul ima 32 pina, njemu se pridružuju dve susedne 16-bitne reči. Pri tome, koja reč će biti pridružena modulu zavisi od slotu u kome se modul nalazi. Format adresiranja prikazan je na Sl. 3-18.

Da bi se ilustrovao način povezivanja adresa modula sa odgovarajućim datotekama posmatraće se kontroler koji se sastoji iz jedne šasije sa 7 slotova koja je povezana sa šasijom od 10 slotova kao što je to ilustrovano na Sl. 3-19. Kao što se vidi prva šasija sadrži procesorski modul u slotu 0; kombinovani U/I modul sa 6 ulaza i 6 izlaza u slotu 1; ulazni modul sa 32 ulazna pina u slotu 2 itd. U drugoj šasiji koriste se samo prva četiri slotova, dok se preostali slotovine koriste. Struktura datoteka 0 i 1 koje odgovaraju datoj konfiguraciji prikazana je u tabelama T. 4 i T. 5.



Sl. 3-19 Konfiguracija kontrolera.

T. 4 Organizacija datoteke 0

Slot - U/I tip (pinovi)	Adresa reči	Adrese bitova	Data file 0 – Output image															
			bitovi															
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1 - out (0-5)	O:1	O:1/0 ... O:1/5	Ne koristi se															
3 - out (0-15)	O:3	O:3/0 ... O:3/15																
4 - out (0-7)	O:4	O:4/0 ... O:4/7	Ne koristi se															
5 - out (0-15)	O:5	O:5/0 ... O:5/15																
5 - out (16-32)	O:5.1	O:5.1/0 ... O:5.1/15																
9 - out (0-15)	O:9	O:9/0 ... O:9/15																
10 - out (0-15)	O:10	O:10/0 ... O:10/15																

T. 5 Organizacija datoteke 1

Slot - U/I tip (pinovi)	Adresa reči	Adrese bitova	Data file 1 – Input image															
			bitovi															
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1 - in (0-5)	I:1	I:1/0 ... I:1/5	Ne koristi se															
2 - in (0-15)	I:2	I:2/0 ... I:2/15																
2 - in (16-32)	I:2.1	I:2.1/0 ... I:2.1/15																
4 - in (0-7)	I:4	I:4/0 ... I:4/7	Ne koristi se															
6 - in (0-15)	I:6	I:6/0 ... I:6/15																
7 - in (0-15)	I:7	I:7/0 ... I:7/15																
8 - in (0-7)	I:8	I:8/0 ... I:8/7	Ne koristi se															

Potrebno je da se istakne da su *I* i *O* datoteke jedine dve datoteke kod kojih elementi nisu poredani u neprekidnom nizu i kod kojih su elementi promenljive dužine. U ovom primeru tako posle elementa O:1, kao susedna reč u memoriji nalazi se element O:3. Isto tako posle elementa O:5 koji zauzima dve reči dolazi element O:9 koji zauzima jednu reč. Ovo je, naravno, prirodna posledica činjenice da su redni brojevi elemenata vezani za slotove, a da je njihova dužina određena brojem pinova na modulu. Neophodno je, međutim da se to ima na umu prilikom korišćenja indeksnog adresiranja ili pri radu sa nizovima podataka. Naime, kao što je već istaknuto, kod datoteka ovog tipa indeksna adresa se određuje tako što se baznoj adresi dodaje sadržaj indeksnog registra, pri čemu je pomeraj izražen u *rečima*. To znači da redni broj reči na koju pokazuje indeksni registar ne mora biti, a najčešće i nije jednak rednom broju elementa datoteke.

4 Leder programiranje

Ako se PLC posmatra kao mikroprocesorski sistem, što on i jeste, onda bi se moglo očekivati da se za njegovo programiranje koriste standardni programski jezici. Međutim, ako se pođe od činjenice da je PLC projektovan kao namenski mikroprocesorski sistem za upravljanje i nadzor rada nekog procesa, i da u skladu sa tim ima poseban operativni sistem koji obezbeđuje periodično ponavljanje sken ciklusa, onda je logično očekivati da je za njegovo programiranje razvijen i poseban programski jezik. Kao što je već ranije istaknuto, PLC je početno razvijen sa idejom da zameni relejne sisteme. To znači da se očekivalo da on realizuje odgovarajuću vremensku sekvencu logičkih operacija. Pored toga, uspešna primena PLC-a u praksi, zahtevala je i da se njegovo programiranje prilagodi tehnici koja je svim korisnicima relejnih sistema dobro poznata. Iz svih ovih razloga, za projektovanje PLC-ova razvijen je programski jezik zasnovan na *leder (lestvičastim) dijagramima – leder programski jezik*.

4.1 Rang

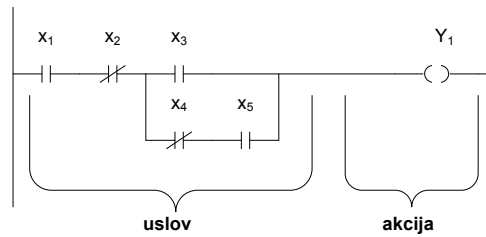
Jedna programska linija leder jezika sastoji se iz niza grafičkih simbola (programskih naredbi) koji predstavljaju različite logičke elemente i druge komponente kao što su tajmeri i brojači, koji su poredani duž horizontalne linije – *rang (rung)* – koja je na oba kraja spojena sa dvema vertikalnim linijama. Prema tome, leder dijagram ima izgled *lestvica*, odakle potiče i njegov naziv (*ladder – lestvice*).

Svaki rang leder dijagrama sastoji se iz *dva dela*. Na levoj strani ranga nalazi se *uslov* izražen u formi kontaktne (prekidačke) logike, dok se na desnoj strani ranga nalazi *akcija* koja treba da se izvrši ukoliko je *uslov ispunjen (true)* (Sl. 4-1).

- **Uslov** – Grafički simboli na levoj strani ranga odnose se ili na stanja signala koji predstavljaju fizičke ulaze PLC-a, i čije su vrednosti tokom ulaznog dela sken ciklusa smeštene u *input image file*, ili na stanja internih promenljivih, čije su vrednosti smeštene u odgovarajućim datotekama. Svaki simbol predstavlja jednu *unarnu binarnu operaciju* kojoj je pridružena odgovarajuća tablica istinitosti. Uz grafički simbol naznačava se i *adresa* promenljive koja predstavlja operand. Pri ispitivanju istinitosti uslova smatra se da se nad svim simbolima u jednoj liniji (*redna, serijska veza*) obavlja *logička "I" operacija*. To znači da je uslov istinit ukoliko je svaki pojedinačni iskaz istinit. Na levoj strani ranga dozvoljena su i *granjanja (paralelene veze)*. Pri ispitivanju istinitosti uslova paralelene veze se tretiraju kao *logička "ILI" operacija*. To znači da će iskaz predstavljen nizom paralelnih grana biti istinit, ako bar jedna od grana sadrži istinit iskaz. Potrebno je da se istakne da leva strana ranga može biti formirana i tako da na njoj nema ni jednog simbola. U tom slučaju smatra se da je uslov koji se na taj način definiše uvek istinit.
- **Akcija** – Grafički simboli na desnoj strani ranga odnose se ili na fizički izlaz (promenljive smeštene u *output image file*, koje će biti prenete na izlaze kontrolera u toku izlaznog dela sken ciklusa) ili na interne promenljive, čije su vrednosti smeštene u odgovarajućim datotekama. Svaki simbol predstavlja jednu *naredbu* koja se izvršava ako je uslov na desnoj strani istinit. Uz simbol se naznačava i *adresa* promenljive čija se vrednost menja prilikom izvršavanja naredbe, ili koja na bilo koji drugi način učestvuje u realizaciji naredbe (npr. otpočinjanje ili zaustavljanje neke aktivnosti, skok na neki drugi rang, poziv potprograma itd.). *Serijska veza* na desnoj strani ranga *nije* dozvoljena, dok *paralelna veza* označava da se više različitih naredbi izvršavaju kao rezultat ispitivnja istinitosti jednog istog uslova.

U literaturi je uobičajeno da se i simboli koji označavaju *uslov* i simboli koji označavaju *akciju* označavaju kao *naredbe*. Otuda je neophodno da se istakne suštinska razlika između *naredbi uslova* i *naredbi akcije*. Naime, izvršavanje *naredbi uslova* obavlja se tako što se u zavisnosti od vrednosti operanda, prema pridruženoj tablici istinitosti, naredbi *dodeljuje vrednost (0 ili 1)*. Dakle, *naredbe uslova* se izvršavaju u svakom sken ciklusa i rezultat njihovog izvođenja je *vrednost naredbe*. Za razliku od toga *naredbama akcije* se ili *dodeljuje vrednost nekoj promenljivoj* ili *izvršava neka druga*

aktivnost. Ove naredbe se izvršavaju samo ako je *uslov* koji im prethodi *istinit* (dodeljena mu je vrednost 1). Pri tome se samim *naredbama akcije ne dodeljuje nikakva vrednost*.



Sl. 4-1 Leder rang

Leder program se izvršava u toku programskog dela sken ciklusa i to tako što se obrađuje rang po rang u nizu kako su oni definisani. U svakom rangu ispituje se istinitost uslova i ukoliko je uslov istinit izvršavaju se odgovarajuće naredbe u desnom delu ranga. To znači da promenljive na desnom delu ranga mogu menjati svoju vrednost samo jedanput u toku sken ciklusa, i to upravo onda kada se odgovarajući rang ispituje. Potrebno je zapaziti, međutim, da ukoliko se promenljiva na desnoj strani ranga odnosi na fizički izlaz, vrednost izlaza neće biti promenjena u istom trenutku vremena. Naime, za vreme programskog skena menjaju se samo vrednosti promenljivih smeštenih u *slici izlaza*. Tek kasnije, za vreme izlaznog dela sken ciklusa, sve promenljive iz slike izlaza biće prenete na odgovarajuće izlazne linije. Ista stvar važi i za ulazne promenljive. Drugim rečima, za vreme programskog skena ispitivanje istinitosti uslova odnosi se na vrednosti promenljivih u *slici ulaza*, koje su tu upisane za vreme ulaznog dela sken ciklusa koji je prethodio programskom skenu, a ne na trenutne vrednosti promenljivih na ulaznim linijama. Naravno, svi uslovi i naredbe koji su vezani za interne promenljive izvršavaju se u trenutku skaniranja pojedinog ranga.

4.2 Bit naredbe

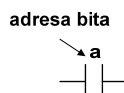
Bit naredbe su, kao što samo ime kaže naredbe čiji su operandi *bitovi*. Sa gledišta lokacije operanada, to znači da se oni najčešće nalaze u datoteci 3 (bit file), digitalnim ulaznim ili izlaznim datotekama (input image file 1 ili output image file 0) ili u korisničkim datotekama bit tipa. Pored toga, adresirani operand može da se nalazi i u bilo kojoj drugoj datoteci u okviru koje je moguće adresirati pojedini bit. Za vreme programskog skena u okviru bit naredbi ispituje se stanje pojedinog bita, ili se njegova vrednost postavlja na 1 (*set*) ili na 0 (*reset*).

4.2.1 Bit naredbe za definisanje uslova

Ove naredbe se postavljaju na levoj strani ranga i definišu uslov koji se odnosi na stanje bita čija je adresa definisana u naredbi. Kao rezultat izvođenja naredba dobija istinosnu vrednost *true* (*istinit*) ili *false* (*neistinit*).

- **XIC - Examine if closed (ispitivanje da li je kontakt zatvoren)**

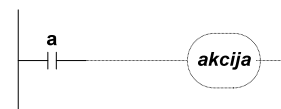
Grafički simbol



Tablica istinitosti

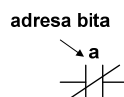
Stanje bita "a"	Vrednost XIC naredbe
0	False
1	True

Položaj u rangu



- **XIO - Examine if open (ispitivanje da li je kontakt otvoren)**

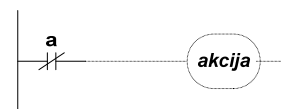
Grafički simbol



Tablica istinitosti

Stanje bita "a"	Vrednost XIC naredbe
0	True
1	False

Položaj u rangu

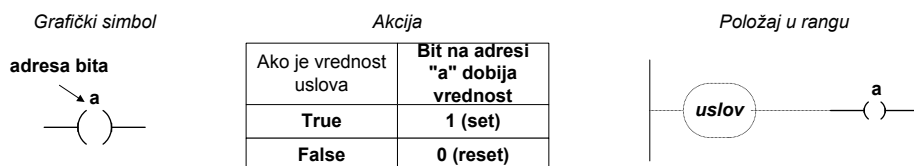


Nazivi ove dve naredbe potiču od ispitivanja binarnih signala koji dolaze sa prekidačkih kola. U tom smislu *XIC* naredba se odnosi na *normalno otvoren prekidač*, dok se *XIO* naredba odnosi na *normalno zatvoren prekidač*.

4.2.2 Bit naredbe za postavljanje vrednosti izlaza

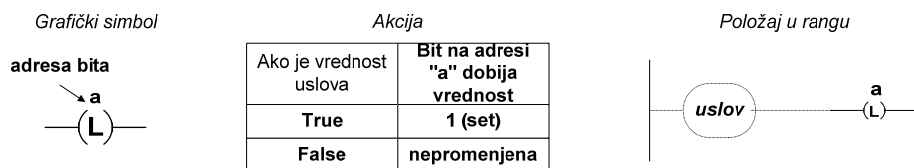
Ovim naredbama se bitu čija je adresa navedena u naredbi dodeljuje vrednost 1 ili 0. Podsetimo se da se ove naredbe nalaze na desnoj strani ranga, što znači da će se one izvršiti samo ako je iskaz (uslov) na levoj strani ranga istinit.

- **OTE - Output energize (pobuđivanje izlaza)**



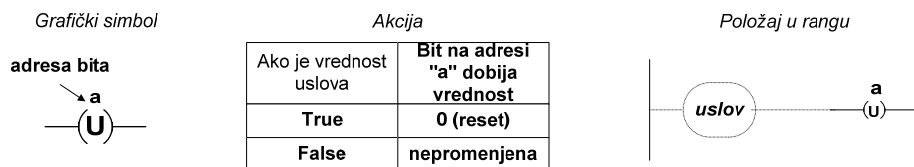
Potrebno je da se zapazi da se ovom naredbom vrednost bita čija je adresa "a" može promeniti samo jedanput za vreme sken ciklusa. Ova vrednost ostaće neizmenjena sve do sledećeg sken ciklusa, kada će se pri skeniranju odgovarajućeg ranga ponovo ispitati uslov i izvesti odgovarajuća akcija.

- **OTL - Output latch (pamćenje izlaza)**



OTL naredbom se adresirani bit može isključivo postaviti na 1. Naime za razliku od *OTE* naredbe kojom se vrednost bita može postavljati na 0 ili 1 svaki put kad se rang skenira, kod *OTL* naredbe vrednost bita se postavlja (lečuje) na 1 u prvom skenu u kome je *uslov* istinit. Nakon toga ova naredba postaje neosetljiva na istinosnu vrednost *uslova*. To znači da će vrednost bita ostati neizmenjena bez obzira na to kako se menja vrednost *uslova*.

- **OTU - Output unlatch (resetovanje izlaza)**

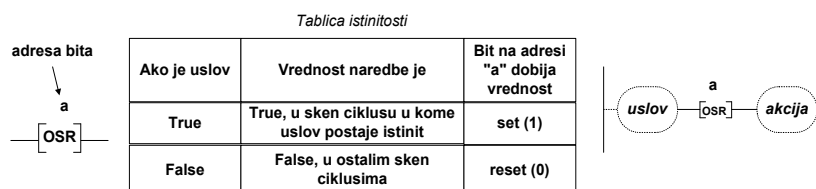


OTU naredbom se adresovani bit može isključivo postaviti na 0. Pri tome, vrednost bita se postavlja (lečuje) na 0 u prvom skenu u kome je *uslov* ispunjen. Nakon toga ova naredba postaje neosetljiva na vrednost *uslova*.

Potrebno je da se istakne da se *OTL* i *OUT* naredba koriste uvek u paru, pri čemu se u obe naredbe adresira isti bit.

4.2.3 Bit triger naredba

- **OSR - One-shot rising (uzlazna ivica)**



OSR naredba omogućava da se obezbedi izvođenje neke akcije *samo jedanput*. Potrebno je da se istakne da je ovo specifična naredba koja istovremeno pripada i kategoriji *uslova* i kategoriji *akcije*. Naime ova naredba se postavlja u rangu *između* dela koji predstavlja *uslov* i dela koji predstavlja *akciju*. Kada se u toku sken ciklusa detektuje da je uslov *promenio* svoju vrednost sa *neistinit* na *istinit* (uzlazna ivica) onda *OSR* naredba takođe dobija vrednost *istinit* (što ovu naredbu svrstava u kategoriju naredbi uslova). Istovremeno se i bitu čija je adresa pridružena toj naredbi dodeljuje vrednost 1 (po čemu se ova naredba svrstava i u kategoriju akcija). Obe ove vrednosti ostaju nepromenjene do sledećeg sken ciklusa, kada naredba dobija vrednost *neistinit*, dok se adresirani bit postavlja na vrednost 0 ili 1 u zavisnosti od vrednosti uslova. U narednim sken ciklusima vrednost naredbe ostaje nepromenjena sve dok se u *uslovu* (koji predstavlja ulaz u *OSR*) ponovo ne detektuje prelaz “neistinit/istinit”.

Potrebno je istaći da bit čija je adresa pridružena ovoj naredbi ne predstavlja vrednost naredbe. Naime, ovaj bit se koristi kao interna promenljiva i služi za pamćenje *vrednosti uslova* koji prethodi *OSR* naredbi. Vrednost ovog bita je 1 ako je *uslov istinit*, odnosno 0 ako je *uslov neistinit*. U tom smislu, sa aspekta dodeljivanja vrednosti bitu čija se adresa navodi u *OSR* naredbi, ova naredba je identična sa *OTE* naredbom. Navedeni bit se može nalaziti u bilo kojoj bit-adresibilnoj datoteci *izuzev* datoteke ulaza i izlaza.

Vrednost koju dobija *OSR* naredba koristi se kao *uslov* za izvođenje naredbe *akcije* koja se nalazi na desnoj strani ranga (neposredno iza *OSR* naredbe). Shodno tome, naredba *akcije* biće izvršavana *po jedanput* pri svakom prelazu *uslova* “neistinit/istinit”.

Iza *OSR* naredbe se može nalaziti samo jedna naredba akcije.

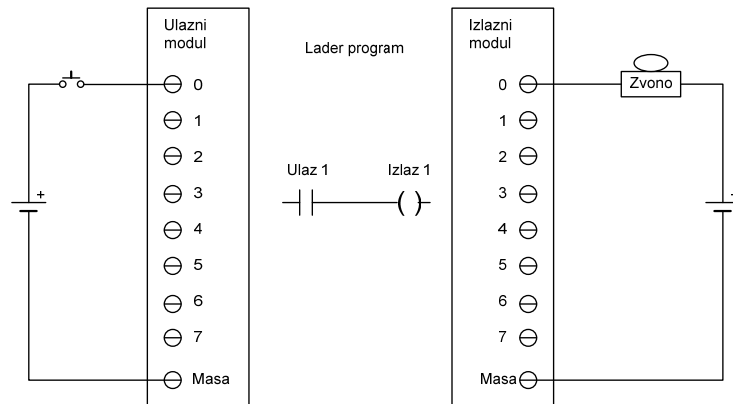
4.3 Kontakti

Većina ulaza u PLC su jednostavni uređaji koji mogu biti u stanju uključeno (*on*) ili u stanju isključeno (*off*). Ovakvi ulazi su prekidači i digitalni senzori koji detektuju uslove tipa: objekat je prisutan, puno/prazno i td. U leder programu stanje prekidača se ispituje naredbama *XIC*, za normalno otvorene prekidače/senzore, i *XOC*, za normalno zatvorene prekidače/senzore.



Sl. 4-2 Jednostavan leder dijagram.

Prekidač za zvono na ulaznim vratima je primer normalno otvorenog prekidača. Pritiskom na prekidač, zvono se spaja sa izvorom napajanja, struje počinje da teče i zvono zvonilo. (Ukoliko bi smo za ovu namenu koristili normalno zatvoren prekidač, zvono bi zvonilo za sve vreme dok je prekidač nepritisnut, a ne bi zvonilo samo dok je prekidač pritisnut, što je očigledno neželjeno ponašanje.) Odgovarajući leder dijagram prikazan je na Sl. 4-2. Program se sastoji iz samo jednog ranga, koji u delu uslova sadrži *XIC* naredbu, koja predstavlja prekidač, a u delu akcija *OTE* naredbu, koja predstavlja zvono. Ako je uslov tačan (prekidač pritisnut), akcija se izvršava (zvono se pobuđuje). Konceptualni prikaz PLC sistema za ovu namenu dat je na Sl. 4-3 (iako je sasvim jasno da za ovu namenu PLC predstavlja krajnje neracionalno rešenje). Prekidač je preko eksternog izvora napajanja priključen na ulaz 1, dok je zvono, takođe preko eksternog izvora napajanja, priključeno na izlaz 1 PLC kontrolera. Centralni deo slike prikazuje logiku po kojoj procesor određuje izlaz u zavisnosti od ulaza.



Sl. 4-3 Konceptualni pogled na PLC sistem.

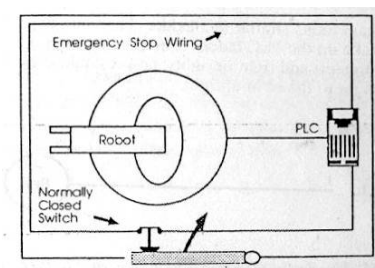
Zamislimo senzor koji treba da detektuje prisustvo metalnog predmeta na pokretnoj traci. I za ovu namenu, senzor sa normalno otvorenim kontaktima predstavlja logičan izbor – senzor se uključuje kada metalni predmet dođe ispred senzora; kada metalni prođe, senzor ponovo prelazi u isključeno stanje.

Normalno zatvoreni prekidači/senzori se koriste kada treba obezbediti veću sigurnost sistema. Ovakav prekidač je u stanju zatvoreno (propušta struju) za sve vreme dok nije pritisnut, dok se pritiskom na prekidač njegovi kontakti otvaraju (struja ne teče). Alarmni sistem je primer sistema gde je poželjno koristiti normalno zatvorene prekidače. Pretpostavimo da alarmni sistem treba da detektuje otvaranje ulaznih vrata. Ova jednostavna funkcija se može ostvariti pomoću normalno-otvorenog prekidača (slučno kao u primeru zvona na ulaznim vratima): kada se vrata otvore, prekidač se zatvara i alarm se uključuje. Međutim, rešenje sa normalno-otvorenim prekidačem ima jedan ozbiljan nedostatak. Pretpostavimo da se prekidač pokvario ili da se žica kojom je prekidač povezan sa PLC modulom prekinula. Očito, u tom slučaju, alarm se nikada neće uključiti, bez obzira da li su ulazna vrata otvorena ili ne. Drugim rečima, vlasnik kuće nije dobio informaciju da se sistem pokvario i zato nastavlja da koristi sistem kao da je sve u redu.

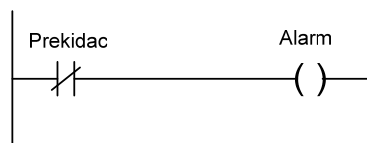
Ispravno rešenje je ono koje može da obezbedi aktiviranje alarma kada se vrata otvore, ali i onda kada sistem otkáže. Bolja varijanta je da se alarm aktivira zato što je sistem otkazao, iako nema provalnika, nego da je provala u toku, a alarm “čuti” zato što je prekidač pokvaren. Ovakvo ponašanje se može lako realizovati uz pomoć normalno-zatvorenog prekidača – otvaranje vrata i prekid žice (slučajan ili nameran) ima isti efekat: prekid strujnog kola.

Slična razmatranja imaju čak i veći značaj kada se radi o industrijskim primenama, gde otkaz neke mašine može uzrokovati veliku štetu ili povrede ljudi. Zato se prilikom projektovanja sistema i razvoja lider programa posebna pažnja posvećuje bezbednosti sa ciljem da u slučaju otkaza sistem bude postavljen u stanje koje će biti bezbedno za ljude i sam proces.

Razmotrimo sistem sa (Sl. 4-4). Slika predstavlja proizvodnu ćeliju u kojoj radi robot. Ćelija je ograđena ogradom sa jednim ulaznim vratima. Kao kontroler ćelije koristi se PLC. Da bi se osiguralo da niko ne može ući u ćeliju dok robot radi, iskorićen je sigurnosni prekidač. Ako neko uđe u ćeliju, PLC će detektovati da je prekidač otvoren i uključice alarm. Za ovu namenu treba koristiti normalno-zatvoren prekidač. Ako se žica koja povezuje prekidač sa PLC kontrolerom prekine, PLC će “misliti” da je neko ušao u ćeliju i aktiviraće alarm. Kaže se da je ovako projektovan sistem *bezbedan na otkaze*.



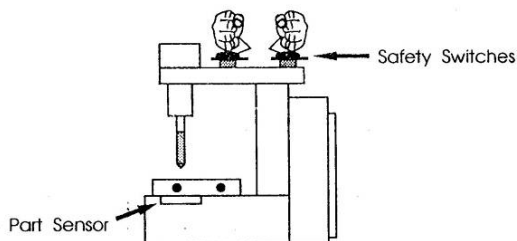
Sl. 4-4 Obezbeđenje proizvodne ćelije. (Normally Closed Switch – Normalno zatvoren prekidač)



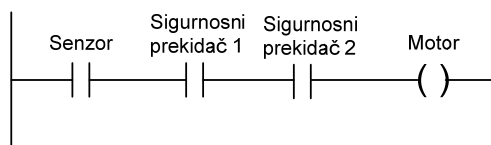
Sl. 4-5 Rang leder dijagrama sa normalno-zatvorenim prekidačem.

Na Sl. 4-5 je prikazan leder program za prethodno opisanu primenu. Leder program se sastoji iz samo jednog ranga koji u delu uslova sadrži XCI naredbu, koja predstavlja normalno-zatvoreni prekidač, a u delu akcija OTE naredbu, koja predstavlja alarm. Uslov će biti tačan i akcija će biti izvršena, ako je na odgovarajućem ulazu PLC modula prisutna 0, odnosno ako je prekidač otvoren, tj. pritisnut.

U istom rangu može se naći više od jednog prekidača. Na primer, zamislimo mašinu za bušenje rupa. Motor bušilice se uključuje pod uslovom da je predmet koji se buši prisutan i da je operater pritisnuo oba sigurnosna prekidača (Sl. 4-6). Na Sl. 4-7 je prikazan odgovarajući leder program. Program se sastoji iz samo jednog ranga koji u delu uslova sadrži serijsku vezu tri XIO naredbi od kojih prva odgovara senzoru za detekciju prisustva predmeta, dok druge dve odgovaraju sigurnosnim prekidačima. Seriska veza prekidača realizuje logički AND uslov (da bi se akcija obavila, svi prekidači moraju biti uključeni).

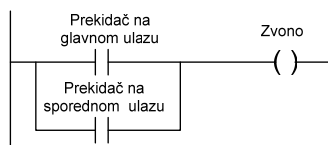


Sl. 4-6 Bušilica sa sigurnosnim prekidačima.



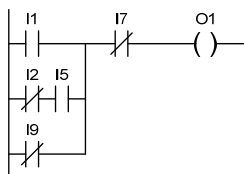
Sl. 4-7 Rang leder dijagrama sa serijskom vezom prekidača.

Često se javlja potreba da se izlaz aktivira ako je ispunjen barem jedan od više uslova. Zamislimo zgradu sa glavnim i sporednim ulazom. Na oba ulaza postoje prekidači za zvono. Pritisak na bilo koji od ova dva prekidača uključuje zvono. Na Sl. 4-8 je prikazan odgovarajući leder program. Leder program se sastoji iz samo jedan rang, koji u delu uslova sadrži grananje, tj. dve paralelne putanje (ili uslova) koje mogu uključiti zvono. Grananje predstavlja logičku OR operaciju nezavisnih uslova. Zvono zvući ako je pritisnut prekidač na glavnom ulazu ili prekidač na sporednom ulazu ili oba prekidača istovremeno.



Sl. 4-8 Rang leder dijagrama sa paralelnom vezom prekidača.

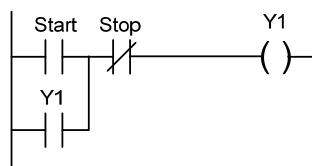
U opštem slučaju, rang leder dijagrama može sadržati proizvoljnu kombinaciju redno i paralelno vezanih prekidača (Sl. 4-9).



Sl. 4-9 Rang sa redno-paralelnom vezom prekidača.

4.4 Start/stop kolo

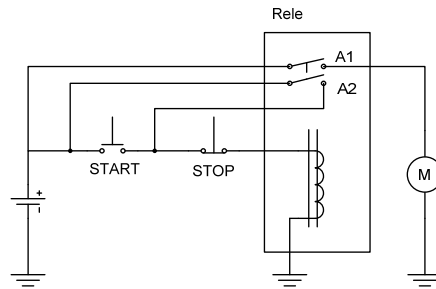
Start/stop kolo se veoma često koristi u industrijskim primenama. Na primer, mašina može imati start prekidač za početak rada i stop prekidač za zaustavljanje. Primer leder dijagrama start/stop kola prikazan je na Sl. 4-10. Start je normalno-otvoren, a Stop normalno-zatvoren prekidač. Pritiskom na prekidač Start, uslov ranga postaje tačan, a izlaza Y1 se aktivira. Uočimo da se izlaz Y1 koristi i kao ulaz. S obzirom da je sada $Y1=1$, uslov ostaje tačan, a izlaz aktivan i nakon otpuštanja prekidača Start. Pritiskom na prekidač Stop, uslov ranga postaje netačan, a izlaz se deaktivira. Pošto je sada $Y1=0$, uslov ostaje netačan, a izlaz neaktivan i nakon otpuštanja prekidača Stop. Da bi se izlaz ponovo aktivirao potrebno je ponovo pritisnuti taster Start. Opisani postupak formiranja start/stop kola se zove *samodržanje*, s obzirom da izlazna promenljiva zadržava vrednost i posle prestanka uslova za njeno aktiviranje.



Sl. 4-10 Start/stop kolo.

Ponašanje Start/stop kola je identično ponašanju SR leča, pri čemu prekidač Start ima ulogu setovanja, a Stop ulogu resetovanja leča.

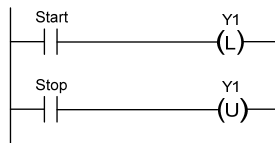
Osim u leder programima, Start/stop kolo se često realizuje i u relejnoj tehnici, uz pomoć relea sa dva normalno-otvorena kontakta. Na Sl. 2-4-11 je prikazana realizacija start/stop kola za upravljanje motorom. Jedan kontakt relea, A1, kontroliše napajanje motora, dok se drugi kontakt, A2, koristi za realizaciju start/stop kola. Pritiskom na prekidač Start, kroz namotaj relea počinje da teče struja, oba kontakta, A1 i A2 se zatvaraju i motor počinje da radi. Ovakvo stanje se zadržava i nakon otpuštanja prekidača Start, jer struja za namotaj relea nastavlja da teče kroz zatvoren kontakt A2. Pritiskom na prekidač Stop, struja kroz namotaje relea se prekida, kontakti A1 i A2 se otvaraju, motor prestaje da radi a kroz namotaje relea više ne protiče struja.



Sl. 2-4-11 Realizacija Start/Stop kola pomoću relea.

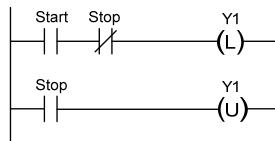
Na prvi pogled čini se da start/stop kolo predstavlja neracionalno rešenje, jer se ista funkcija (prosto aktiviranje/deaktiviranje izlaza) može ostvariti pomoću samo jednog, dvo-položajnog prekidača, koji bi direktno upravljao izlazom. Razlog za korišćenje start/stop kola je bezbednost. Pretpostavimo da u sistemu sa Sl. 2-4-11, motor radi (kroz namotaje relea teče struja) i da u jednom trenutku dođe do nestanka električne energije. Motor se zaustavlja, a kontakti relea otvaraju. Kada naknadno električna energija dođe, motor ostaje isključen, a da bi se ponovo uključio neophodno je pritisnuti prekidač Start. Ako bi smo za upravljanje motorom koristili samo jedan prekidač koji bi direktno kontrolisao napajanje motora, motor bi po dolasku električne energije nastavio da radi, zato što je prekidač ostao uključen, što može biti kritično sa stanovišta bezbednosti.

Start/stop kolo se može realizovati i uz pomoć naredbi OTL i OTU, kao što je prikazano na slici. Uočimo da su sada oba ulaza, Start i Stop, sa normalno-otvorenim kontaktima i da se obe naredbe OTL i OTU odnose na isti izlaz, Y1. Postavljanjem ulaza Start na 1, izvršava se naredba OTL, koja postavlja izlaz Y1 na 1 (Y1 se setuje), koji ostaje 1 i kada se Start vrati na 0. Da bi se izlaz Y1 postavio na 0 (tj. resetovao), potrebno je postaviti Stop=1, što aktivira naredbu OTU. Izlaz Y1 zadržava vrednost 0 i nakon postavljanja ulaza Stop na 0.



Sl. 4-12 Realizacija start/stop kola pomoću naredbi OTL i OTU.

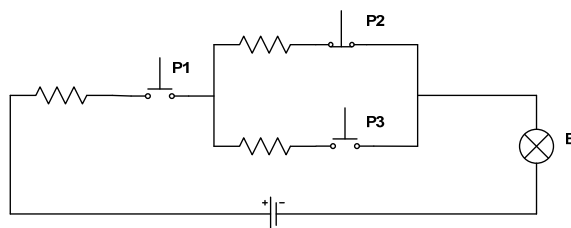
U rešenju sa Sl. 4-12, kritična situacija je ona kada su oba prekidača, Start i Stop, zatvorena (=1). Naime, ovakva situacija nesme da se javi na ulazu, jer pod tim uslovom kolo može početi da osciluje. Da bi se predupredilo ovakvo neželjeno ponašanje, leder program sa Sl. 4-12 može se proširiti XIC naredbom u prvom rangu koja će sprečiti da pri Start=1 uslov postane tačan ako je Stop=1 (Sl. 4-13). Očigledno, u rešenju sa Sl. 4-13 ulaz Stop ima viši prioritet, tako da se pri Start=Stop=1 kolo resetuje.



Sl. 4-13 Start/stop kolo koje rešava problem Start=Stop=1.

4.5 Primeri

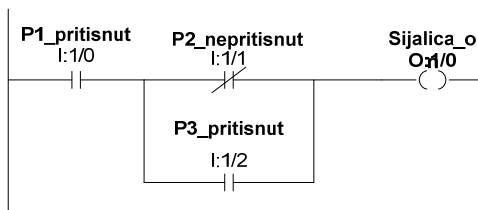
Primer 1. Napisati leder program za PLC koji zamenjuje električno kolo prikazano na Sl. 4-14, u kome se pomoću prekidača vrši paljenje i gašenje sijalice. Kao što se vidi, sijalica S će svetleti kada je zatvoren prekidač P_1 i jedan od prekidača P_2 ili P_3 .



Sl. 4-14 Elektročno kolo.

Rešenje: Za formiranje programa neophodno je raspolagati informacijom o vrsti prekidača kao i načinu njihovog vezivanja za U/I modul. Neka su P_1 , P_2 i P_3 tasteri, pri čemu su P_1 i P_3 u normalnom stanju otvoreni, a P_2 je u normalnom stanju zatvoren. To znači da će pritisak na P_1 ili P_3 dati digitalni signal koji predstavlja logičku jedinicu, dok će pritisak na P_2 dati logičku nulu. Pretpostavimo da su prekidači P_1 , P_2 i P_3 kao i sijalica S prikačeni na kombinovani digitalni U/I modul koji je smešten u slot 1 PLC-a. Pretpostavimo nadalje da se linije sa prekidačkih kola dovode na pinove 0, 1 i 2 ulaznog dela modula, dok je kolo u kome se nalazi sijalica vezano za pin 0 izlaznog dela modula. Shodno tome, adrese prekidača P_1 , P_2 i P_3 su respektivno I:1/0, I:1/1 i I:1/2, dok je adresa sijalice O:1/0.

U cilju formiranja levog dela ranga treba uočiti da je *uslov* za paljenje sijalice da se istovremeno pritisne taster P_1 i jedan od tastera P_2 ili P_3 . Budući da su tasteri P_1 i P_3 normalno otvoreni, pritisak na njih dovodi do zatvaranja odgovarajućih prekidačkih kola, tako da se može detektovati pomoću *XIC* naredbe, koja će dobiti vrednost *istinit* kada su vrednosti odgovarajućih bitova u slici ulaza postavljene na 1. Pritisak na taster P_2 koji je normalno zatvoren, dovodi do otvaranja njegovog prekidačkog kola, što znači da se može detektovati pomoću *XIO* naredbe, koja će dobiti vrednost *istinit* kada je vrednost odgovarajućeg bita u slici ulaza postavljena na 0. Konačno, kako se nad tasterima 2 i 3 zahteva logička *ILI* operacija, to odgovarajuće naredbe moraju biti vezane paralelno. Ispunjenost *uslova* treba da obezbedi da se na izlaznom pinu generiše naponski signal koji će da prouzrokuje paljenje sijalice. Ovaj zahtev se može ostvariti *OTE* naredbom. U skladu sa time odgovarajući rang leder programa ima izgled kao na Sl. 4-15.



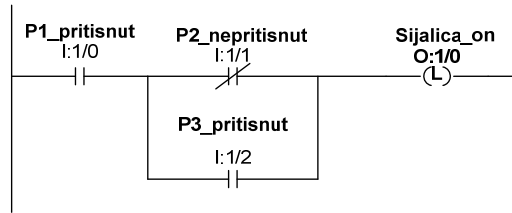
Sl. 4-15 Rang leder programa koji realizuje električno kolo.

Svakom bitu koji se koristi u leder programu može se pridružiti simboličko ime. U principu nema nikakvih posebnih pravila u pogledu davanja imena. Ipak praksa je pokazala da je pogodno da se ime formira tako da što vernije opisuje fizičko značenje signala na koji se odnosi. Pored toga, u cilju lakše provere ispravnosti programa, pogodno je da se ime formira tako da odgovara stanju pri kome bit ima vrednost 1. Poštujući taj princip, u ovom primeru je bitu I:1/0 dato ime *p1_pritisnut*, dok je bitu I:1/1 dato ime *p2_nepritisnut*. U skladu sa time prva *XIC* naredba koja ispituje vrednost bita na adresi I:1/0 dobiće vrednost *istinit* ako bit ima vrednost 1 što znači da taster **P_1 jeste pritisnut**. Isto tako *XIO* naredba koja ispituje vrednost bita na adresi I:1/1 dobiće vrednost *istinit* ukoliko bit nema vrednost 1, što znači da taster P_2 **nije nepritisnut**.

Opisani sistem će ispravno raditi samo dotle dok se odgovarajući tasteri drže pritisnuti. Naime, čim se taster otpusti on se vraća u normalni položaj i u sledećem sken ciklusu, uslov više neće biti ispunjen,

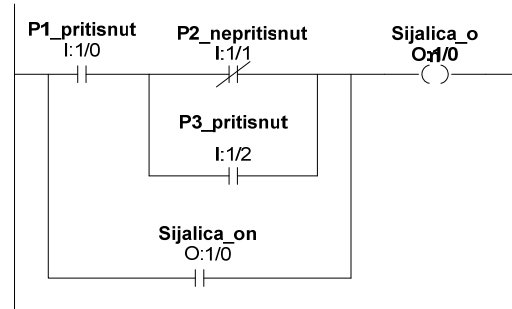
pa će se sijalica ugaziti. Prirodno je međutim da se zahtev postavi tako da sijalica nastavi da svetli i posle otpuštanja tastera.

Postavljeni zadatak može se rešiti tako što će se na neki način upamtiti da je uslov za paljenje sijalice u nekom trenutku bio ispunjen. U tu svrhu može se na izlaznom delu ranga umesto *OTE* naredbe postaviti *OTL* naredba koja će obezbediti trajno postavljanje (lečovanje) izlaza (Sl. 4-16).



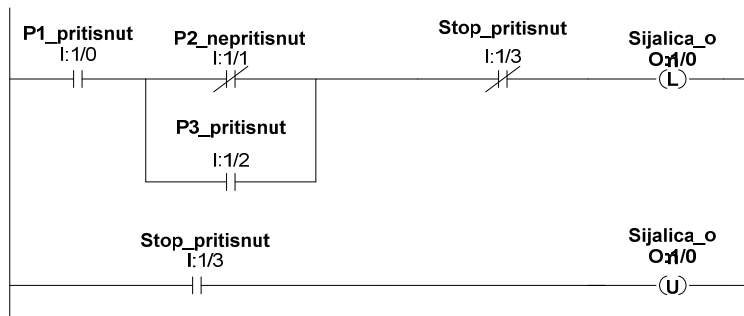
Sl. 4-16 – Trajno postavljanje izlaznog bita

Umesto korišćenjem naredbe *OTL* isti efekat se može postići i postupkom samodržanja. Naime, ako se u delu ranga koji predstavlja *uslov* doda još jedna paralelna grana sa *XIC* naredbom u kojoj se ispituje upravo bit koji se postavlja kao izlaz tog ranga (Sl. 4-17) onda će, čim se pritiskanjem tastera *uslov* prvi put ispuni, odgovarajući bit biti postavljen na 1, što znači da će u sledećim sken ciklusima naredba u paralelnoj grani stalno imati vrednost *istiniti*, pa se vrednost izlaznog bita neće menjati sa promenom stanja tastera.



Sl. 4-17 – Postupak samodržanja

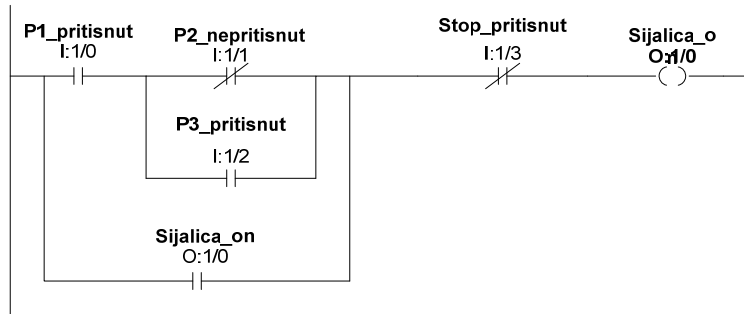
Potrebno je istaći da su poslednja dva primera formirana tako da će sijalica kad se jedanput upali nastaviti da svetli neograničeno dugo.



Sl. 4-18 – Paljenje i gašenje sijalice pomoću naredbi za lečovanje

Da bi se omogućilo i gašenje sijalice neophodno je da se sistemu doda još jedan taster (*Stop*). Pritiskom na ovaj taster, jedanput upaljena sijalica, bi bila isključena. Ako se pretpostavi da je ovaj taster normalno otvoren i da je vezan na pin 3 istog ulaznog modula, onda se postavljeni zadatak može realizovati na način koji je prikazan na slikama Sl. 4-18 i Sl. 4-19. Ovde je na red sa delom ranga kojim se ostvaruje paljenje sijalice, vezan uslov kojim se proverava da li *Stop* taster *nije pritisnut*. Sve dok *Stop* taster nije pritisnut, vrednost tog dela uslova je *istinita*, dakle on ne utiče na ponašanje sijalice. Kad se *Stop* taster pritisne, vrednost tog dela uslova postaje *neistinita*, a budući da je to redni

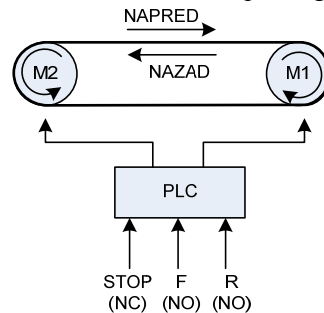
(serijski) uslov, i vrednost celog ranga postaje *neistinita*, pa se aktivira se naredba *unleth* (Sl. 4-18), odnosno izlazni bti se *OTE naredbom* postavlja na 0 (Sl. 4-19).



Sl. 4-19 – Paljenje i gašenje sijalice pomoću postupka samodržanja

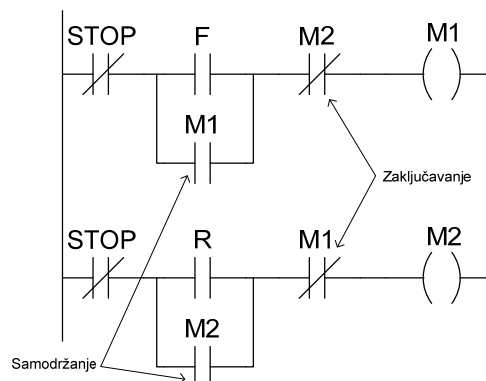
Naravno, ceo problem oko paljenja i gašenja sijalice bio bi rešen rangom koji je prikazan na Sl. 4-15, da su umesto tastera korišćeni dvopoložajni prekidači.

Primer 2. Sl. 4-20 prikazuje pokretnu traku (dvosmernu). Traka se pokreće pomoću dva motora, M1 i M2. Kada je M1 uključen, a M2 isključen, traka se kreće u smeru "napred"; kada je M1 isključen, a M2 uključen, traka se kreće u smeru "nazad", a kada su oba motora isključena, traka se ne kreće. Traka se pokreće u željenom smeru pritiskom na jedan od tastera: F, za napred, i R za nazad, oba sa normalno otvorenim kontaktima (NO). Takođe, postoji i taster STOP (sa normalno zatvorenim kontaktima) koji služi za momentalno zaustavljanje trake. Smer kretanja trake se može promeniti samo ako traka stoji. To znači da se traka mora zaustaviti pre nego što se pokrene u suprotnom smeru.



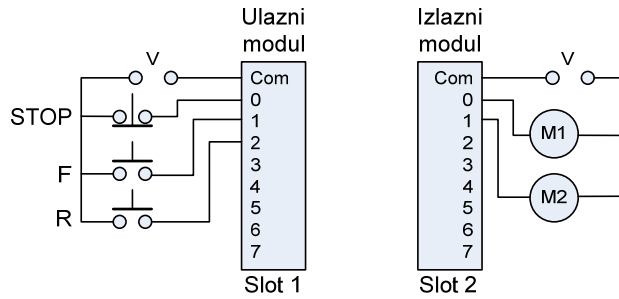
Sl. 4-20 Sistem pokretne trake.

Leder dijagram koji reguliše rad opisanog sistema prikazan je na Sl. 4-21. Radi se o tzv. "napred-nazad" kolu. Postoje dva ranga, od kojih svaki upravlja jednim izlazom (motorom). U okviru svakog ranga realizovano je samodržanje (da bi traka nastavila da se kreće i nakon otpuštanja tastera F ili R), i *zaključavanje* (onemogućava da se aktivira izlaz iz datog ranga, ako je izlaz onog drugog ranga aktivan - onemogućava naglu promenu smeru kretanja trake). Takođe, pritiskom na taster STOP njegovi kontakti se otvaraju, što "raskida" oba ranga i isključuje oba motora.



Sl. 4-21 Leder dijagram "napred-nazad" kola.

Na Sl. 4-22 je prikazana jedna moguća šema veze, koja pokazuje spregu tastera i motora sa ulaznim i izlaznim modulom PLC sistema. Ulazni modul je smešten u slot 1, a izlazni u slot 2. Takođe, sa slike vidimo na koje pinove su povezani tasteri STOP, F, i R, odnosno motori M1 i M2.

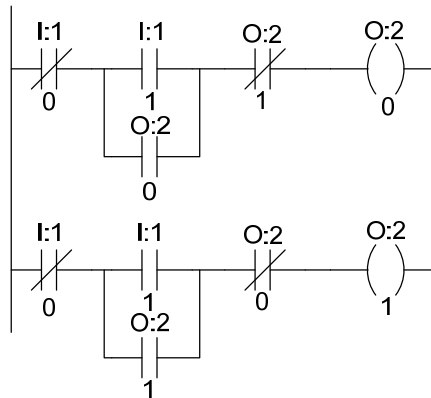


Sl. 4-22 Šema veze

Shodno šemi veze u sledećoj tabeli su navedene njihove odgovarajuće adrese:

S ignal	Ulaz/ Izlaz	A dresa
S TOP	Ulaz	I: 1/0
F	Ulaz	I: 1/1
R	Ulaz	I: 1/2
M 1	Izlaz	O: 2/0
M 2	Izlaz	O: 2/1

Na osnovu poznatih adresa ulaza i izlaza u mogućnosti smo da nacrtamo leder dijagram u konačnom obliku, sa navedenim fizičkim adresama umesto simboličkih imena ulaza i izlaza (Sl. 4-23).

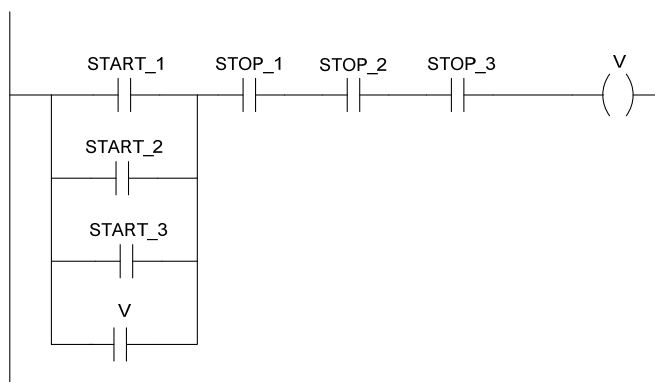


Sl. 4-23 Leder dijagram napred-nazad kola sa fizičkim adresama ulaza i izlaza.

Primer 3. Višestruko start-stop kolo.

Sistem za ventilaciju u nekoj hal se može uključivati/isključivati sa tri različita mesta. Na svakom mestu postoji start i stop taster. Start tasteri su sa normalno otvorenim (NO), a stop sa normalno zatvorenim (NC) kontaktima. Kreirati leder dijagram.

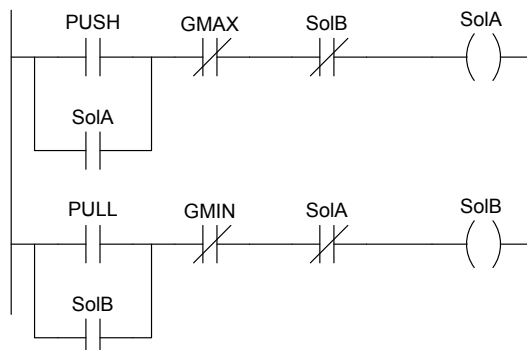
Ulazi u PLC su tri para tastera: START_i, STOP_i, i=1, 2 i 3. Izlaz iz PLC-a je signal V koji upravlja sistemom za ventilaciju. Leder dijagram je prikazan na Sl. 4-24. Uočimo paralelnu vezu start i rednu vezu stop tastera. Stop tasteri su sa NC tipa i u nepritisnutom stanju su zatvoreni, tj. daju logičku vrednost 1. Sa druge strane, kontakti start tastera su NO tipa i nepritisnutom stanju su otvoreni, tj. daju logičku vrednost 0. To znači da pritisak na bilo koji start taster pobuđuje izlaz V, što ima za posledicu aktiviranje samodržanja. Pritisak na bilo koji stop taster "raskida" uslov ranga i isključuje izlaz V.



Sl. 4-24 Leder dijagram za primer 3.

Primer 4. Hidrauličnim cilindrom se upravlja pomoću dva jednosmerna solenoida, A i B. Solenoid A otvara ventil za dovod komprimovanog vazduha kojim se cilindar izvlači, dok solenoid B otvara ventil za "dovod" vakuuma kojim se cilindar uvlači. Postoje dva granična prekidača, GMIN i GMAX, za indicaciju krajnjeg uvučenog (GMIN) i krajnjeg izvučenog položaja cilindra. Izvlačenje cilindra se startuje tasterom PUSH, a uvlačenje tasterom PULL (oba NO tipa). Cilindar se uvek izvlači do krajnje izvučene (GMAX) pozicije kada se ukida dovod komprimovanog vazduha. Slično, cilindar se uvek uvlači do krajnje uvučene pozicije (GMIN), kada se ukida "dovod" vakuuma. Dakle, ulazi u PLC sistem su: tasteri PUSH i PULL i granični prekidači GMAX i GMIN, dok su izlazi: SolA (za pobudu solenoida A) i SolB (za pobudu solenoida B). Realizovati leder dijagram. Obezbediti da oba ventila nikada ne budu istovremeno otvorena.

Leder dijagram je prikazan na Sl. 4-25. Dva ranga realizuju start-stop kolo za pobudu solenoida A i B. Start signal u prvom rangu je PUSH, a stop GMAX. U drugom rangu, start signal je PULL, a stop GMIN. Ispitivanje izlaznog bita SolB u prvom i izlaznog bita SolA u drugom rangu obezbeđuje "zaključavanje" odgovarajućeg ranga za vreme dok traje pobuda izlaza u onom drugom rangu.



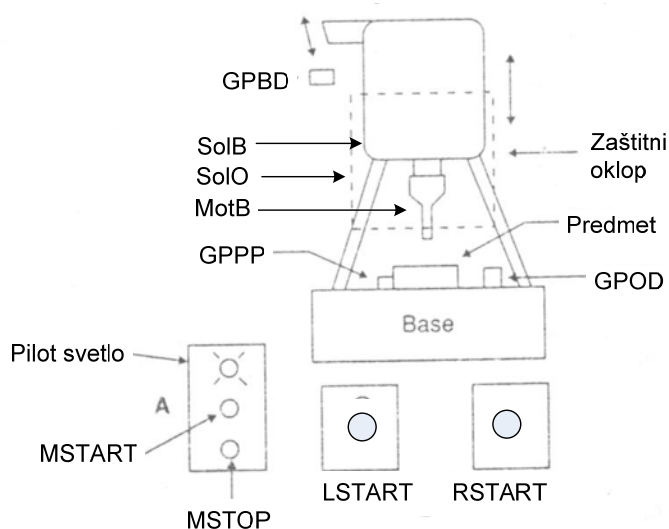
Sl. 4-25 Leder dijagram za primer 4.

Primer 5. Primer industrijskog procesa - poluautomatska presa za bušenje (Sl. 4-26).

U ovom primeru razmotrićemo postupak kreiranja leder programa za jedan tipični industrijski proces. Postupak čini nekoliko koraka. Prvo, neophodno je razjasniti i razumeti način rada, odnosno tok odvijanja procesa, kao i ulogu svih senzora i aktuatora, pomoću kojih se proces nadgleda i preko kojih se procesom upravlja. Nakon toga, u drugom koraku, potrebno je definisati sekvencu operacija koje čine jedan ciklus rada procesa. Tipično, proces se startuje, obavi jedan ciklus rada, a zatim se automatski resetuje i vraća u početni položaj, spreman za novo startovanje. U trećem koraku, treba identifikovati sve ulaze i izlaze, tj. signale koji se povezuju na ulazne i izlazne PLC module, a potiču od priključenih senzora i aktuatora. Takođe, treba definisati fizičke adrese ulaza i izlaza. Konačno, u četvrtom koraku, kreira se leder dijagram.

Korak 1 (Opis sistema je tipično obuhvaćen tekstom zadatka i pratećom slikom). Poluautomatska presa za bušenje prikazana je na Sl. 4-26. Presu čine: (a) pokretna bušilica, (b) zaštitni, pokretni oklop i (c) podloga za postavljanje predmeta koji se buši. Bušilica poseduje svoj motor koji okreće burgiju

(signal MotB), a spušta se naniže pomoću vazdušnog, pneumatskog cilindra. Ventil za dovod/ispuštanje komprimovanog vazduha u/iz pneumatskog cilindra se otvara/zatvara pomoću jednosmernog solenoida (signal SolB). Pojednostavljeno, aktiviranjem signala SolB, (SolB=ON), bušilica se spušta naniže; deaktiviranjem signala SolB (SolB=OFF), bušilica se pod dejstvom opruge, vraća u prvobitni, gornji položaj. Slično važi i za zaštitni oklop, koji takođe ima svoj pneumatski cilindar za spuštanje, kojim se takođe upravlja jednosmernim solenoidom (signal SolO; SolO=ON - oklop se spušta; SolO =OFF - oklop se podiže pod dejstvom opruge u polazni, gornji položaj). U sastavu prese su i tri granična prekidača: GPOD - za indicaciju krajnjeg donjeg položaja zaštitnog oklopa; GPPP - za indicaciju da je na podlogu za bušenje postavljen predmet i GPBD - za indicaciju da je burgija probila predmet.



Sl. 4-26 Poluautomatska presa za bušenje.

Presom se upravlja preko komandnog pulta koji čine četiri tastera i jedna, tzv. *pilot* svetiljka (PS). MSTART i MSTOP su *master* tasteri, pomoću kojih se celokupan sistem uključuje/isključuje. Pritiskom na taster MSTART sistem se uključuje, a svetiljka PS pali; pritiskom na taster MSTOP sistem se isključuje, a svetiljka PS gasi i trenutno prekida pobudu svih aktuatora. Pojedinačni ciklusi bušenja se startuju istovremenim pritiskom na dva tastera, LSTART i RSTART. Dejstvo ovih tastera je omogućeno samo dok je sistem uključen (PS svetli), a na mestu za bušenje je prisutan predmet za bušenje (aktivan je granični prekidač GPPP).

Radi postizanja veće bezbednosti u radu, predviđeno je se pre uključivanja bušilice najpre spusti zaštitni oklop, uz zahtev da za sve vreme spuštanja zaštitnog oklopa oba tastera, LSTART i RSTART moraju biti pritisnuti. Kada se oklop, pod dejstvom pneumatskog cilindra spusti u krajnji donji položaj (aktivan je granični prekidač GPOD), bušilica se automatski uključuje, a tasteri LSTART i RSTART mogu da se otpuste. Ciklus bušenja se završava kada se aktivira granični prekidač GPBD, što treba da isključi pobudu solenoida, a bušilica i zaštitni oklop se, pod dejstvom opruga, vraćaju u polazni, gornji položaj.

Korak 2. Pošto smo razjasnili način rada sistema, potrebno je kreirati listu operacija koje čine jedan ciklus rada. Ova lista pomoćiće nam da kreiramo leder dijagram. U konkretnom primeru, sled operacija je sledeći:

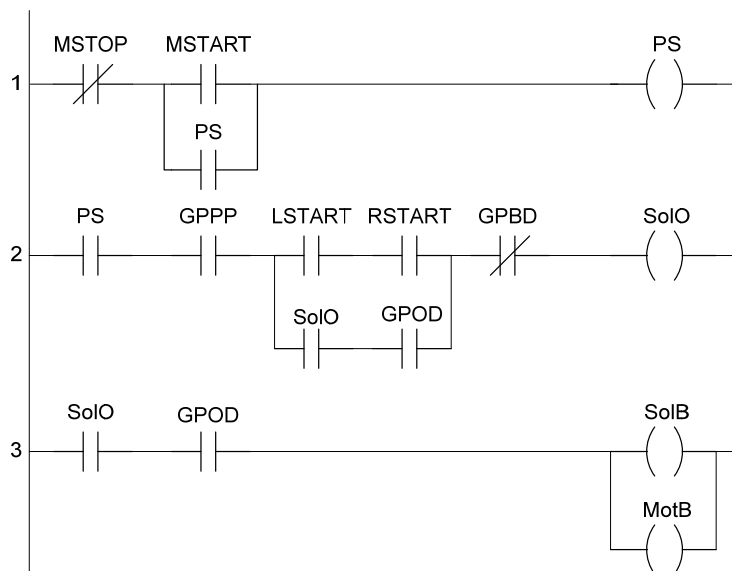
1. Postaviti predmeta na mesto za bušenje, tako da aktivira GPPP.
2. Pritisnuti istovremeno tastere LSTART i RSTART.
3. Zaštitni oklop se spušta i kada stigne u donji položaj aktivira granični prekidača GPOD.
4. Bušilica počinje da se okreće i spušta.
5. Burgija probija predmet i aktivira granični prekidač GPBD.
6. Aktuatori se isključuju, a bušilica i zaštitni oklop se podižu u gornji položaj.

Korak 3. U ovom koraku, treba identifikovati sve ulaze i izlaze sistema i definisati njihove fizičke adrese. Ispod je data tabela ulaza/izlaza za konkretan primer:

Opis	Naziv	Ulaz/Izlaz	Fizička adresa
Master start taster	MSTART	Ulaz	I:1/0
Master stop taster	MSTOP	Ulaz	I:1/1
Levi start taster bušilice	LSTART	Ulaz	I:1/2
Desni start taster bušilice	RSTART	Ulaz	I:1/3
Granični prekidač donjeg položaja oklopa	GPOD	Ulaz	I:1/4
Granični prekidač za predmet na poziciji bušenja	GPPP	Ulaz	I:1/5
Granični prekidač donjeg položaja bušilice	GPBD	Ulaz	I:1/6
Pilot svetlo	PS	Izlaz	O:2/0
Motor bušilice	MotB	Izlaz	O:2/1
Solenoid pneumatskog cilindra bušilice	SolB	Izlaz	O:2/2
Solenoid pneumatskog cilindra zaštitnog oklopa	SolO	Izlaz	O:2/3

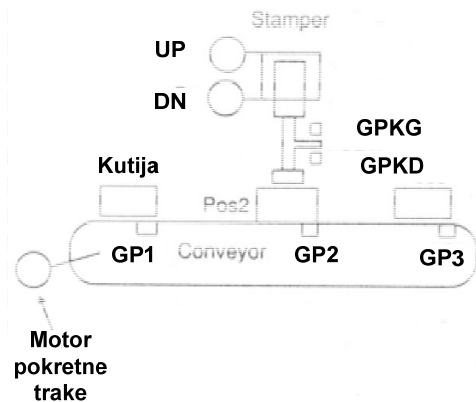
Korak 4. U ovom trenutku trebalo bi da imamo jasnu i preciznu predstavu o načinu rada sistema i o svim ulazim i izlazima i njihovoj ulozi, što je preduslov za uspešno kreiranje leder dijagrama.

Leder dijagram za upravljanje poluautomatskom bušilicom prikazan je na Sl. 4-27. Rang 1 upravlja pilot svetlom (PS), koje se pali tasterom MSTART, a gasi tasterom MSTOP. U ovom rangu je realizovano samodržanje, tako da svetlo ostaje upaljeno i nakon otpuštanja tastera MSTART. Rang 2 upavlja solenoidom zaštitnog oklopa, a rang 3 solenoidom i motorom bušilice. Uočimo da isključene pilot svetla, pomoću tastera MSTOP, takođe isključuje sve aktuatorne, ako su uključeni, zato što se bit PS nalazi u uslovu ranga 2, a izlazni bit SolO u uslovu ranga 3 (PS=OFF, garantuje SolO=SolB=MotB=OFF). Solenoid oklopa (SolO) se aktivira pod uslovom da je na podlogu za bušenje postavljen predmet (GPPP=ON) i da su pritisnuta oba tastera LSTART i RSTART. U trenutku kada se oklop spusti u krajnji donji položaj (GPOD=ON) aktivira se samodržanje u rang 2, koje obezbeđuje da će SolO ostati aktivan i ako se tasteri LSTART i RSTART otpuste (oklop je spušten i radnik može slobodno da skloni ruke sa tastera). Takođe, u trenutku kada se oklop spusti do kraja (GPOD=ON) aktiviraju se motor i solenoid bušilice (rang 3). Kada bušilica probije predmet, aktiviraće se granični prekidač GPBD=ON, što "raskida" uslov ranga 2 i isključuje pobudu solenoida zaštitnog oklopa (SolO=OFF). SolO=OFF u rang 3 isključuje motor i solenoid bušilice (SolB=MotB=OFF). Bez pobude solenoida, zaštitni oklop i bušilica se pod dejstvom opruga vraćaju u prvobitni, gornji položaj. Time je jedan ciklus bušenja završen, i sistem je spreman za novi ciklus.



Sl. 4-27 Leder dijagram poluautomatske bušilice.

Primer 6. Kreirati leder dijagram za PLC koji upravlja sistemom sa Sl. 4-28 (mašina za pečatiranje kutija). Postavljanjem kutije na pokretnu traku, na poziciju 1 (aktivan je granični prekidač GP1), traka se automatski pokreće (uključuje se motor, M=ON) i prenosi kutiju do pozicije 2. Kada kutija stigne na poziciju 2 (granični prekidač GP2 je aktivan), traka se zaustavlja (M=OFF), da bi se kutija pečatirala. Za pečatiranje se koristi poseban mehanizam kojim se upravlja pomoću dva signala UP i DN. Za DN=ON, klip na čijem kraju je pečat se spušta, a za UP=ON podiže. Za indicaciju graničnih pozicija klipa se koriste dva granična prekidača: GPKG (klip je u krajnjem gornjem položaju) i GPKD (klip je u krajnjem donjem položaju). Dakle, kada kutija stigne na poziciju 2, aktivira se signal DN sve dok se klip ne spusti do nivoa prekidača GPKN. Zatim se DN isključuje, a uključuje UP, da bi se klip vratio u gornji položaj, do nivoa prekidača GPKG. Nakon obavljenog pečatiranja, traka se ponovo pokreće, kako bi se kutija prenela do pozicije 3 (granični prekidač GP3). U tom momentu, traka se ponovo zaustavlja, a kutija se manuelno uklanja sa trake.



Sl. 4-28 Mašina za pečatiranje kutija.

Definišimo najpresevcu operacija u toku jednog ciklusa rada sistema:

1. Postaviti kutiju na poziciju 1, tako da aktivira granični prekidač GP1.
2. Motor se uključuje i traka se kreće sve dok kutija ne stigne na poziciju 2 i aktivira GP2.
3. Motor se isključuje, a signala DN aktivira što spušta klip do pozicije graničnog prekidača GPKD.
4. Pobuda DN se isključuje, a UP uključuje što podiže klip do nivoa graničnog prekidača GPKG.
5. Pobuda UP se isključuje, a motor pokretne trake uključuje, čime se kutija prenosi do pozicije 3 (GP3).
6. Motor se isključuje
7. (Pretpostavka je da će pečatirana kutija biti manuelno sklonjena sa pozicije 3, pre nego što se na poziciju 1 postavi nova - korak 1).

Zadatak ćemo rešiti tako što ćemo najpre, za svaki izlazni signal (M, DN i UP), definisati uslove pod kojima se signal aktivira i deaktivira, a zatim ćemo izvedene uslove iskoristiti za kreiranje uslova rangova kojima se signal setuje (naredbom OTL), odnosno resetuje (naredbom OTU).

Motor (signal M) se uključuje pod sledećim uslovima:

- kutija je postavljena na poziciju 1: GP1, ili
- završeno pečatiranje - kutija je na poziciji 2, pobuda UP je uključena i klip se upravo vratio u gornji položaj: GP2·UP·GPKG

Dakle, objedinjeni logički uslov za setovanje signala M glasi:

$$GP1 + GP2 \cdot UP \cdot GPKG$$

Motor se isključuje pod sledećim uslovima:

- kutija je stigla na poziciju 2 - kutija je na poziciji 2, pobuda UP je isključena i klip je u gornjem položaju: $GP2 \cdot \overline{UP} \cdot GPKG$ ili
- kutija je na poziciji 3: GP3

Objedinjeni uslov za isključenje motora glasi:

$$GP2 \cdot \overline{UP} \cdot GPKG + GP3$$

Signal za spuštanje klipa (DN) se setuje pod uslovom:

- kutija je na poziciji 2, klip je u gornjem položaju i signal za podizanje kutije isključen:
 $GP2 \cdot GPKG \cdot \overline{UP}$

Signal za spuštanje klipa (DN) se resetuje pod uslovom:

- klip je stigao u krajnji donji položaj: GPKD

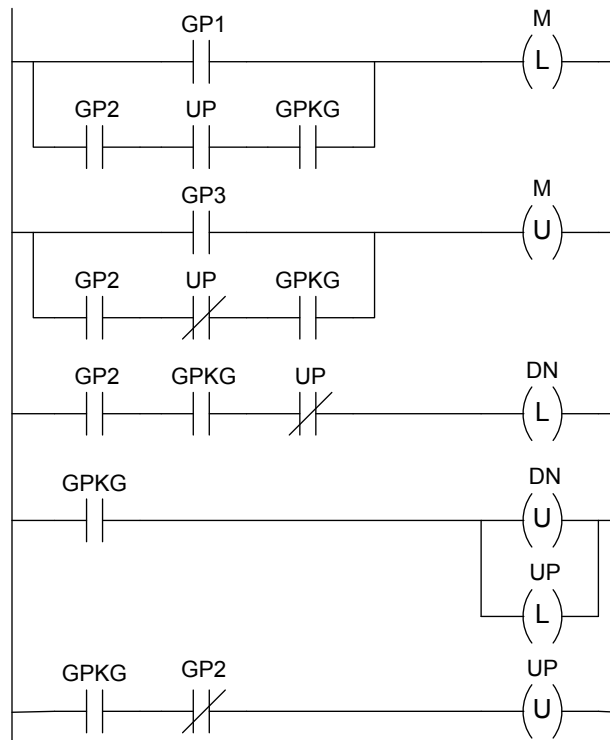
Signal za podizanje klipa (UP) se setuje pod uslovom:

- klip je stigao u krajnji donji položaj: GPKD

Signal za podizanje klipa se resetuje pod uslovom:

- klip je u krajnjem gornjem položaju i kutija je napustila poziciju 2: $GPKG \cdot \overline{GP2}$

Sa izvedenim uslovima za setovanje i resetovanje izlaznih signala, kreiranje leder dijagram je trivijalan zadatak (Sl. 4-29).



Sl. 4-29 Leder dijagram za problem 6.

Primer 7. Relizovati leder dijagram kontrolera motora (Sl. 4-30) prema sledećim zahtevima:

Ulazi u PLC su:

SW0 - glavni on/off prekidač. (Kada je OFF, svi izlazi su OFF).

SW1 - start taster motora. Normalno-otvoren.

SW2 - stop taster motora. Normalno-zatvoren.

SW3 - senzor temperature. SW3=ON - temperatura motora je niža od granične;
SW3=OFF - temperatura motora je viša od granične.

Izlazi iz PLC-a su:

G - zelena svetiljka. (G = ON, svetiljaka svetli)

R - crvena svetiljka. (R = ON, svetiljaka svetli)

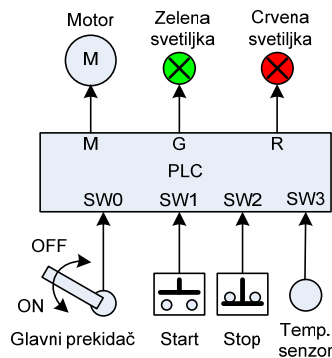
M - pobuda motora. (M = ON, motor radi)

Načina rada:

Kada je temperatura motora normalna, zelena svetiljaka svetli, čak iako je motor zaustavljen.

Kada se motor pregreje treba zaustaviti motor i upaliti crvenu svetiljku, koja ostaje upaljena sve dok se motor ne ohladi.

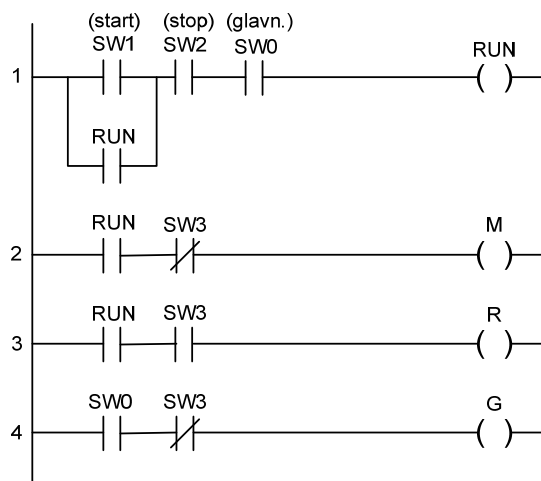
Kada se ohladi, motor se automatski pokreće, osim ako u međuvremenu nije pritisnut stop taster (SW2).



Sl. 4-30 Kontroler motora za problem 7

Rešenje:

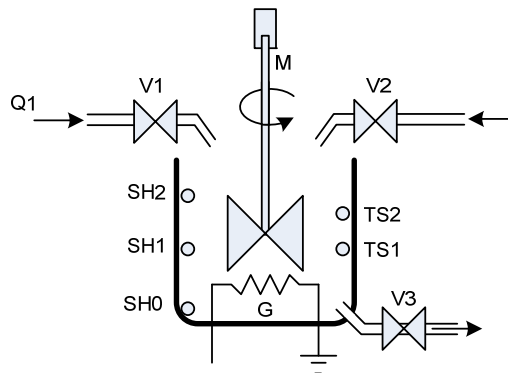
Leder dijagram je prikazan na Sl. 1-1Sl. 4-31. Uveden je pomoćni bit RUN, kojim se postavlja start-stop kolom iz ranga 1. RUN predstavlja dozvolu rada sistema. RUN se setuje start tasterom, pod uslovom da je glavni prekidač u položaju ON, a resetuje pritiskom na taster stop ili prebacivanjem glavnog prekidača u položaj OFF. Rang 2 definiše uslov pod kojim je aktivan motor, rang 3 uslov pod kojim je upaljena crvena, a rang 4 uslove pod kojim je upaljena zelena sijalica.



Sl. 4-31 Leder dijagram za problem 7

Primer 8. Kreirati leder program za upravljanje hemijskim procesom prema sledećim zahtevima. Proces počinje pritiskom na taster START, i odvija se u sledećih pet koraka:

- Nalivanje tečnosti Q1 do nivoa H1.
- Grejanje do temperature T1.
- Nalivanje tečnosti Q2 do nivoa H2.
- Mešanje i hlađenje do temperature T2.
- Ispuštanje smeše.



Sl. 4-32

Ulazi i izlazi sistema su:

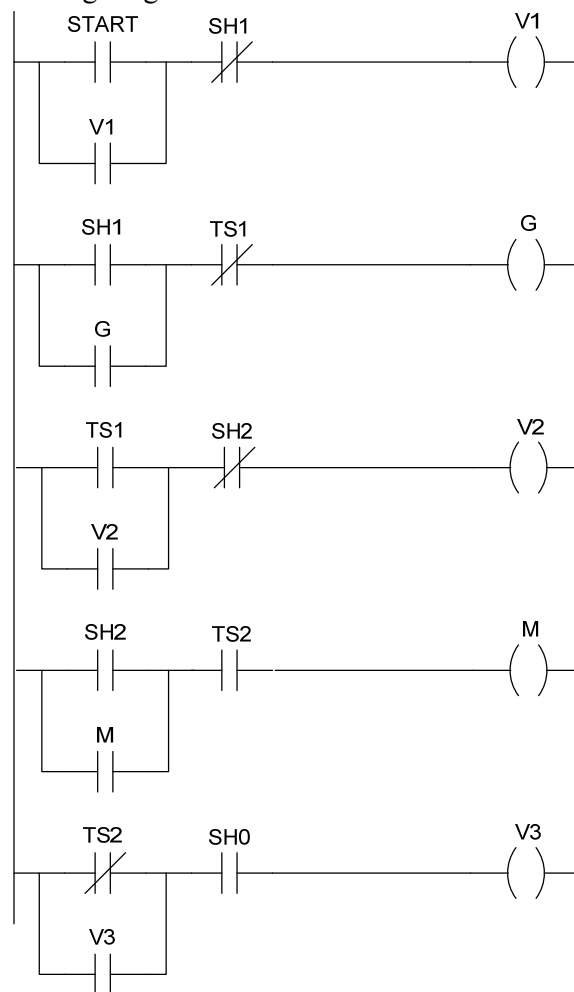
ST ART	Start prekidač	
SH 0	Senzor nivoa tečnosti postavljen na dno rezervoara	
SH 1	Senzor nivoa tečnosti postavljen u rezervoaru na visini H1.	ON - senzor prekriven tečnošću OFF senzor nije prekriven tečnošću
SH 2	Senzor nivoa tečnosti postavljen u rezervoaru na visini H2.	
V1	Ventil za dovod tečnosti Q1.	
V2	Ventil za dovod tečnosti Q2.	ON – ventil otvoren OFF – zatvoren
V3	Ispusni ventil rezervoara.	
TS 1	Temperaturski senzor podešen na temp. T1	ON - temperatura više od T1 (T2) OFF - temperatura niža od T1 (T2)
TS 2	Temperaturski senzor podešen na temp. T2	
M	Motor mešalice	

G	Grejač	
---	--------	--

Rešenje:

Proces čini niz koraka (sekvenca operacija), a u svakom koraku aktivan je jedan od aktuatora (izlaznih signala). Prvi korak se inicira pritiskom na taster START, dok se svaki sledeći korak inicira uslovom koji završava prethodni korak. Na primer, korak 1 (punjenje do nivoa H1) završava se kada tečnost dostigne nivo H1. Identična uslov (nivo tečnosti je H1) predstavlja znak za početak drugog koraka (grejanje do temperature T1). Slično, uslov koji prekida korak 2 (temperatura je dostigla T2 °C) je u isto vreme uslov koji započinje (startuje) korak 3 (nalivanje tečnosti do nivoa H2) i td.

Leder dijagram je prikazan na Sl. 4-33. Kao što vidimo, u dijagramu sadrži 4 ranga, po jedan za svaki izlazni signal. Svaki rang realizuje "start/stop" kolo, gde je uslov za "start" identičan uslovu za "stop" iz prethodnog ranga.



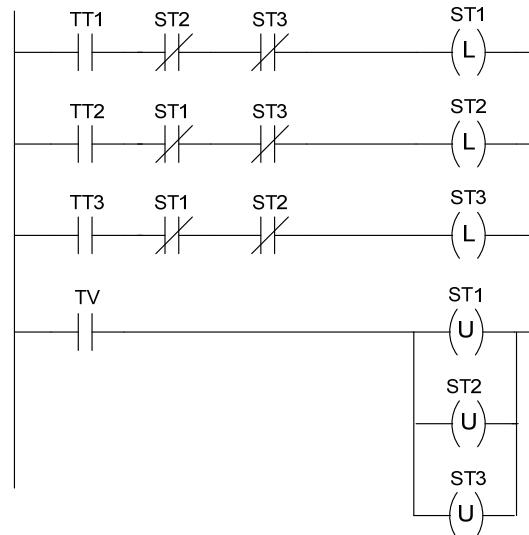
Sl. 4-33 Leder dijagram za problem 8

Primer 9. U TV studiju nalaze se tri pulta za tri učesnika u kvizu. Na svakom pultu je postavljen jedan taster i jedna sijalica. Takmičarima se postavlja pitanje, a onaj ko prvi pritisne svoj taster dobija priliku da da odgovor. Taster koji je prvi pritisnut pali sijalicu tog takmičara koja ostaje upaljena sve dok voditelj kviza ne pritisne taster za resetovanje. Sijalice

preostala dva takmičara ostaju ugašene bez obzira da li su oni pritisnuli svoje tastere ili ne. Realizovati odgovarajući leder dijagram.

Rešenje:

Ulazi u PLC sistem su: tastere takmičara, TT1, TT2 i TT3, i taster voditelja TV. Izlazi PLC sistema su tri sijalice takmičara: ST1, ST2 i ST3.



Sl. 4-34 Leder dijagram za problem 9

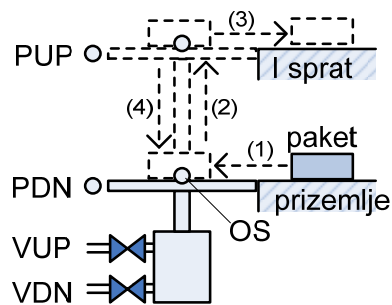
Primer 10. Realizovati leder dijagram upravljačke jedinice lifta koji se koristi za prenos paketa sa prizemlja na prvi sprat (Sl. 4-35).

Lift se podiže/spušta pomoću hidrauličnog cilindra. Da bi se lift podigao potrebno je otvoriti ventil VUP (VUP=ON), a da bi se spustio ventil VDN (VDN=ON). Sve dok su oba ventila zatvorena, lift miruje na zatečenoj poziciji.

Pozicija lifta se detektuje pomoću dva granična prekidača PUP i PDN. Gornji granični prekidač se zatvara (PUP=ON) kada lift stigne na prvi sprat, a donji (PDN=ON) kada se lift spusti u prizemlje.

Prisustvo paketa na platformi lifta se detektuje pomoću optičkog senzora OS. Senzor prelazi u stanje ON kada se na platformu stavi paket, a vraća u stanje OFF kada se sa platforme skloni paket.

Inicijalno, lift je u prizemlju. Kada se na platformu lifta postavi paket (korak (1)), lift se podiže na prvi sprat (korak (2)) i ostaje na toj poziciji sve dok se paket ne skloni s platforme (korak (3)), a zatim se spušta u prizemlje (korak (4)).



Sl. 4-35 Upravljačka jedinica lifta

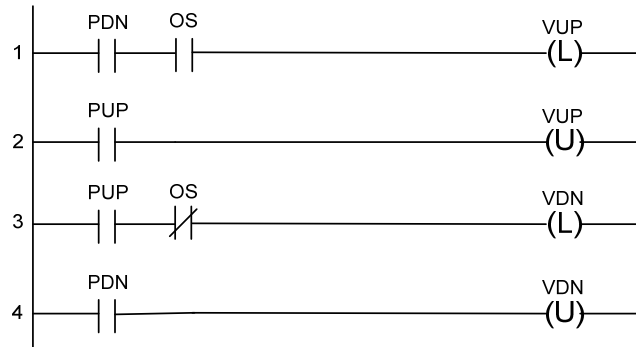
Rešenje:

Rang 1: Ventil VUP se otvara kada je platforma je u prizemlju (PDN=ON) i paket stavljen na platformu (OS=ON)

Rang 2: Ventil VUP se zatvara kada platforma stigne na prvi sprat (PUP=ON)

Rang 3: Ventil VDN se otvara kada je platforma na prvom spratu (PUP=ON) i paket se skloni s platforme (OS=OFF)

Rang 4: Ventil VDN se zatvara kada je platforma u prizemlju (PDN=ON).



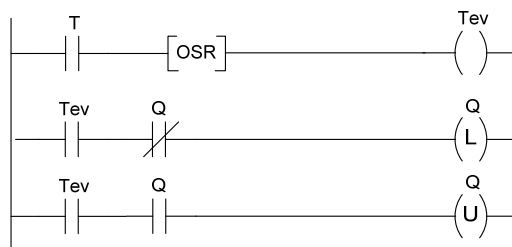
Sl. 4-36 Leder dijagram za problem 10

Primer 11. Realizovati funkciju T flip-flopa u leder jeziku. Ulaz je T, a izlaz Q. Svaka promena ulaza T sa 0 na 1 menja (komplementira) stanje izlaza Q. (Primena OSR naredbe).

Rešenje:

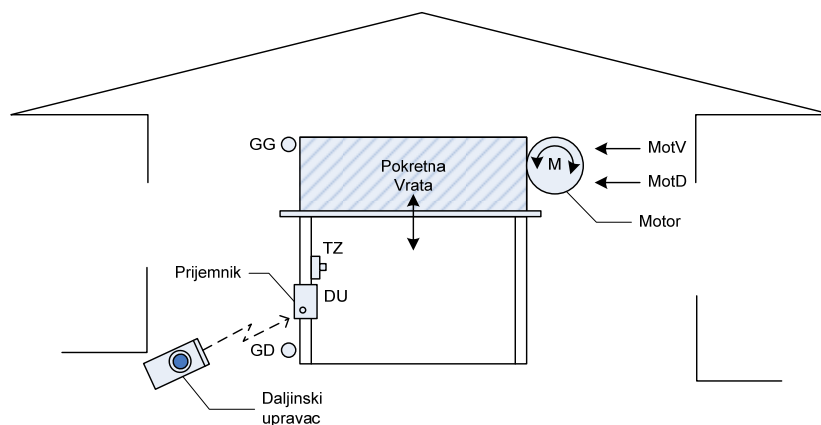
Tev je pomoćni bit, iz interne memorije PLC kontrolera, koji je jednak 1 samo u onim sken ciklusima kada se T menja sa 0 na 1. U svim ostalim sken ciklusima, Tev je 0.

Polazeći od činjenice da se leder program izvršava rang-po-rang, u redosledu kako su rangovi poređani u dijagramu, može se učiniti da rešenje sa Sl. 4-37 nije ispravno. Pretpostavimo da je $Q=0$ i da je T upravo postalo 1. U prvom rangu, postavlja se $Tev=1$; u drugom, zbog $Tev=1$ i $Q=0$, Q se setuje i postaje $Q=1$ i u trećem rangu, zbog $Tev=1$ i sada $Q=1$, Q se resetuje i vraća na $Q=0$! Međutim, ovakvo zaključivanje nije dobro iz razloga što su efekti svih akcija učinjenih na bitovima u toku jednog programskog skena vidljivi tek na kraju tog sken ciklusa. Drugim rečima, sva postavljanja vrednosti bitova učinjena u jednom prolasku kroz leder dijagram se odlažu do kraja tog sken ciklusa. Ili, leder program u tokom jednog sken ciklusa barata sa vrednostima koje su bitovi imali na početku sken ciklusa, bilo da se ti isti bitovi modifikuju leder programom ili ne. Model izvršavanja leder programa je sličan modelu izvršavanja VHDL procesa, kod kojih se takođe, kao što znamo, promene učinjene nad signalima u toku jednog izvršenja procesa prenose na signale tek na kraju procesa. U konkretnom primeru to znači da u drugom rangu $Tev=1$ i $Q=0$ ne setuje Q istog momenta, već će se setovanje obaviti tek kada se izvrše svi rangovi, a da će se u uslovu trećeg ranga koristiti stara vrednost Q-a, tj. vrednost koju je bit Q imao na početku sken ciklusa, tj. $Q=0$. Zbog toga će uslov u trećem rangu biti netačan, konačni efekat na kraju sken ciklusa će biti $Q=1$!



Sl. 4-37 Leder dijagram za problem 11

Primer 12. Sistem za upravljanje garažnim vratima (Sl. 4-38).



Sl. 4-38 Sistem za upravljanje garažnim vratima

Vrata se spuštaju/podizaju pomoću dvosmernog motora. Motorom se upravlja pomoću dva signala MotV i MotD. Za MotV=ON motor je uključen, a za MotV=OFF isključen. MotD određuje smer rotacija motora, tako da se za MotD=ON vrata spuštaju, a za MotD=OFF podizaju.

Vrata se mogu spustiti/podignuti pomoću tastera na zidu u garaži (TZ) ili pomoću tastera na daljinskom upravljaču. Signal iz daljinskog upravljača se prenosi do prijemnika, a izlaz iz prijemnika (DU) i taster u garaži su ravnopravni u smislu uticaja na rad sistema.

Pritisak na bilo koji od dva taster za vreme dok su vrata u pokretu, zaustavlja vrata. Sledeći pritisak ponovo pokreće vrata, ali u suprotnom smeru.

Granični prekidači GG i GD služe za indicaciju krajnje gornje (GG) i krajnje donje (GD) pozicije vrata. Vrata, kada su jednom pokrenuta, spuštaju se do nivoa prekidača GD, odnosno podizaju do nivoa prekidača GG, osim ako spuštanje/podizanje nije prekinuto pritiskom na taster (TZ ili DU).

Dakle, ulazi u PLC sistem su: TZ, DU, GG i GD, a izlazi MotV i MotD.

Rešenje:

Zadatak ćemo rešiti tako što ćemo najpre identifikovati događaje koji se javljaju u sistemu, a zatim ćemo definisati kako događaji utiču na rad (stanje) motora. Događaj je promena stanja nekog ulaznog signala. Od interesa su sledeća dva događaja:

- Pritisnut taster (TZ ili DU).
- Aktiviran granični prekidač (GG ili GD)

Razmatrano na apstraktnom nivou, događaj nema trajanje, tj. možemo ga zamisliti kao impuls beskonačno kratkog trajanja koji ima dejstvo samo u trenutku kada se desio. Na primer, nije od interesa stanje graničnog prekidača (0 ili 1) već samo trenutak promene sa 0 na 1 (trenutka zatvaranja njegovih kontakta).

Događaji utiču na stanje sistema, tj. menjaju stanje izlaza, tako što setuju ili resetuju izlazne signale. Na primer, događaj "aktiviran granični prekidač" isključuje motor (tj. resetuje signal MotV). Dejstvo događaja "pritisnut taster" nešto složenije, jer zavisi od trenutnog stanja sistema (da li vrata stoje, da li se spuštaju ili podižu ili da li su u krajnjem gornjem ili krajnjem donjem položaju).

U leder jeziku, apstrakciju događaja možemo ostvariti pomoći naredbe OSR koja detektuje rastuću ivicu uslova koji joj prethodi u rangui i postavlja bit, koji sledi u naredi akcije, na 1 u trajanju od jednog sken ciklusa.

Leder dijagram je prikazan na slici. Prva dva ranga služe za detekciju dva događaja koja su od interesa za upravljanje sistemom. Svakom događaju odgovara jedan pomoćni bit (iz interne memorije PLC kontrolera, tj. iz datoteke B). Bit događaja (Tev ili Gev) će imati vrednost 1 samo u onom sken ciklusu kada se događaj desio.

Smer kretanja vrata se postavlja u trenutku uključena motora (Tev=ON i MotV=OFF), a da li će smer biti postavljen na spuštanje (MotD=ON) ili podizanje (MotD=OFF), zavisi od dodatnih uslova (rangovi 3 i 4).

Rang 3. Smer "spuštanje" se postavlja (setovanje izlaznog bita MotD) ako su vrata u krajnjem gornjem položaju ili ako su prilikom podizanja vrata pre vremena zaustavljena (tj. GG nije aktivan i smer je "podizanje").

Rang 4. Smer "podizanje" se postavlja (resetovanje izlaznog bita MotD) ako su vrata u krajnjem donjem položaju ili ako su prilikom spuštanja vrata pre vremena zaustavljena (tj. GD nije aktivan i smer je "spuštanje").

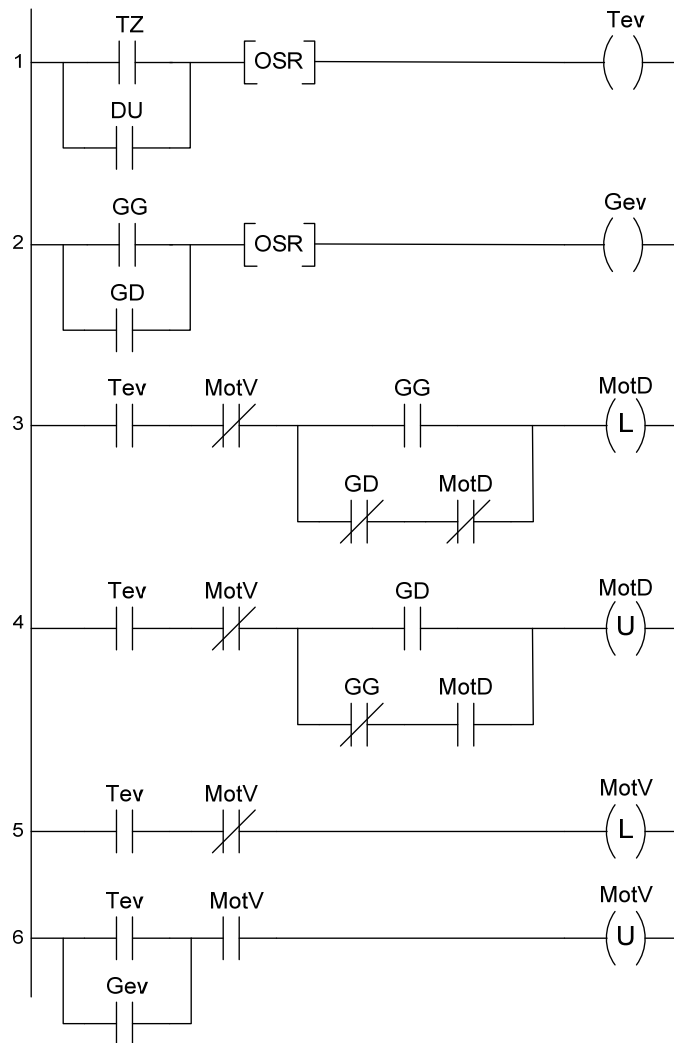
Signali motora, MotV i MotD, se postavljaju pomoću naredbi OTL i OTU. Setovanje, odnosno resetovanje izlaza MotV i MotD uvek je inicirano događajima, tj. bitovima Tev i Gev, a da li će se setovanje ili resetovanje desiti ili ne, dodatno zavisi od trenutnog stanja sistema.

Rang 5. Motor se uključuje:

- pritiskom na taster, pod uslovom da je motor isključen.

Rang 6. Motor se isključuje:

- pritiskom na taster, pod uslovom da je motor uključen.
- aktiviran granični prekidač



Sl. 4-39 Leder dijagram za problem 12

4.6 Metod konačnih automata

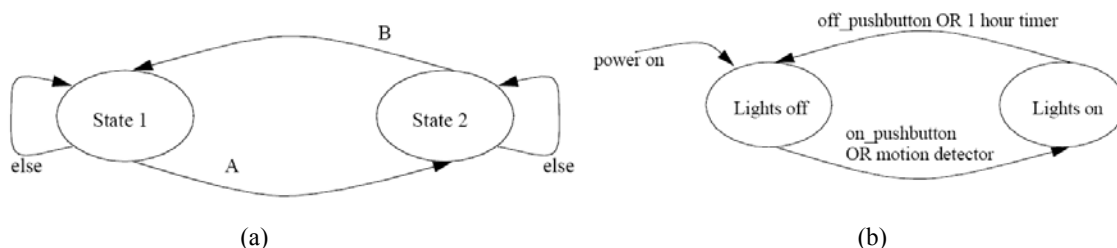
Većina sistema u oblasti automatskog upravljanja su, po svojoj prirodi, sekvencijalni. Tokom normalnog rada, sekvencijalni sistem prolazi kroz više koraka, tj. faza ili stanja. U svakom stanju, sistem se ponaša na drugačiji način. Na primer, zamislimo semafor za regulisanje saobraćaja. Jedno stanje semafora može biti ono kada je dozvoljen saobraćaj u jednom, a drugo kada je dozvoljen saobraćaj u drugom smeru. Svakom stanju semafora odgovara jedna specifična kombinacija upaljenih crvenih, žutih i zelenih svetala, a stanja se izmenjuju po unapred utvrđenom redosledu, tj. sekvenci. Ova sekvenca može biti fiksna (nepromenljiva), ali se može i menjati pod dejstvom nekih specifičnih ulaznih uslova. Na primer, ako na semaforu postoji "pešačko dugme", tada će redosled u kome se izmenjuju semaforska svetla svakako zavisiti i od toga da li je dugme pritisnuto ili ne. Dakle, pod stanjem se može smatrati vremenski interval u kojem su izlazi sistema stabilni (ne menjaju se) i u kome sistem reaguje na ulaznu pobudu na način specifičan za to stanje.

Sekvencijalni sistemi su po pravilu složeni, a njihovo projektovanje je otežano obiljem detalja o kojima treba voditi računa. Zbog toga, direktno projektovanje, gde se na osnovu polazne specifikacije, a bez neke veće razrade, odmah pristupa implementaciji (pisanju programa, ili crtanju leder dijagrama) ima ograničen domet, a dobijena rešenja često nisu ispravna ili zahtevaju naknadno temeljno testiranje i ispravljanje učinjenih grašaka. Iz tog razloga, razvijeno je više, u osnovi srodnih, metoda projektovanja sekvencijalnih sistema kod kojih je naglasak upravo na razradi problema sa ciljem da se, pre same implementacije, detaljnom analizom identifikuju karakteristike sistema, kao što su stanja, prelazi i događaji, na osnovu kojih se kreira apstraktni model ponašanja sistema koji se potom koristi za direktnu implementaciju. Jedan od takvih metoda je upravo metod konačnih automata. Ovaj metod se primenjuje ne samo za opis ponašanja sistema automatskog upravljanja, već i u mnogim drugim oblastima, kao što je projektovanje upravljačkih jedinica digitalnih sistema, ili projektovanje konkurentnog softvera.

Model konačnog automata zasnovan je na konceptu *stanja* i *prelaza* između stanja. U svakom trenutku, automat je u jednom od konačnog broja svojih stanja. Pod dejstvom događaja ili uslova, a u zavisnosti od tekućeg stanja i trenutne vrednosti ulaza, automat prelazi u novo stanje i generiše odgovarajuće izlaze. Konačni automat se može predstaviti dijagramom (grafom) stanja. U ovom dijagramu, krugovima su predstavljena stanja, a strelicama (granama) prelazi između stanja.

Dijagram stanja prikazan na Sl. 4-40(a) ima dva stanja State 1 i State 2. Ako je sistem u stanju State 1 i desi se događaj A, sistem prelazi u stanje State 2, inače ostaje u stanju State 1. Slično, ako je sistemu stanju State 2 i desi se događaj B, sistem se vraća u stanje State 1, inače ostaje u stanju State 2. Grane na Sl. 4-40(a) označene sa *else* važe u slučajevima kada ni jedan uslov naveden na izlaznim granama iz dataog stanja nije ispunjen. Uobičajeno je da se ove grane ne crtaju, već se podrazumevaju, tj. ako ni jedan uslov koji vodi sistem u neko drugo stanje nije ispunjen, sistem ostaje u zatečenom stanju.

Dijagram stanja sa Sl. 4-40(a) može se iskoristiti za modelovanje ponašanja kontrolera osvetljenja, kao što je prikazano na Sl. 4-40(b). Stanjima su sada data imena koja ukazuju na režim rada sistema. U stanju *Light off* svetlo je ugašeno, a u stanju *Ligh on* svetlo je upaljeno. Događaj koji prevodi sistem iz stanja *Light off* u stanje *Light on* je "prekidač uključen ili detektovan pokret". Događaj koji gasi svetlo, tj. prevodi sistem iz stanja *Ligh on* u stanje *Light off* je "isteklo vreme od 1h ili prekidač isključen". Strelica usmerena ka stanju *Light off* i označena sa *reset* ukazuje na početno stanje sistema, tj. na stanje u koje će sistem biti postavljen kada počne sa radom.



Sl. 4-40. (a) Dijagram stanja sa dva stanja; (b) dijagram stanja kontrola osvetljenja.

Pr. 4-1. Dijagram stanja upravljačke jedinice lift

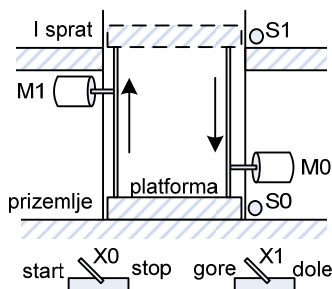
Razmatramo projektovanje upravljačke jedinice lifta koja se koristi za prenos tereta između prizemlja i I sprata (Sl. 4-41). Sistem čine: a) pokretna platforma; b) Start/stop prekidač (X0; u položaju *start* važi X0=ON, a u položaju *stop* X0=OFF); c) Gore/Dole prekidač (X1; u položaju *gore* važi X1=ON, a u položaju *dole* X1=OFF); d) dva senzora za indicaciju krajnjih pozicija platforme (S0 – platforma je u prizemlju; S1 – platforma je na I spratu); e) motor za podizanje platforme (M1) i f) motor za spuštanje platforme (M0).

Prebacivanjem prekidača X0 u položaj Start, platforma se spušta u prizemlje, bez obzira na svoju trenutnu poziciju.

Prebacivanjem prekidača X0 u položaj Stop, platforma se trenutno zaustavlja.

Ako je prekidač X1 u položaju Gore, platforma se podiže na I sprat. Ako se u trenutku prebacivanja prekidača u položaj Gore platforma kreće naniže, platforma se spušta do prizemlja, a zatim podiže na I sprat.

Ako je prekidač X1 u položaju Dole, platforma se spušta u prizemlje. Ako se u trenutku prebacivanja prekidača u položaj Dole, platforma kreće naviše, platforma se podiže do I sprata, a zatim spušta u prizemlje.



Sl. 4-41 Lift.

Prvi, i često najvažniji korak u postupku projektovanja podrazumeva kreiranje dijagrama stanja. Pri tome, prilikom identifikovanja stanja sistema, treba imati na umu sledeće:

- Stanja su stvar subjektivnog izbora projektanta. Razmatrajući isti sistem, dva projektanta mogu doći različitih skupova stanja i kreirati različite dijagrame stanja, a da oba dijagrama budu korektan.
- Stanja su svojstvena konkretnom sistemu i imaju smisla samo u kontekstu sistema koji se opisuje.
- Prethodna i sledeća stanja nekog stanja **nisu** deo tog stanja. Drugim rečima, ponašanje sistema u nekom stanju ne zavisi od ponašanja sistema u drugim stanjima, tj. sistem se u nekom stanju S ponaša uvek isto, bez obzira iz kog stanja je sistem stigao u stanje S, odnosno u koje naredno stanje će sistem preći.
- Između stanja treba da postoji jasno razgraničenje
- Podelu na stanja treba vršiti na način da u toku svakog stanja, sistem obavlja jednu aktivnost koja ima svoj početak i kraj (ta aktivnost može biti i prosto čekanje da se nešto desi).

Razmatrajući specifikaciju rada lifta, u mogućnosti smo da identifikujemo sledećih pet stanja ovog sistema:

Stanje	Opis
Isključeno	Lift je isključen
Dole	Lift je u prizemlju
Gore	Lift je na I spratu
Podizanje	Lift se podiže na I sprat
Spuštanje	Lift se spušta u prizemlje

Stanja su dobro identifikovana ako:

- U svakom stanju izlazi sistema imaju definisanu, nepromeljivu vrednost. U konkretnom sistemu, postoje dva izlaz, M1 i M0, koji upravljaju motorima za podizanje (M1), odnosno spuštanje (M0) lifta. Motor M1 je uključen (M1=ON) u stanju Podizanje, a isključen u svim ostalim stanjima. Slično, motor M0 je uključen (M0=ON) u stanju Spuštanje, a isključen (M0=OFF) u svim ostalim stanjima.

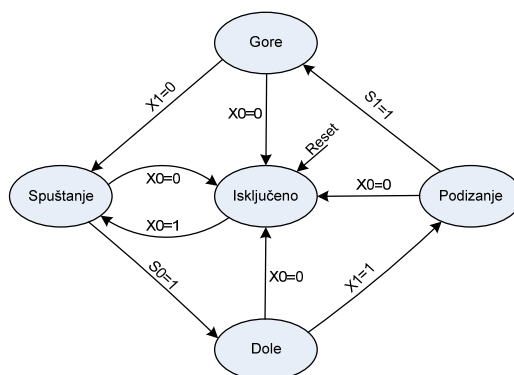
Stanje	Izlazi	
	M1	M0
Isključeno	0	0
Dole	0	0
Gore	0	0
Podizanje	1	0
Spuštanje	0	1

- U svakom stanju, sistem čeka na uslov (događaj) koji će ga prevesti u sledeće stanje. Stanju može biti pridružen jedan ili više uslova (prelaza). U konkretnom sistemu, stanje Isključeno se napušta kada se prekidač X0 prebaci u položaj Start, tj. X0=1, a novo stanje je Spuštanje. U stanju Dole postoje dva uslova (događaja) koji uslovljavaju napuštanje ovog stanja: X0 prebačen u položaj STOP (novo stanje je Isključeno) ili prekidač X1 prebačen u položaj Gore (novo stanje je Podizanje). Uslov za kraj stanja Gore je pored X0=0 i prebacivanje prekidača X1 u položaj Dole (X1=0), što vodi sistem u stanje Spuštanje. Uslov za kraj stanja Podizanje i Spuštanje je aktiviranje graničnog prekidača. Stanje Podizanje se završava kada se aktivira S1, što ukazuje da je lift na I spratu, odnosno da sistem treba da pređe u stanje Gore. Spuštanje se završava kada se aktivira S0 (lift je u prizemlju), a novo stanje je Dole:

Stanje	Uslov	Novo stanje
Isključeno	X0=1	Spuštanje
Dole	X0=0	Isključeno
	X1=1	Podizanje
Gore	X0=0	Isključeno
	X1=0	Spuštanje
Podizanje	X0=0	Isključeno
	S1=1	Gore
Spuštanje	X0=0	Isključeno
	S0=1	Dole

Primitimo da je u stanjima Podizanje i Spuštanje, sistem neosetljiv na promene prekidača X1. To znači da će sistem, uprkos promeni položaj ovog prekidača nastaviti da se podiže, odnosno spušta do svoje krajnje pozicije, Gore ili Dole, a tek će onda ispitati X1 i započeti kretanje u suprotnom smeru ako je X1 promenjeno (što je u skladu za postavkom zadatka).

Na osnovu uvedenih stanja i identifikovanih prelaza između stanja, lako možemo nacrtati dijagram stanja (Sl. 4-42).



Sl. 4-42 Dijagram stanja upravljačke jedinice lifta.

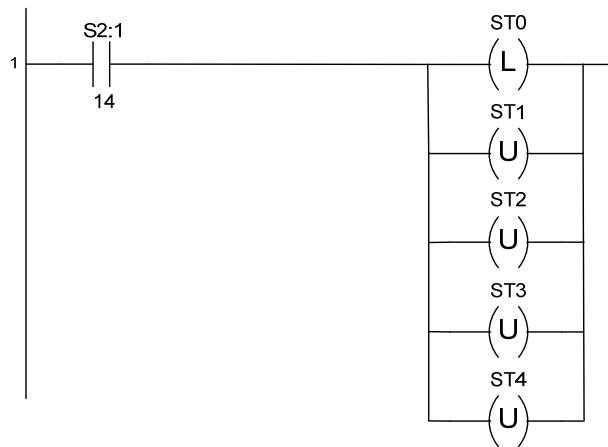
4.6.1 Realizacija konačnog automata u leder jeziku

Model konačnog automata uvodi pojam stanja. Dakle, pored ulaza i izlaza, sistem u svom radu dodatno manipuliše i stanjima. Informacija o tekućem stanju mora da se pamti, kako bi sistem znao u kom je stanju i shodno tome kako da reaguje. Za pamćenje tekućeg stanja koriste se bitovi iz interne memorije PLC kontrolera (iz datoteke B). Po pravilu, koristi se *one-hot* kodiranje stanja. To znači da se svakom stanju pridružuje jedan bit koji ima vrednost 1 dok je sistem u tom stanju. Kada sistem menja stanje, bit tekućeg stanja se resetuje, a bit sledećeg setuje. Na ovaj način, uvek je setovan tačno jedan bit stanja, dok su ostali resetovani.

U tabeli ispod uvedeni su bitovi stanja za primer upravljačke jedinice lifta. Radi kompletnosti dati su i spiskovi ulaza i izlaza zajedno sa njihovim usvojenim fizičkim adresama.

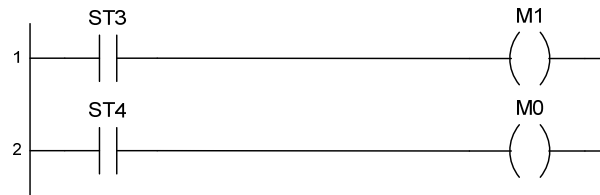
Stanja			Ulazi			Izlazi		
Stanje	Simboličko ime	Bit stanja	Ulazni uređaja	Simboličko ime	Fizička adresa	Izlazni uređaj	Simboličko ime	Fizička adresa
Isključeno	ST0	B3:1/0	Prekidač start/stop	X0	I:1/0	Motor za podizanje lifta	M1	O:2/0
Dole	ST1	B3:1/1	Prekidač gore/dole	X1	I:1/1	Motor za spuštanje lifta	M0	O:2/1
Gore	ST2	B3:1/2	Gran. prek. u prizemlju	S0	I:1/2			
Podizanje	ST3	B3:1/3	Gran. prek. na I spratu	S1	I:1/3			
Spuštanje	ST4	B3:1/4						

U leder dijagramu koji realizuje konačni automat mogu se uočiti više delova, od kojih se svaki odnosi na jedan aspekt rada sistema. Prvi deo se bavi postavljanjem sistema u inicijalno stanje. Naime, neophodno je obezbediti da na početku rada, sistem bude postavljen u odgovarajuće početno stanje, tj. potrebno je setovati bit inicijalnog stanja i resetovati bitove svih ostalih stanja. Za ovu namenu se može koristiti jedan poseban bit iz statusne datoteke PLC kontrolera koji ima vrednost 1 samo u toku prvog sken ciklusa, a u svim ostalim vrednost 0. Ovaj bit se zove "first scan bit", a njegova adresa kod kontrolera SLC 500 je S2:1/14. Na Sl. 4-43 je prikazan segment leder dijagrama upravljačke jedinice lifta koji se bavi inicijalizacijom stanja. Kao što vidimo, ovaj deo leder dijagrama se sastoji iz samo jednog ranga koji za slučaj da je "first scan bit" jednak 1, setuje bit ST0 (bit inicijalnog stanja, tj. stanja Isključeno), a resetuje bitove svih ostalih stanja.



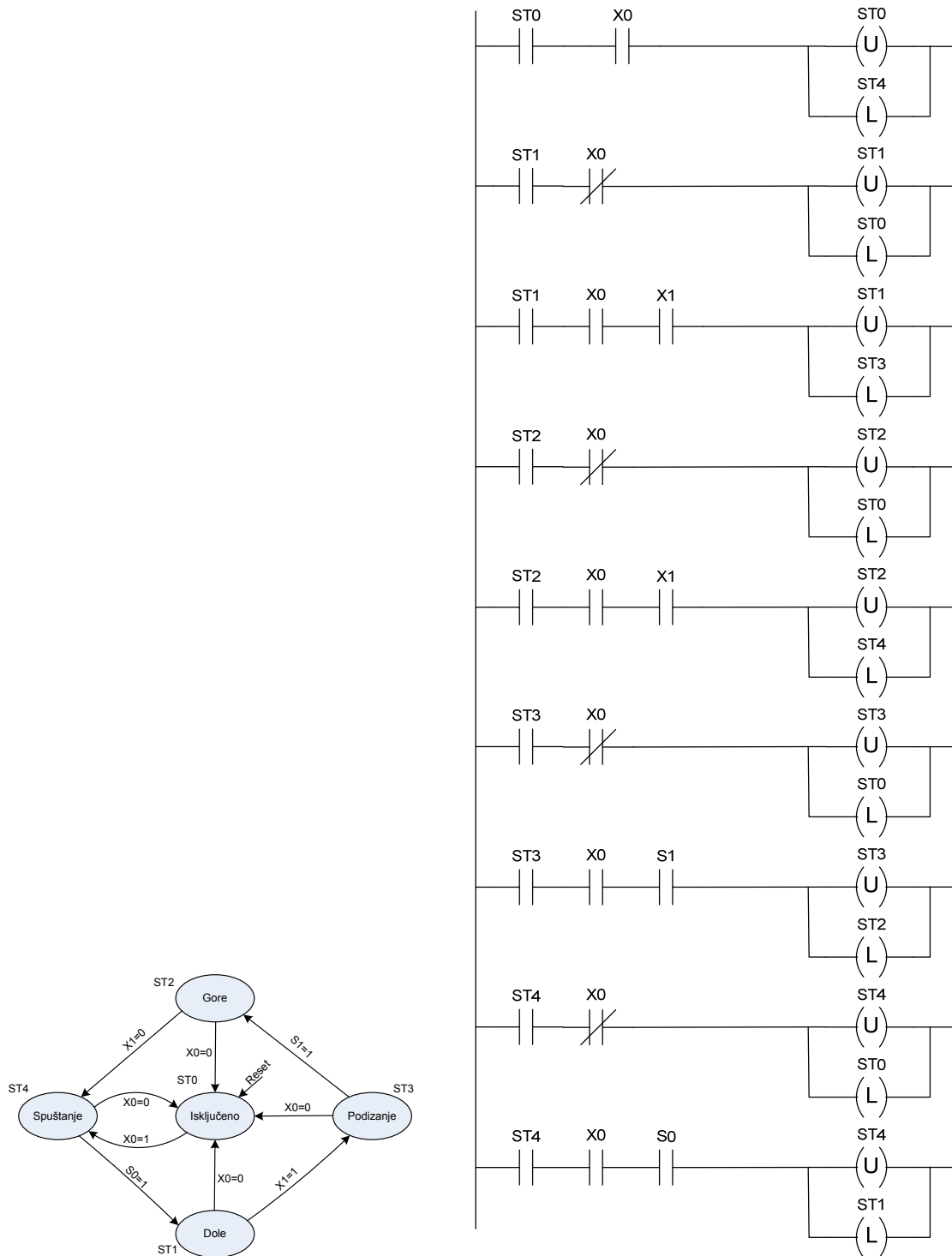
Sl. 4-43 Leder dijagram upravljačke jedinice lifta (Inicijalizacija).

Drugi deo leder dijagrama se bavi pobuđivanjem izlaza. Svaki izlaz je pobuđen u jednom ili više stanja. Za svaki izlaz se formira jedan rang, koji pobuđuje taj izlaz, a čiji je uslov formiran OR-ovanjem bitova stanja u kojima izlaz ima vrednost 1. Ovaj deo leder dijagrama upravljačke jedinice lifta prikazan je na Sl. 4-44. Motor M1 je uključen dok se lift podiže (tj. dok je sistem u stanju Podizanje - ST3), dok je motor M0 uključen dok se lift spušta (stanje Spuštanje - ST4).



Sl. 4-44 Leder dijagram upravljačke jedinice lifta (Postavljanje izlaza).

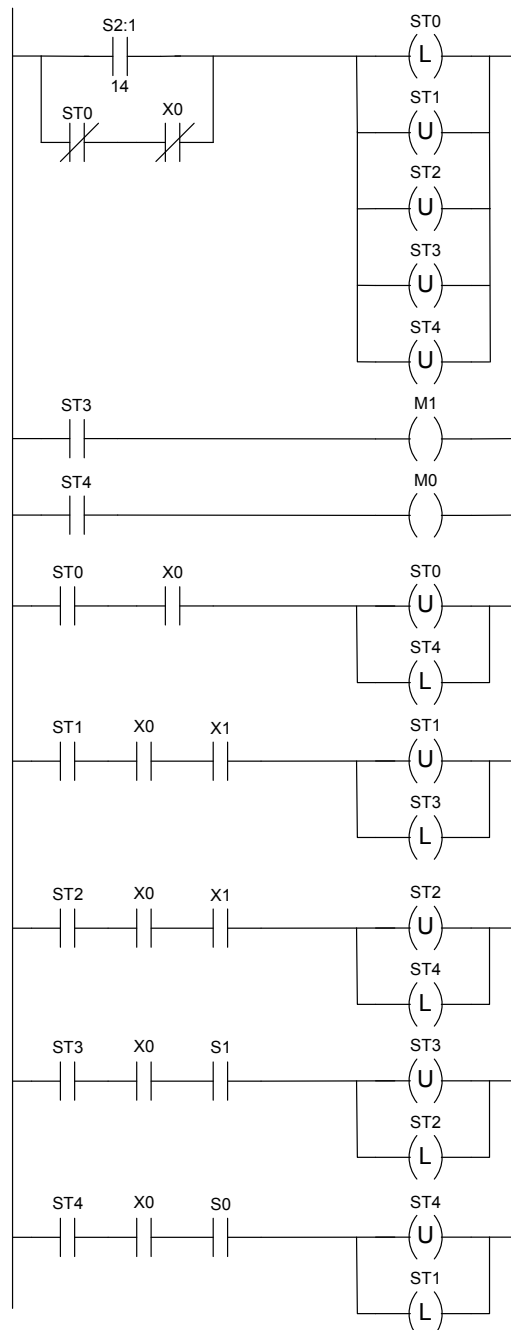
Treći deo leder dijagrama je zadužen za promenu stanja. Za svaki prelaz iz dijagrama stanja kreira se jedan rang. Uslov ranga kombinuje bit izvornog stanja i uslov koji odgovara tom prelazu. U delu akcije, resetuje se bit izvornog (tekućeg), a setuje bit odredišnog (sledećeg) stanja. Na Sl. 4-45 je prikazan deo leder dijagrama upravljačke jedinice lifta koji se bavi prelazima. Uočimo da za svaki prelaz (granu) iz dijagrama stanja, u ovom delu leder dijagrama postoji jedan rang. Uslov ranga je AND operacija nad bitom stanja i uslovom koji je pridružen odgovarajućem prelazu. U delu akcije ranga, uvek postoje dve naredbe, jedna resetuje bit tekućeg, a druga setuje bit sledećeg stanja.



Sl. 4-45 Leder dijagram upravljačke jedinice lifta (Promena stanja).

Leder dijagram upravljačke jedinice lifta (Sl. 4-45) dobijen je na osnovu dijagrama stanja sa Sl. 4-42. primenom jednostavnih pravila konverzije. Najopširniji deo dijagrama je onaj koji reguliše prelaze između stanja. Veliki broj rangova u ovom delu je posledica toga što se svaka tranzicija razmatra nezavisno. Međutim, ako se uoče izvesne zakonitosti u ponašanju sistema, koje važe za više različitih prelaza, dijagram se može uprostiti (smanjiti broj rangova). Na primer, prebacivanjem prekidača X0 u položaj Stop ($X0=0$), sistem prelazi u stanje Isključeno (tj. u inicijalno stanje), bez obzira u kom stanju se sistem trenutno nalazi. Ovo zapažanje nam omogućava da iz leder dijagrama eliminišemo

rangove koji odgovaraju ovim tranzicijama, a uslov "X0=0" pridružimo prvom delu leder dijagrama, koji se bavi inicijalizacijom sistema. Kompletan leder dijagram, nakon učinjene optimizacije, prikazan je na Sl. 4-46.



Sl. 4-46 Konačni leder dijagram upravljačke jedinice lifta.

4.6.1.1 Jednačine stanja

Alternativni način za konverziju dijagrama stanja u leder dijagram sastoji se u tome da se dijagram stanja najpre prevede u logičke jednačine koje se potom prevode u leder dijagram. Pri tome, moguća su dva pristupa. Prvi je da se dijagram stanja prevede u *jednačine stanja*, a drugi da se dijagram stanja prevede u *jednačine prelaza*. U ovom odeljku, razmotrićemo jednačine stanja.

Postupak konverzije podrazumeva da se za svako stanje piše se jedna logička jednačina. Formalna definicija jednačine stanja data je ispod. Pojednostavljeno rečeno, dato stanje biće ON ako je već ON

ili je postavljeno na ON prelazom iz nekog drugog stanja i pri tome nije ispunjen ni jedan uslov za napuštanje tog stanja. Prvi deo jednačine (u zagradama) "uključuje" stanje, a drugi ga "isključuje". Napomenim da jednačinu inicijalnog stanja treba proširiti članom "+ FS", gde je FS "first scan bit".

$$STATE_i = \left(STATE_i + \sum_{j=1}^n (T_{j,i} \cdot STATE_j) \right) \cdot \prod_{k=1}^m (\overline{T_{i,k}} \cdot STATE_i)$$

gde je,

$STATE_i$ - bit stanja i

n - broj prelaza u stanje i

m - broj prelaza iz stanja i

$T_{j,i}$ - logički uslov za prelaz iz stanja j u stanje i

$T_{i,k}$ - logički uslov za prelaz iz stanja i u stanje k

Napišimo jednačine stanja za primer upravljačke jedinice lifta shodno dijagramu stanja sa Sl. 4-42. Počnimo od stanja Dole. U ovo stanje se dolazi iz stanja Spuštanje kada se aktivira senzor S0, a iz ovog stanja se odlazi kada se lift isključi (X0=0) ili se X1 prebaci u položaj gore (X1=1). Dakle, jednačina stanja Dole glasi:

$$ST1 = (ST1 + ST4 \cdot S0 \cdot X0) \cdot \overline{ST1} \cdot \overline{X0} \cdot \overline{ST1} \cdot X1$$

U stanje Dole uvire samo jedna grana (izvire iz stanja Spuštanje) i zato u zagradi, pored ST1 postoji samo jedan dodatni član, $ST4 \cdot S0 \cdot X0$, koji odgovara ovom prelazu. Uočimo da je pored uslova $S0=1$, ovim članom obuhvaćen i uslov $X0=1$, koji nije naveden kao uslov na grani koja povezuje stanja Spuštanje i Dole. Iz stanja Spuštanje izvire dve grane, jedna ka stanju Dole, a druga ka stanju Isključeno (za slučaj $X0=0$). U takvim situacijama, kada iz nekog stanja izvire više od jedne grane, može se desiti da bude zadovoljeno više od jednog uslova za napuštanje stanja (npr. može se desiti da dok je sistem u stanju Spuštanje istovremeno se X0 promeni na 0 i S0 na 1. Postavlja se pitanje u koje stanje će sistem preći. Da bi se razrešila ova dilema, izlaznim granama se dodeljuju prioriteta, a sistem prelazi u stanje u koje vodi grana (sa ispunjenim uslovom) najvišeg prioriteta. U konkretnom slučaju, za stanje Spuštanje, viši prioritet se daje grani koja ovo stanje povezuje sa stanjem Isključeno. To znači da ako se u stanju Spuštanje u isto vreme prekidač X0 prebaci u položaj *stop* i aktivira senzor S0, sistem će preći u stanje Isključeno. Upravo je to razlog za prisustvo bita X0 u članu $ST4 \cdot S0 \cdot X0$ ($X0=0$, poništava $S0=1$ i sistem neće preći u stanje ST1, tj. Dole).

Uočimo, takođe da jednačina stanja Dole nije pogodna za direktnu relaizaciju pomoću leder naredbi. Problem je u komplementiranim članovima na desnom kraju jednačine (u leder jeziku ne postoji naredba za komplement logičkih izraza složenijih od jedne promenljive). Zato je neohodno, primenom DeMorganovih pravila eliminisati "dugačke" komplemente:

$$\overline{ST1} \cdot \overline{X0} \cdot \overline{ST1} \cdot \overline{X1} = (\overline{ST1} + X0)(\overline{ST1} + \overline{X1}) = \overline{ST1} + X0 \cdot \overline{X1}$$

Konačno:

$$ST1 = (ST1 + ST4 \cdot S0 \cdot X0)(\overline{ST1} + X0 \cdot \overline{X1})$$

Na sličan način možemo da izvedemo i jednačine za stanja Gore, Podizanje i Spuštanje:

$$ST2 = (ST2 + ST3 \cdot S1 \cdot X0)(\overline{ST2} + X0 \cdot X1)$$

$$ST3 = (ST3 + ST1 \cdot X1 \cdot X0)(\overline{ST3} + X0 \cdot \overline{S1})$$

$$ST4 = (ST4 + ST2 \cdot \overline{X1} \cdot X0)(\overline{ST4} + X0 \cdot \overline{S0})$$

Konačno, razmotrimo i inicijalno stanje, Isključeno. Sledeći opštu jednačinu stanja (uz dodatak bita FS) dolazimo do sledećeg izraza:

$$ST0 = (ST0 + ST1 \cdot \overline{X0} + ST2 \cdot \overline{X0} + ST3 \cdot \overline{X0} + ST4 \cdot \overline{X0})(\overline{ST0} \cdot \overline{X0}) + FS$$

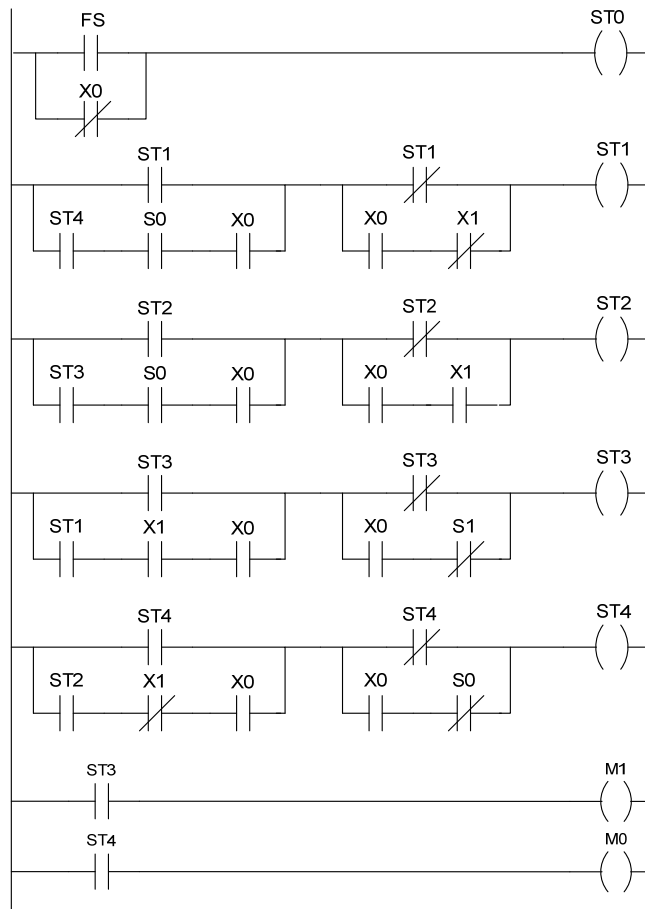
Uočimo da članovi iz prve zagrade koji odgovaraju granama koje uviru u stanje Isključeno sadrže uvek samo uslov $X0=0$. To je iz razloga što svaka od ovih grana imaju viši prioritet u odnosu na drugu granu koja izvire iz istog stanja. Takođe, primetimo da izraz u prvoj zagradi može da se svede na:

$$(ST0 + ST1 \cdot \overline{X0} + ST2 \cdot \overline{X0} + ST3 \cdot \overline{X0} + ST4 \cdot \overline{X0}) = ST0 + \overline{ST0} \overline{X0} = ST0 + \overline{X0}$$

zato što je $ST1 + ST2 + ST3 + ST4$ isto što i $\overline{ST0}$. Konačno, u kombinaciji sa preostalim delo izraz, dolazimo do jednačine stanja Isključeno u obliku:

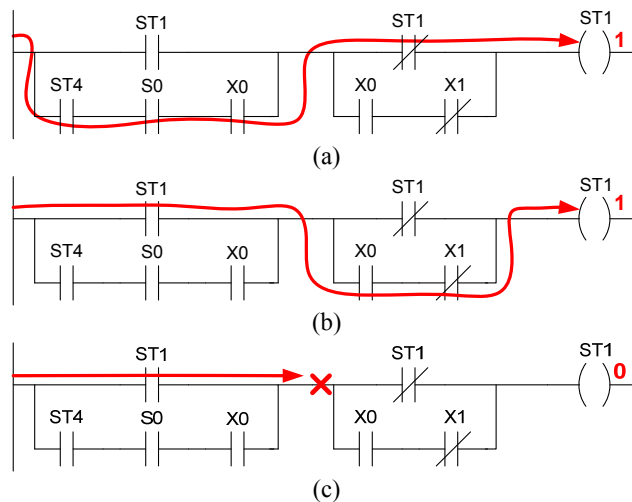
$$ST0 = \overline{X0} + FS$$

Pošto smo izveli jednačine svih stanja, spremni smo za crtanje leder dijagrama. Deo leder dijagrama koji postavlja izlaze (Sl. 4-44) ostaće isti, dok delove koji se bave inicijalizacijom (Sl. 4-43) i prelazima između stanja (Sl. 4-45) zamenjujemo rangovima koji realizuju jednačine stanja (Sl. 4-47).



Sl. 4-47 Leder dijagram upravljačke jedinice lifta kreiran na osnovu jednačina stanja.

Lako možemo uočiti pravilnost u strukturi "rangova stanja". Svaki rang koji reguliše bit stanja u delu uslova sadrži dve redno vezane paralelne sekcije. Prva (leva) sekcija je zadužena za ulazak u dato stanje i samodržanje dok je sistem u datom stanju. Duga sekcija reguliše napuštanje datog stanja, tj. definiše uslove pod kojima sistem više nije u datom stanju. Razmotrimo rang za stanje ST1. Na Sl. 4-48 je prikazan tok signala prilikom ulaska u stanje ST1 (Sl. 4-48a), za vreme dok je u stanju ST1 (Sl. 4-48b) i prilikom napuštanja stanja ST1 (Sl. 4-48c).

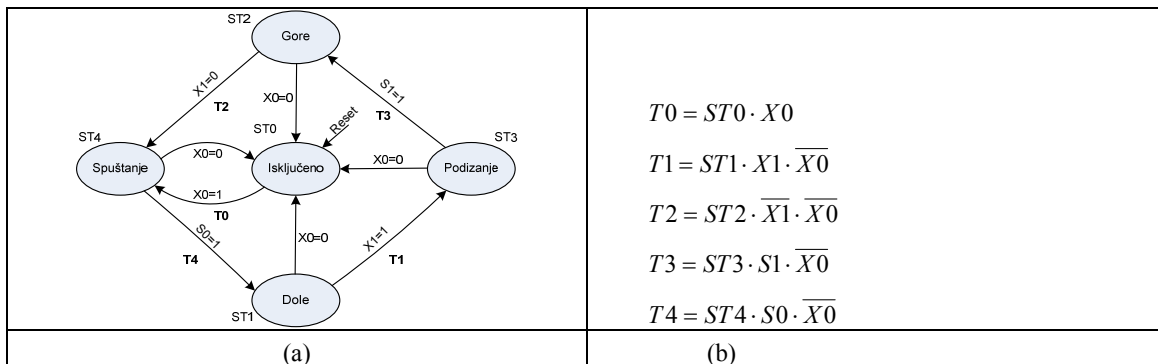


Sl. 4-48 Upravljanje bitom stanja: (a) ulazak u stanje ($ST1=OFF$, i $ST4=S0=X0=ON \Rightarrow ST1=ON$); (b) samodrzanje ($ST1=ON$ i $X0=ON$ i $X1=OFF \Rightarrow ST1=ON$); (c) napuštanje stanja ($ST1=ON$ i $X0=OFF$ ili $X1=ON \Rightarrow ST1=OFF$).

4.6.1.2 Jednačine prelaza

Dijagram stanja se može konvertovati u leder dijagram i tako što će se najpre izvesti jednačine za prelaze koje će se potom iskoristiti za pisanje jednačina stanja. U ovom slučaju, za svaki prelaz (granu) iz dijagrama stanja uvodimo poseban pomoćni bit (bit prelaza) koji dobija vrednost 1 kada je sistem u stanju iz kojeg prelaz izvire i uslov pridružen prelazu tačan, inače dobija vrednost 0. Na ovaj način se pojednostavljaju jednačine stanja, jer umesto celokupnih uslova prelaza u njima sada figurišu samo bitovi prelaza.

Razmotrimo, ponovo, problem upravljačke jedinice lifta i izvedimo jednačine prelaza shodno dijagramu stanja sa Sl. 4-49(a). Dijagram stanja sadrži 10 grana (računajući i *reset* granu). Međutim, kao i do sada, prelaze koje sistem vode u stanje Isključeno razmatraćemo objedinjeno, što smanjuje broj prelaza za koje ćemo pisati jednačine na 5. U tom cilju, rezervisaćemo 5 bit-promenljivih, T1, T2, T3, T4 i T5 - jedan bit za svaki prelaz (Sl. 4-49a). Odgovarajuće jednačine prelaza su date na Sl. 4-49b.



Sl. 4-49 Tranzicije u grafu stanja upravljačke jedinice lifta: (a) označavanje tranzicija; (b) jednačine prelaza.

Sa jednačinama prelaza na raspolaganju u mogućnosti smo da napišemo jednačine stanja:

$$ST1 = (ST1 + T4) \cdot \overline{T1}$$

$$ST2 = (ST2 + T3) \cdot \overline{T2}$$

$$ST3 = (ST3 + T1) \cdot \overline{T3}$$

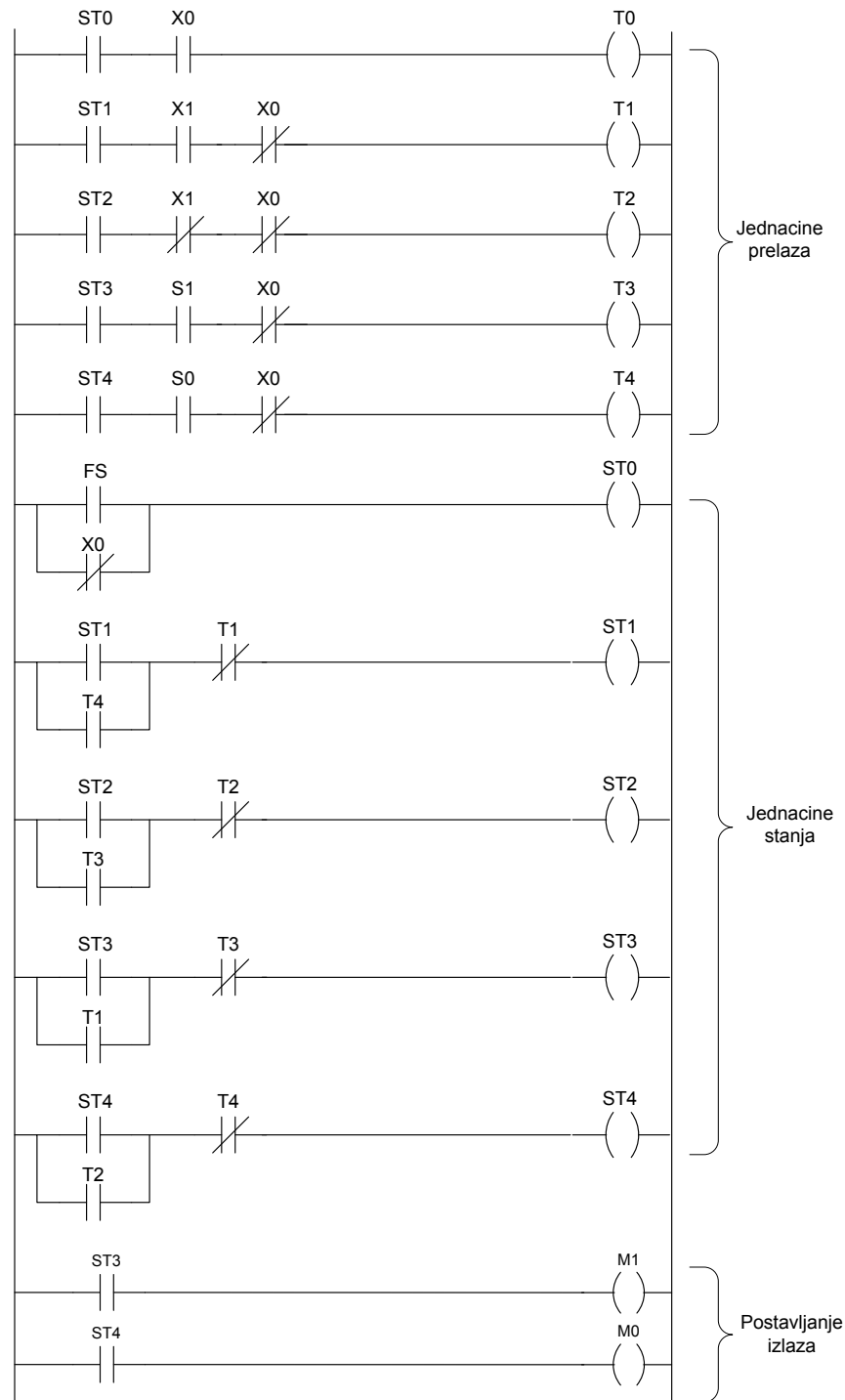
$$ST4 = (ST4 + T2) \cdot \overline{T4}$$

Opšti oblik jednačine stanja je identičan onome koji smo već koristili: Sistem je u datom stanju ako je već u tom stanju ili je ispunjen uslov (tranzicije) za ulazak u to stanje (izrazi u zagradi); sistem nije više u datom stanju ako je uspunjen uslov izlazne tranzicije iz tog stanja.

Jednačina stanje Isključeno je, kao i ranije:

$$ST0 = \overline{X0} + FS$$

Na Sl. 4-50 je prikazan leder dijagram kreiran na osnovu izvedenih jednačina.



Sl. 4-50 Leder dijagram upravljačke jedinice lifta kreiran na osnovu jednačina prelaza i stanja.

Pr. 4-2 Mašina za presovanje.

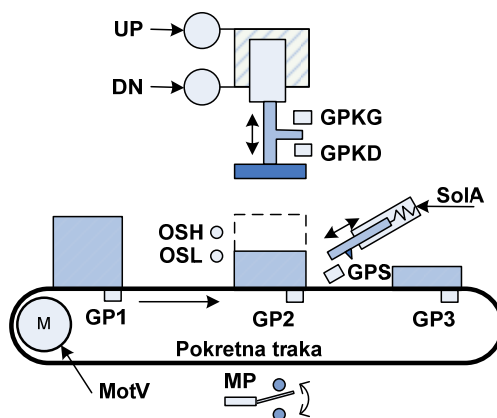
Na Sl. 4-51 je prikazana mašina za presovanje metalnih odlivaka. Elementi mašine su:

- (a) Pokretna traka i motor za pokretanje trake M (upravlja se signalom MotV);
- (b) Tri granična prekidača, GP1, GP2 i GP3, za detekciju prisustva objekta na tri karakteristične pozicije;
- (c) Dva optička senzora, OSH i OSL, za proveru visine objekta - OSL je postavljen na visini h_1 , a OSH na visini h_2 . Ako je objekat niži od h_1 , važi $OSL=OSH=OFF$; ako je objekat viši od h_1 , a niži od h_2 , važi $OSL=ON$, $OSH=OFF$ i ako je objekat viši od h_2 , važi $OSL=ON$ i $OSH=ON$.
- (d) Presa. Presom se upravlja pomoću dva jednosmerna solenoida koji otvaraju/zatvaraju ventile za dovod/ispuštanje komprimovanog vazduha. Solenoidima se upravlja signalima UP i DN. Klip prese se izvlači (spušta) ako je $DN=ON$, a uvlači (podiže) ako je $UP=ON$. Dva granična prekidača, GDKD i GPKG, detektuju krajnji donji, odnosno krajnji gornji položaj klipa.
- (e) Jednosmerni solenoida za uklanjanje objekta sa mesta za presovanje. Ovaj solenoid se pobuđuje (izvlači) ako je signal $SolA=ON$. GPS je granični prekidač koji se koristi za indikaciju krajnjeg izvučenog položaja solenoida. Nakon ukidanja pobude ($SolA=OFF$), solenoid se pod dejstvom opruge automatski vraća u uvučen položaj.
- (f) Dvopoložajnog prekidača MP za uključivanje/isključivanje mašine ($MP=ON$ - mašina je uključena; $MP=OFF$ - mašina je isključena).

Traka počinje da se kreće kada se na poziciju graničnog prekidača GP1 postavi objekat za presovanje. Kada se na poziciji za presovanje nađe objekat ($GP2=ON$), ispituje se njegova visina.

- (i) Ako je objekat niži od h_1 , traka se zaustavlja, a objekat se uklanja sa trake pomoću solenoida ($SolA$). Traka ostaje nepokretna sve dok se na poziciju GP1 ne postavi novi objekat.
- (ii) Ako je objekat viši od h_1 , ali niži od h_2 , traka se ne zaustavlja, a objekat se prenosi sve do pozicije GP3 kada traka konačno staje. Očekuje se da će objekat biti manuelno uklonjen sa pozicije GP3. Tek nakon toga, dozvoljeno je postaviti novi objekat na poziciju GP1.
- (iii) Ako je objekat viši od h_2 , započine presovanje sa ciljem da se njegova visina smanji ispod h_2 . Presovanje može da traje jedan ili više ciklusa. Ciklus presovanja podrazumeva spuštanje klipa prese do pozicije GPKD i povratak na početnu poziciju GPKG. Na kraju svakog ciklusa (kada je klip ponovo na poziciji GPKG) ispituje se visina objekata. Ako objekat još uvek viši od h_2 , presovanje se ponavlja. Ako je, kojim slučajem objekat postao niži od h_1 , postupa se kao u tački (i). Ako je visina objekta postala manja od h_2 , ali ne i manja od h_1 , zadatak je ispunjen; traka se ponovo pokreće da bi odnela objekat do pozicije GP3, gde će objekat biti manuelno uklonjen sa trake.

Prebacivanje prekidača MP u položaj OFF trenutno zaustavlja rad mašine (isključuje sve aktuatore i onemogućava pokretanje), ako se traka kreće. Međutim, ako je u toku ciklus presovanja ili sklanjanje objekta sa trake pomoću solenoida $SolA$, isključenje mašine se odlaže do momenta kada se ciklus presovanja ili sklanjanje završi.



Sl. 4-51 Mašina za presovanje.

Mašina za presovanje se odlikuje složenijim ponašanjem u odnosu na primere koje smo do sada razmatrali. Rad sistema iz većine prethodno razmatranih primera se svodio na niz uzastopnih koraka koji su se menjali jedan za drugim, a iskakanje iz definisane sekvence se dešavalo samo pod izuzetnim okolnostima (npr. aktiviran stop). Mašina za presovanje se karakteriše "razgranatijim" ponašanjem, jer se tok koraka usmerava pod dejstvom više različitih uslova. Na primer, objekat sa pozicije presovanja može odmah biti sklonjen sa pozicije za presovanje, može se obaviti jedan ili više ciklusa presovanja, a nakon presovanja objekat će biti sklonjen sa trake ili će se preneti do mesta za manuelno uklanjanje. Iako na prvi pogled složeni, problemi ovog tipa se relativno lako mogu rešiti primenom metoda konačnih automata.

Prvi korak u postupku rešavanja datog problema podrazumeva identifikaciju stanja sistema. Potpuno razumevanje problema koji se rešava je preduslov za uspešno izdvajanje stanja. Studentima se preporučuje da pre crtanja dijagrama stanja, pažljivo, i više puta pročitaju teksta zadatka i sačine listu operacija, kako bi otklonili bilo kakve nedoumice o tome šta i kako sistem treba da radi.

"Nulto" stanje je stanje u kome sistem boravi dok je isključen (dok je $MP=OFF$). Ovo stanje je ujedno i inicijalno stanje sistema (stanje u koje se sistem postavlja odmah nakon uključivanja napajanja). Stanje *Isključeno* se napušta prebacivanjem prekidača MP u položaj ON.

Prvo stanje, *ČekanjeNaObjekat* bi bilo ono dok mašina čeka da na pokretnu traku bude postavljen novi objekat za presovanje. Ovo stanje se završava kada GP1 postane ON.

Drugo stanje, *PrenosNaP2*, je stanje u kome je sistem dok se objekat sa pozicije graničnog prekidača GP1 prenosi na poziciju graničnog prekidača GP2 (pozicija P2). Za vreme ovog stanja, uključen je motor pokretne trake, a stanje se završava kada se aktivira GP2.

Podsetimo se da je stanje dobro definisano, ako u toku trajanja stanja svi izlazi sistema imaju konstantnu vrednost i ako je uslov za završetak stanja jasno definisan.

Treće stanje, *SpuštanjePrese*, odnosi se na vremenski interval kada se klip prese izvlači i deluje silom na objekat koji se presuje. U toku ovog stanja, aktivan je signal DN ($DN=ON$), a stanje se završava kada klip stigne u krajnji izvučeni položaj ($GPKD=ON$).

Četvrto stanje, *PodizanjePrese*, je ono kada se klip prese uvlači. U ovom stanju važi $UP=ON$, a uslov za kraj stanje je $GPKG=ON$.

Peto stanje *Sklanjanje*, traje dok traje sklanjanje objekta sa mesta za presovanje pomoću solenoida A. U ovom stanju važi $SolA=ON$, a stanje je završeno kada $GPS=ON$.

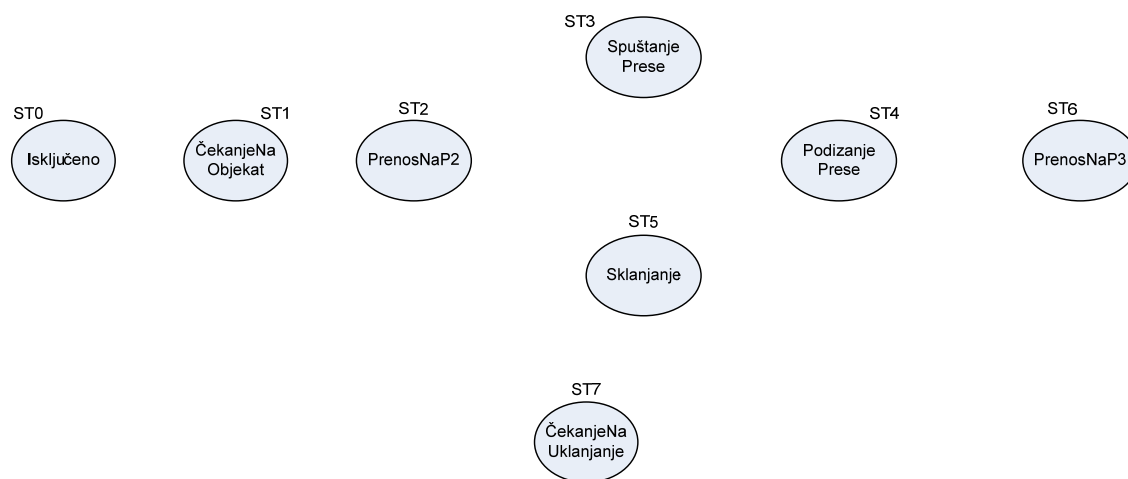
U šestom stanju, *PrenosNaP3*, vrši se prenos objekta sa pozicije P2 na poziciju P3. Motor pokretne trake je uključen ($MotV=ON$), a stanje traje sve dok se ne aktivira granični prekidač GP3.

I konačno, sedmo stanje, *ČekanjeNaUklanjanje*, je stanje u kome mašina čeka da sa pozicije P3 bude uklonjen objekat. Svi aktuatori su neaktivni, a stanje se završava kada $GP3=OFF$.

U tabeli su navedena stanja zajedno sa simboličkim imenima pridruženih bitova stanja i vrednostima izlaza u svakom stanju.

Stanje	Bit stanja	Izlazi			
		MotV	DN	UP	SolA
Isključeno	ST0	0	0	0	0
ČekanjeNaObjekat	ST1	0	0	0	0
PrenosNaP2	ST2	1	0	0	0
SpuštanjePrese	ST3	0	1	0	0
PodizanjePrese	ST4	0	0	1	0
Sklanjanje	ST5	0	0	0	1
PrenosNaP3	ST6	1	0	0	0
ČekanjeNaUklanjanje	ST7	0	0	0	0

Nakon što smo identifikovali stanja sistema, crtamo dijagram stanja. Dijagram stanja, osim stanja sadrži i prelaze između stanja. Dok stanja opisuju statičko ponašanje sistema, prelazi opisuju dinamiku sistema - pod kojim uslovima se iz jednog prelazi u neko drugo stanje. Možemo da počnemo od dijagrama koji sadrži samo stanja (Sl. 4-52), a onda da za svako stanje dodajemo prelaze koji izviru iz tog stanja.



Sl. 4-52 "Prazan" dijagram stanja mašine za presovanje.

Iz stanja Isključeno izvire samo jedna grana koja vodi sistem u stanje ČekanjeNaObjekat onda kada se sistem uključi (MP=ON).

Iz stanja ČekanjeNaObjekat postoje dva prelaza. Prvi ka stanju PrenosNaP2, čiji je uslov GP1=ON (na poziciju P1 postavljen objekat), i drugi nazad ka stanju Isključeno, pod uslovom da je sistem isključen (MP=OFF).

Za stanje PrenosNaP2 vezuju se čak četiri izlazne grane. Prva grana vraća sistem nazad na Isključeno (MP=OFF). Preostala tri prelaza uslovljena su dolaskom objekta na poziciju za presovanje (GP2=ON), a u koje sledeće stanje će sistem otići zavisi od visine objekta. Za visinu manju od h_1 (OSL=OFF), sledeće stanje je Sklanjanje, za visinu veću od h_1 i manju od h_2 (OSL=ON i OSH=OFF), sledeće stanje je PrenosNaP3, a za visinu veću od h_2 (OSH=ON), sledeće stanje je SpuštanjePrese.

Iz stanja SpuštanjePrese ne postoji povratna grana ka Isključeno, zato što je zabranjeno prekinuti započeti ciklus presovanja. Jedina grana koja izvire iz ovog stanja ogovara prelazu u stanje PodizanjePrese pod uslovom da je klip stigao u krajnji izvučen položaj (GPKD=ON).

Uslov za završetak stanja PodizanjePrese je povratak klipa u krajnji gornji položaj (GPKG=ON), a u koje sledeće stanje će sistem preći zavisi od toga da li je u međuvremenu sistem isključen (MP=OFF - povratak u Isključeno), ili, ako nije, da li je visina objekta postala niža od h_1 (OSL=OFF - za prelaz u

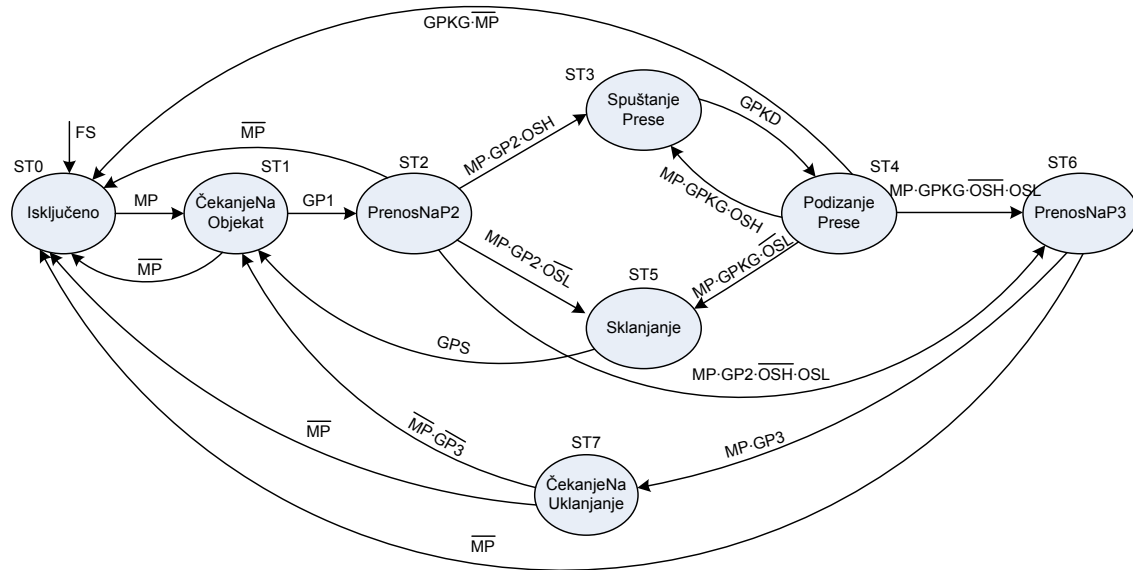
Sklanjanje), da li je u granicama između $h1$ i $h2$ (za prelaz u PrenosNaP3) ili je još uvek veća od $h2$ (za povratak na SpuštanjePrese).

Kompletan dijagram stanja prikazan je na Sl. 4-53. Uočimo da uslovi koji su pridruženi granama nisu pisani na način kao u prethodnom primeru, navođenjem vrednosti ulaza, već u obliku logičkih jednačina. Tako, MP na grani znači $MP=ON$, ili $\overline{MP} \cdot GPKG$ znači $MP=OFF$ i $GPKG=ON$. Takođe, prilikom definisanja uslova vodilo se računa o prioritetima prelaza, tako da prelaz usled isključenja sistema ima najviši prioritet.

Iz stanja Sklanjanje se izlazi kada se aktivira granični prekidač solenoida A ($GPS=ON$), što je znak da je predmet sklonjen sa pokretne trake, a novo stanje je stanje ČekanjeNaObjekat. Ako i kod stanja SpuštanjePrese, uslov $MP=OFF$ se ne ispituje.

Stanje PrenosNaP3 se završava ako se sistem isključi, ili ako se aktivira granični prekidač GP3. Prvi prelaz vraća sistemu stanje Isključeno, a drugi vodi sistem u stanje ČekanjeNaUklanjanje.

Iz stanja ČekanjeNaUklanjanje, takođe, postoje dva prelaza. Prvi ka Isključeno, a drugi (pod uslovom da je objekat uklonjen - $GP3=OFF$) ka stanju ČekanjeNaObjekat.



Sl. 4-53 Kompletan dijagram stanja mašine za presovanje.

Jednačine stanje. Označimo sa $T_{i,j}$ jednačinu za prelaz iz stanja ST_i u stanje ST_j . Jednačine $T_{i,j}$ dobijamo kada izraz napisan na grani koja spaja stanja ST_i i ST_j proširimo članom ST_i . Na primer, $T_{2,3}=ST2 \cdot MP \cdot GP2 \cdot OSH$. Jednačine stanja su:

$$ST0 = (ST0 + T_{1,0} + T_{2,0} + T_{4,0} + T_{6,0} + T_{7,0}) \cdot \overline{T_{0,1}} + FS$$

$$ST1 = (ST1 + T_{0,1} + T_{5,1} + T_{7,1}) \cdot \overline{T_{1,0}} \cdot \overline{T_{1,2}}$$

$$ST2 = (ST2 + T_{1,2}) \cdot \overline{T_{2,0}} \cdot \overline{T_{2,3}} \cdot \overline{T_{2,5}} \cdot \overline{T_{2,6}}$$

$$ST3 = (ST3 + T_{2,3} + T_{4,3}) \cdot \overline{T_{3,4}}$$

$$ST4 = (ST4 + T_{3,4}) \cdot \overline{T_{4,0}} \cdot \overline{T_{4,3}} \cdot \overline{T_{4,5}} \cdot \overline{T_{4,6}}$$

$$ST5 = (ST5 + T_{2,5} + T_{4,5}) \cdot \overline{T_{5,1}}$$

$$ST6 = (ST6 + T_{4,6} + T_{2,6}) \cdot \overline{T_{6,0}} \cdot \overline{T_{6,7}}$$

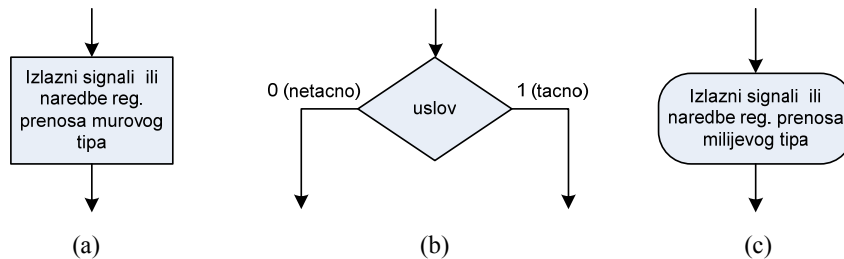
$$ST7 = (ST7 + T_{6,7}) \cdot \overline{T_{7,0}} \cdot \overline{T_{7,1}}$$

Jedan način kako se može kreirati leder dijagram jeste da se uvedu bitovi prelaza koji će se postavljati na osnovu jednačina prelaza, a da se potom bitovi prelaza koriste za realizaciju rangova koji postavljaju bitove stanja. Dugi način je da se u jednačine stanja zamene jednačine prelaza i izvrši pojednostavljenje dobijenih izraza. Broj rangova i utrošenih pomoćnih bitova će biti manji, ali će zato rangovi "stanja" biti nešto složeniji.

Treba primetiti da će se pojedine jednačine stanja zanačajno uprostiti nakon zamene jednačina prelaza. Na primer, razmotrimo stanje PrenosNaP2 (ST2). Uslov za napuštanje ovog stanja je MP prebačen u položaj OFF ili aktiviran granični prekidač GP2, tj. $\overline{MP} + \overline{GP2}$. To znači da će deo jednačine stanja ST2, $\overline{T_{2,0}}\overline{T_{2,3}}\overline{T_{2,5}}\overline{T_{2,6}}$, da se svede na $\overline{ST2} + MP \cdot GP2$. Slično zapažanje važi i za stanje ST4, kod koje se deo jednačine za napuštanje stanja svodi na: $\overline{ST4} + MP \cdot GPKG$

4.6.2 ASM dijagrami

Dijagrami stanja su pogodno sredstvo za predstavljanje konačnih automata sa malim brojem ulaza i izlaza. Međutim, za opis složenijih automata, projektanti obično koriste tzv. *algoritamske mašine stanja*, ili ASM dijagrame (prema eng. *Algorithmic State Machine*). U suštini, ASM dijagrami su vrsta dijagrama toka prilagođenih za predstavljanje stanja, prelaza između stanja i generisanje izlaza konačnog automata. Na Sl. 4-54 su predstavljena tri osnovna grafička simbola koji se koriste za crtanje ASM dijagrama: blok stanja, blok grananja i blok uslovnog izlaza.



Sl. 4-54 Grafički simboli za crtanje ASM dijagrama: (a) blok stanja; (b) blok grananja; (c) blok uslovnog izlaza.

Blok stanja se crta u vidu pravougaonika i služi za predstavljanje jednog stanja konačnog automata. Ekvivalentan je krugu u dijagramu stanja. Ime stanja piše se izvan bloka, obično uz njegov gornji levi ugao. Unutar bloka navedeni su izlazi koji su aktivni (postavljeni) u konkretnom stanju. Za preostale signale se podrazumeva da su neaktivni. Na primer, da bi se naglasilo da u datom stanju izlaz z ima vrednost 1, dovoljno je napisati z umesto $z=1$.

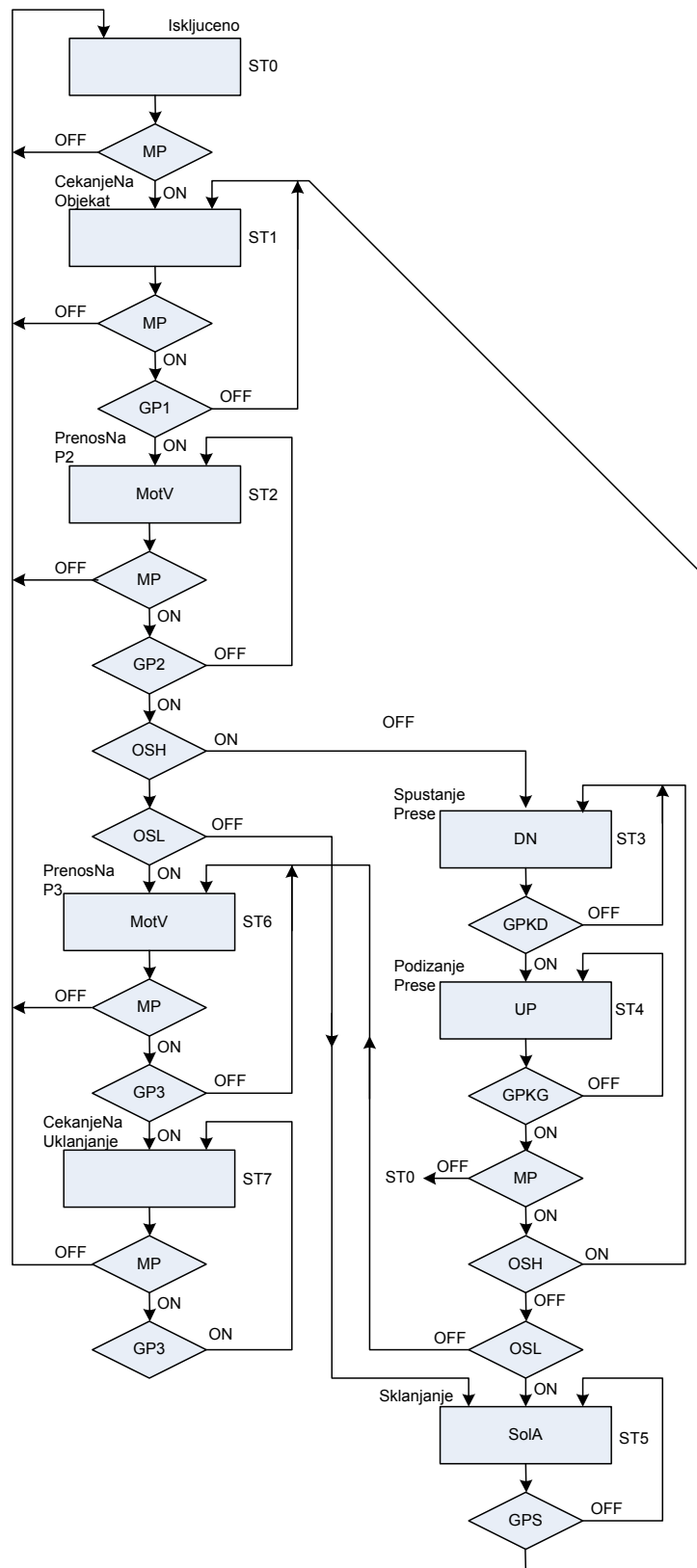
Blok grananja se crta kao romb koji sadrži uslov koji određuje koja će od dve izlazne grane biti izabrana. Uslov kombinuje jedan ili više ulaza konačnog automata u logički izraz čija vrednost može biti tačno (1) ili netačno (0). Na primer, w napisano u bloku grananja znači da je odluka zasnovana na vrednosti signala w (ako je $w=0$, nastavljamo putem *tačno*, a ako je $w=1$, putem *netacno*). Slično, uslov w_1w_2 znači da će biti grana *tačno* biti izabrana ako su oba signala jednaka 1, tj. $w_1=w_2=1$, a da će u svim ostalim slučajevima automat slediti granu *netacno*.

Pr. 4-3 ASM dijagram mašine za presovanje

ASM dijagram mašine za presovanje prikazan je na Sl. 4-55. ASM dijagram je često preglednija predstava rada sistema od konačnog automata. Prvo, ASM dijagram sadrži informaciju o vrednostima izlaza u svakom stanju. Drugo, prioriteti prelaza su ugrađeni u sam dijagram, tj. određeni su redosledom u kome su blokovi grananja poređani ispod bloka stanja. Razmotrimo, na primer stanje ST2. Najviši prioritet ima prelaz u stanje isključeno (ako je $MP=OFF$); zatim, ako je $MP=ON$, ispituje se granični prekidač GP2, a tek onda, ako je $GP2=ON$, dalje se ispituju optički senzori OSH i OSL.

Takođe, jednačine prelaza se mogu lako izvesti. Dovoljno je pratiti putanju od izvornog do određištog stanja kroz blokove grananja. Na primer, da bi sistem prešao iz stanja ST4 u stanje ST6 potrebno je da važi: $ST4=ON$ (da je sistem u stanju ST4), $GPKG=ON$, $OSH=OFF$ i $OSL=ON$, tj. jednačina prelaza iz stanja ST4 u ST6 glasi: $ST4 \cdot GPKG \cdot \overline{OSH} \cdot OSL$.

Postupak realizacije leder dijagrama na osnovu ASM dijagrama je identičan postupku koji se koristi kada se konačni automat opisuje dijagramom stanja.



Sl. 4-55 ASM dijagram mašine za presovanje.

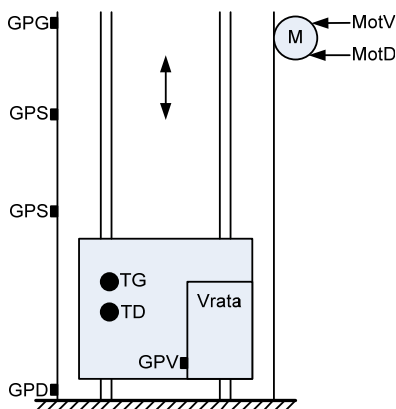
Pr. 4-4 Upravljačka jedinica lifta

Lift se podiže/spušta pomoću dvosmernog motora (Sl. 4-56). Motorom se upravlja pomoću dva signala: MotV i MotD. Za MotV=ON motor je uključen, za MotV=OFF isključen. MotD određuje smer rotacije motora, taka da se za MotD=ON motor spušta a za MotD=OFF motor podiže.

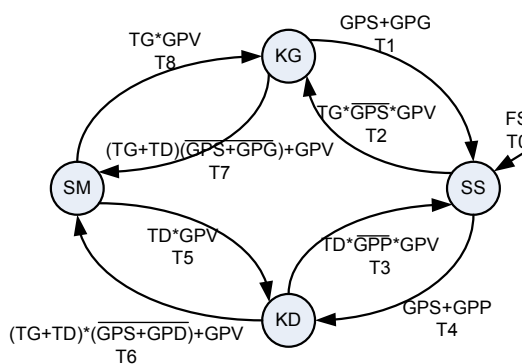
Lift se može spustiti/podići pomoću taster u liftu TG/TD. Pritisak na bilo koji od tastera ukoliko se lift kreće zaustavlja lift.

Granični prekidači GG i GD služe za indicaciju krajnjeg gornjeg i krajnjeg donjeg položaja lifta. Na međuspratovima postoje prekidači GPS koji detektuju prolazak lifta.

Lift se ne kreće ukoliko vrata nisu zatvorena, što se indukuje pomoću prekidača GPV (GPV=ONN - zatvorena) Posle pokretanja, lift se kreće samo do narednog sprata i zaustavlja se posle čega je za dalje kretanje potrebno ponovo pritisnuti taster u liftu.



Sl. 4-56 Lift iz Pr. 4-4



Sl. 4-57 Konačni automat za Pr. 4-4

Rešenje:

Uočena stanja su:

- KG - kretanje na gore
- KD - kretanje na dole
- SS - stoji na spratu
- SM - stoji na međuspratu

Jednačine prelaza su:

$$\begin{aligned}
 T1 &= KG \cdot (GPS + GP) \\
 T2 &= SS \cdot (TG \cdot \overline{GPS} \cdot GPV) \\
 T3 &= KD \cdot TD \cdot \overline{GPP} \cdot GPV \\
 T4 &= SS \cdot (GPS + GPP) \\
 T5 &= SM \cdot TD \cdot GPV \\
 T6 &= KD \cdot ((TG + TD) \cdot (\overline{GPS + GPD}) + GPV) \\
 T7 &= KG \cdot ((TG + TD) \cdot (\overline{GPS + GPG}) + GPV) \\
 T8 &= SM \cdot TG \cdot GPV
 \end{aligned}$$

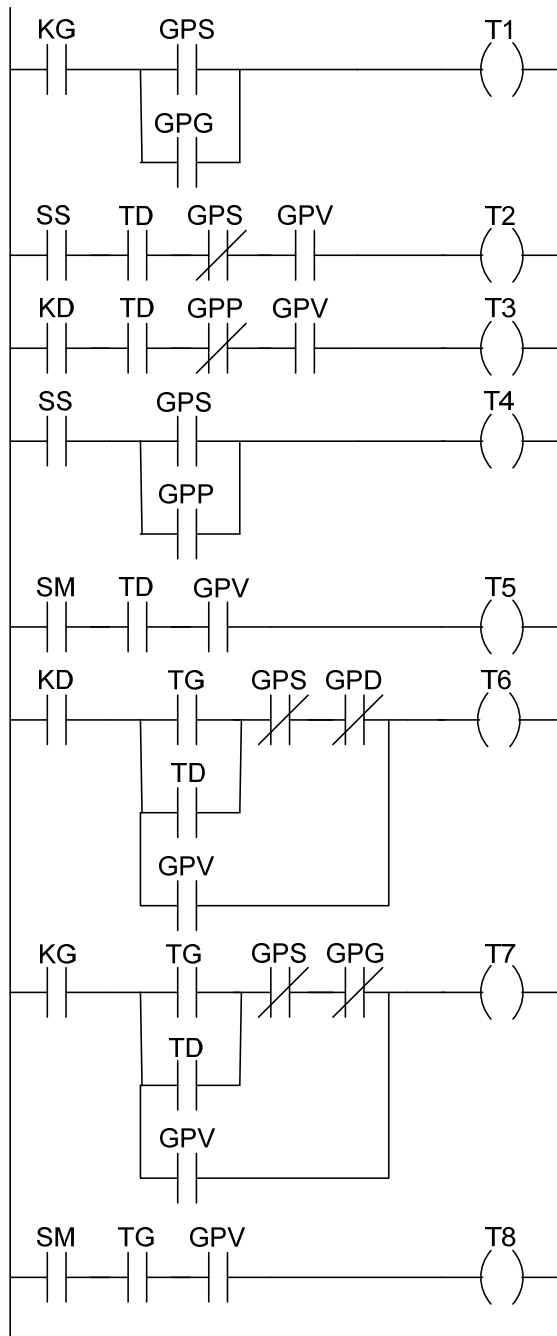
Jednačine stanja:

$$\begin{aligned}
 SS &= (SS + FS + T1 + T3) \cdot \overline{T2} \cdot \overline{T4} \\
 KD &= (KD + T4 + T5) \cdot \overline{T3} \cdot \overline{T6} \\
 SM &= (SM + T7 + T5) \cdot \overline{T5} \cdot \overline{T8} \\
 KG &= (KG + T2 + T8) \cdot \overline{T7} \cdot \overline{T1}
 \end{aligned}$$

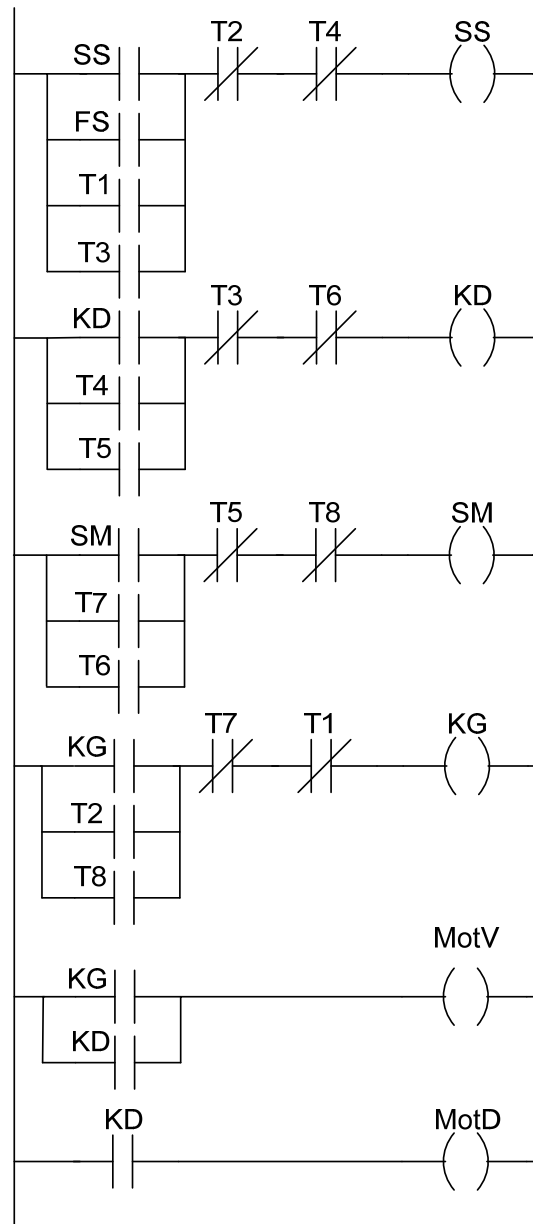
Postavljanje izlaza:

$$\begin{aligned}
 MotV &= KG + KD \\
 MotD &= KD
 \end{aligned}$$

Na Sl. 4-58 i Sl. 4-59 je prikazan leder dijagram za automat.



Sl. 4-58 Leder dijagram za Pr. 4-4



Sl. 4-59 Nastavak leder dijagrama za Pr. 4-4

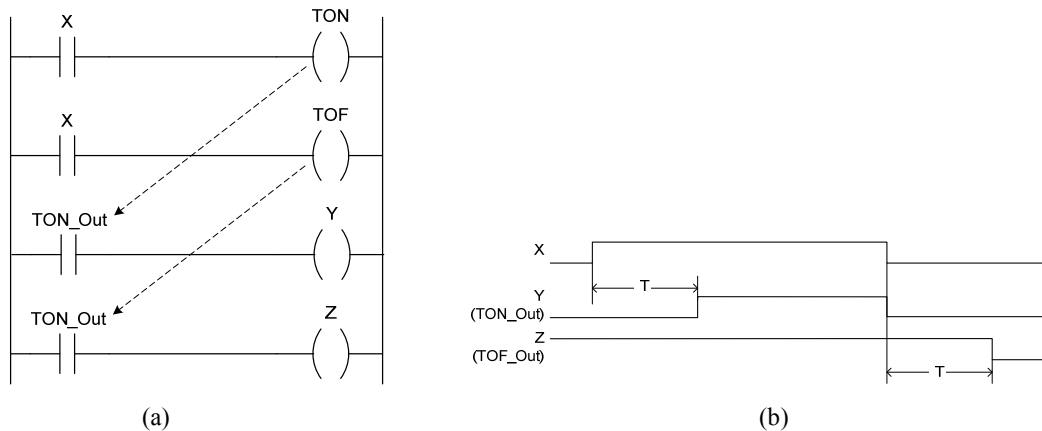
5 Naredbe za merenje vremena i prebrojavanje događaja – tajmeri i brojači

Prilikom upravljanja ili nadzora procesa često je potrebno da se neka aktivnost otpočne ili zaustavi posle određenog vremenskog perioda, ili da se ponovi određeni broj puta. U tom smislu neophodno je da PLC kontroler koji će se koristiti za upravljanje procesom pruži mogućnost za merenje vremena i prebrojavanje događaja. Prebrojavanje događaja obavlja **brojač (counter)**, koji nakon registrovanja unapred zadanog broja događaja generiše odgovarajući signal. Merenje vremena ostvaruje se pomoću **časovnika, tj. tajmera (timer)**. U suštini tajmer izražava vreme kao umnožak određenog osnovnog intervala (*vremenska baza*). To zapravo znači da tajmer radi kao brojač protoka osnovnih intervala i da nakon isteka određenog, unapred zadanog intervala vremena, generiše odgovarajući signal.

5.1 Naredbe za merenje vremena

PLC naredbe za merenje vremena su, posle bit naredbi, najčešće korišćene naredbe u leder programiranju. Podrška za programsko merenje vremena, u vidu specijalizovanih, tzv. tajmerskih naredbi, postoji kod svih danas raspoloživih familija PLC kontrolera. Iako način rada i mogućnosti ovih naredbi zavise od tipa PLC kontrolera, uopšteno govoreći, u leder programiranju se koriste tri vrste tajmera: *delay_on* tajmer (ili TON), *delay_off* tajmer (ili TOF) i RTO tajmer.

Tajmeri su naredbe akcija. Uslov koji prethodi tajmeru upravlja tajmerom tako što mu omogućava ili brani rad. Tajmer, za vreme dok je omogućen, odbrojava osnovne vremenske intervale i , nakon dostizanja zadate vrednosti, postavlja određeni bit (ili bitove) iz interne memorije PLC kontrolera, koji se mogu koristiti u drugim rangovima za formiranje uslova. Sl. 5-1 ilustruje razliku između *delay-on* i *delay-off* tajmera. TON je omogućen dok je uslov tačan, a TOF dok je uslov netačan. TON se isključuje trenutno, a uključuje po isteku zadatog vremena, T . Nasuprot tome, TOF se uključuje trenutno, a isključuje po isteku zadatog vremena, T . (Ovde se pod uključivanjem/isključivanjem smatra setovanje/resetovanje bita koji tajmer kontroliše – TON_Out, odnosno TOF_Out). Drugim rečima, TON kasni uključivanje, a TOF isključivanje. Na primer, TON se može iskoristiti da odloži početak nekog proizvodnog procesa za vreme potrebno da se komora zagreje; TOF se može iskoristiti da ventilator, koji hladi komoru, ostane uključen neko zadato vreme nakon što je sistem isključen.



Sl. 5-1 Delay-on i delay-off tajmer: (a) leder dijagram; (b) vremenski dijagram.

TON i TOF tajmeri uvek počinju svoj rad "od nule", tj. promena uslova na vrednost koja brani rad tajmera ujedno i resetuje tajmer tako da kada sledeći put tajmer bude pušten u rad, njegova početna vrednost biće 0. Za razliku od toga, RTO tajmer sumira (akumulira) vreme za koje je omogućen. Kada je ponovo omogućen, RTO nastavlja odbrojavanje osnovnih vremenskih intervala počev od vrednosti koju je imao u trenutku kada je zaustavljen. Za resetovanje RTO tajmera koristi se posebna naredba.

5.1.1 Parametri tajmera

U seriji kontrolera SLC 5 tajmeri i brojači su realizovani softverski, i koriste se kao naredbe *akcije*. Ne postoji nikakvo posebno ograničenje u pogledu njihovog broja. Jedino ograničenje potiče od veličine raspoložive RAM memorije.

Pri korišćenju tajmera neophodno je da se definišu sledeći parametri.

- **Vremenska baza (*time base*)** određuje dužinu *osnovnog intervala vremena*. Kod kontrolera SLC 5 bira se jedna od dve moguće vrednosti: 0.01 s ili 1.0 s.
- **Zadata vrednost (*preset value - PRE*)** je vrednost kojom se definiše željeni broj osnovnog intervala vremena (čime se određuje ukupno vreme koje časovnik treba da izmeri) koje tajmer treba da registruje pre nego što se generiše signal koji označava da je tajmer završio.

Zadata vrednost za časovnik može da se kreće u intervalu od 0 do +32767, što odgovara vremenu od maksimalno 327.67 s (za vremensku bazu 0.01 s), odnosno maksimalno 32767 s, ili približno 9h i 6min (za vremensku baz 1 s).

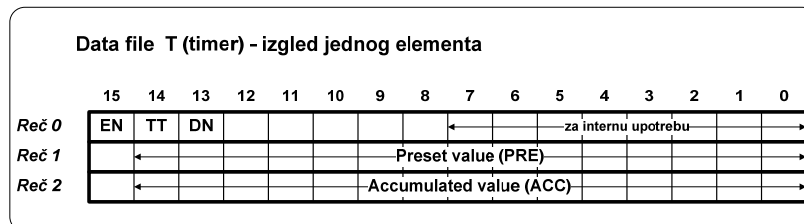
- **Akumulirana vrednost (*accumulated value - ACC*)** predstavlja broj osnovnih vremenskih intervala koje je časovnik izbrojao u nekom trenutku. Kada akumulirana vrednost postane veća ili jednaka od zadate vrednosti časovnik, odnosno brojač, završavaju svoj rad.

5.1.2 Datoteka podataka o tajmerima (timer data file)

S obzirom da je tajmer realizovan softverski, parametri koji definišu njegov rad moraju biti smešteni u memoriji kontrolera. Za pamćenje podataka o tajmerima koristi se datoteka podataka broj 4 (*timer file - T*). U ovoj datoteci može se definisati najviše 256 različitih časovnika. Ukoliko je potrebno da se koristi veći broj tajmera, korisnik može definisati i dodatne datoteke (*korisnički definisane datoteke*) čiji su brojevi od 9 do 255.

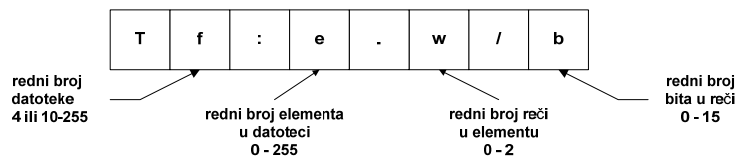
Svakom tajmeru pridružuju se po jedan *element* u odgovarajućoj datoteci. *Osnovni element* ovih datoteka sastoji se od tri 16-bitne reči:

- **Reč 0** je kontrolna reč koja sadrži tri bita koja ukazuju na stanje časovnika, kao i bitove za interno upravljanje radom časovnika
- **Reč 1** sadrži *zadatu vrednost (PRE)*
- **Reč 2** sadrži akumuliranu vrednost (ACC)



Adresibilni bitovi
 EN = bit 15 - enable
 TT = bit 14 - timer timing
 DN = bit 13 - done

Adresibilne reči
 PRE - preset value (zadana vrednost)
 ACC - accumulated value (akumulirana vrednost)



Sl. 5-2 Element datoteke tajmera i adresiranje tajmera.

Na Sl. 5-2 prikazan je jedan element datoteke tajmera, kojim se definiše jedan tajmer. Pored toga, prikazan je format adrese tajmera. Treba zapazi da *broj elementa* definiše jedan određeni tajmer unutar jedne datoteke tajmera. Svaki od tri bita stanja, kao i zadata i akumulirana vrednost mogu se

posebno adresirati i to bilo na standardan način na koji se formira adresa u bilo kojoj datoteci podataka, bilo preko odgovarajućih simbola. To zapravo znači da su sledeće adrese međusobno ekvivalentne:

Tf:e.1 ili Tf:e.PRE
 Tf:e.2 ili Tf:e.ACC
 Tf:e.0/15 ili Tf:e/15 ili Tf:e/EN
 Tf:e.0/14 ili Tf:e/14 ili Tf:e/TT
 Tf:e.0/13 ili Tf:e/13 ili Tf:e/DN

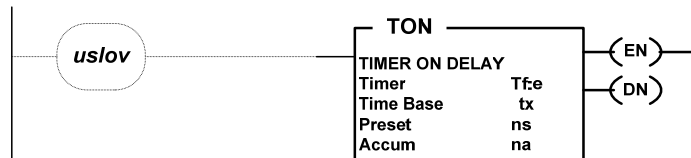
5.1.3 Naredbe časovnika

Kao što je već rečeno naredbe tajmera su naredbe *akcije*, što znači da se nalaze na desnoj strani ranga u leder programu. Postoje tri tipa naredbi kojima se realizuju tri vrste tajmera, i jedna naredba kojom se stanje tajmera resetuje.

Potrebno je istaći da se sam tajmer i način njegovog rada definiše preko naredbe koja se uvrštava u leder program. Drugim rečima, kad se u program stavi jedna od moguće tri naredbe i u njoj naznači adresa tajmera u odgovarajućem formatu, onda operativni sistem PLC kontrolera sam zauzme naznačeni element (tri reči) u datoteci koja je navedena u adresi i popuni ga odgovarajućim sadržajem.

- **Timer on-delay (TON)**

TON naredba, grafički simbol i položaj u rangu



Kao što je već rečeno, stavljanjem ove naredbe u leder program automatski se definiše prva vrsta tajmera i zauzimaju tri reči koje čine element *e* u datoteci časovnika *f*. Prilikom formiranja naredbe specifikaraju se i vremenska baza (tx) i zadata vrednost (ns).

TON naredba započinje rad tajmera (prebrojavanje osnovnih vremenskih intervala) za vreme onog programskog sken ciklusa u kome *uslov* u rangu u kome se naredba nalazi prvi put postaje *istinit* (prelaz *neistini/istinit* – uzlazna ivica). U svakom sledećem sken ciklusu, sve dok je *uslov istinit* časovnik vrši ažuriranje akumulirane vrednosti (ACC) u skladu sa proteklom vremenom između dva ciklusa. Kada akumulirana vrednost dostigne zadatu vrednost, tajmer prekida svoj rad i postavlja DN bit na 1. Pri tome, ako u nekom sken ciklusu *uslov* postane *neistinit*, tajmer prekida svoj rad i akumulirana vrednost se postavlja na 0, bez obzira da li je tajmer pre toga izmerio zahtevano vreme ili ne.

Bitovi stanja časovnika menjaju se u toku programskog sken ciklusa na sledeći način:

DN - Timer done bit se postavlja na 1 kada je $ACC \geq PRE$. On se resetuje na 0 kad *uslov* u rangu postane *neistinit*.

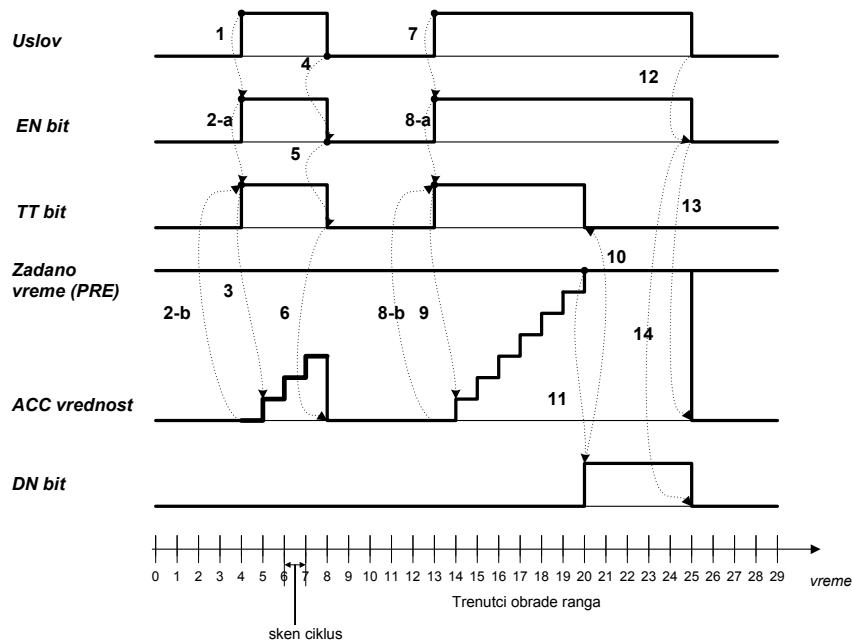
EN - Timer enable bit se postavlja na 1 kada je *uslov* u rangu *istinit* i resetuje na 0 kada *uslov* postane *neistinit*.

TT - Timer timing bit se postavlja na 1 kada je *uslov istinit* i ako je $ACC \leq PRE$. On se resetuje na 0 kada *uslov* postane *neistinit* ili kada se DN bit postavi na 1, odnosno kada se završi merenje vremena.

U vezi sa radom tajmera potrebno zapaziti nekoliko činjenica. Pre svega **tajmer radi samo dok je uslov istinit** (signal na ulazu u časovnik je u stanju “on”). Istinitosnu vrednost uslova pokazuje bit EN. Drugim rečima, ovaj bit ima vrednost 1 onda kada je *uslov istinit* i to označava da je rad tajmera *omogućen (enable)*. Kada je *uslov neistinit*, EN bit ima vrednost 0, što znači da je rad tajmera *onemogućen*. Međutim, činjenica da EN bit ima vrednost 1 ne mora da znači da tajmer zaista i radi,

jer je on mogao i da završi rad zbog isteka zadatog vremena, a da pri tome *uslov* i nadalje ostane *istinit*. **Rad tajmera** indicira TT bit. Naime, taj bit je postavljen na 1 za sve vreme za koje *tajmer aktivno meri vreme* (*timer timing*), i postavlja se na 0 kada tajmer ne radi. Konačno, kada je vrednost DN bita 1, onda to znači da je tajmer *završio* (*done*) svoj posao, tj. izmerio zadato vreme. Pri tome, DN bit ne govori o tome **kada** je tajmer završio sa poslom, jer će on ostati na vrednosti 1 sve dok *uslov* ne postane *neistinit*. Vremenski dijagram rada tajmera ilustrovan je na Sl. 5-3.

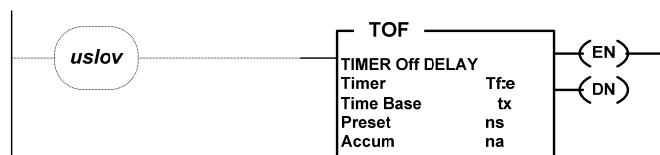
Stanje tajmera se može resetovati posebnom *RES naredbom*, o čemu će kasnije biti više reči.



Sl. 5-3 Vremenski dijagram izvršavanja TON naredbe.

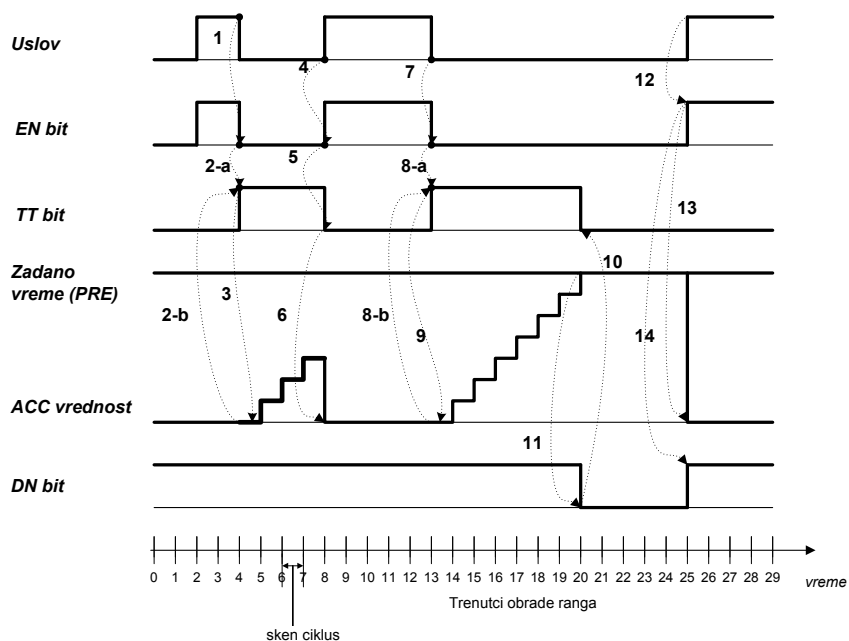
- **Timer off-delay (TOF)**

TOF naredba, grafički simbol i položaj u rangu



Ovom naredbom se definiše druga vrsta tajmera i zauzimaju tri reči koje čine elemenat *e* u datoteci tajmera *f*. Prilikom formiranja naredbe specificiraju se i vremenska baza (*tx*) i zadata vrednost (*ns*). Akumulirana vrednost se automatski postavlja na 0.

TOF naredba započinje rad tajmera za vreme onog programskog sken ciklusa u kome uslov u rangu u kome se naredba nalazi prvi put postaje *neistinit* (prelaz istini/neistinit – silazna ivica). U svakom sledećem sken ciklusu, sve dok je uslov *neistinit* časovnik vrši ažuriranje akumulirane vrednosti (ACC) u skladu sa proteklom vremenom između dva ciklusa. Kada akumulirana vrednost dostigne zadatu vrednost, tajmer prekida svoj rad. Pri tome, ako u nekom sken ciklusu uslov postane *istinit*, tajmer prekida svoj rad i akumulirana vrednost se postavlja na 0, bez obzira da li je tajmer pre toga izmerio zahtevano vreme ili ne.



Sl. 5-4 Vremenski dijagram izvršavanja TOF naredbe

Bitovi stanja TOF tajmera menjaju se u toku programskog sken ciklusa na sledeći način:

DN - Timer done bit se postavlja na 1 kada je uslov istinit. On se resetuje na 0 kada je uslov neistinit i pri tome je $ACC \geq PRE$.

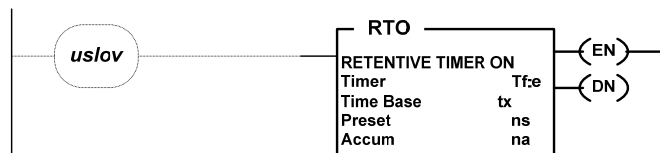
EN - Timer enable bit se postavlja na 1 kada je uslov istinit, i resetuje na 0 kada je uslov neistinit.

TT - Timer timing bit se postavlja na 1 kada je uslov neistinit i pri tome je $ACC \leq PRE$. One se resetuje na nulu kada uslov postane istinit ili kada se DN bit resetuje.

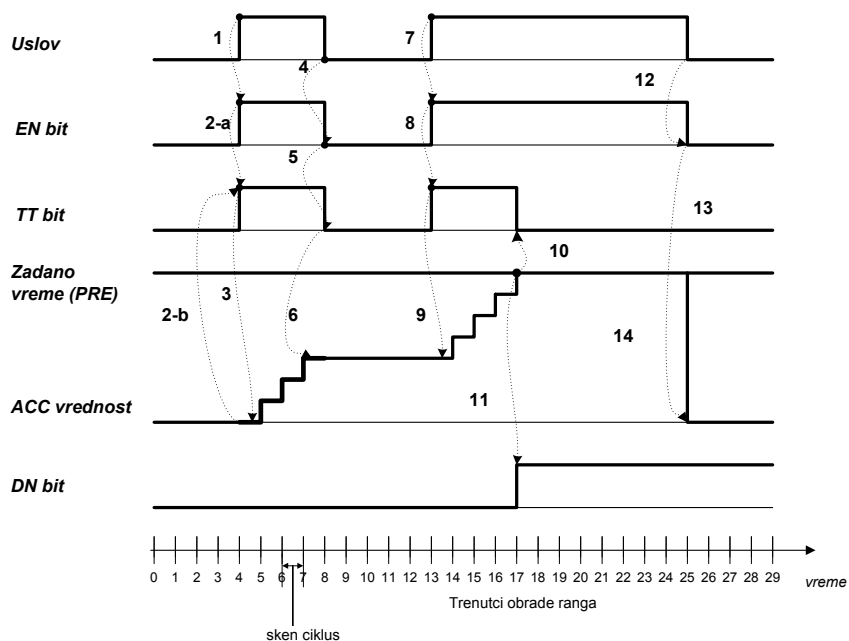
U vezi sa radom TOF tajmera potrebno je da se zapazi nekoliko činjenica. Pre svega tajmer radi samo dok je uslov neistinit (signal na ulazu u tajmer je u stanju "off"). Istinitosnu vrednost uslova pokazuje EN, ali za razliku od TON naredbe, ovde on onemogućava rad tajmera. Drugim rečima, ovaj bit ima vrednost 1 onda kada je uslov istinit i to označava da je rad tajmera onemogućen. Kada je uslov neistinit, EN bit ima vrednost 0, što znači da je rad tajmera omogućen. Međutim, činjenica da EN bit ima vrednost 0 ne mora da znači da tajmer zaista i radi, jer je on mogao i da završi rad zbog isteka zadanog vremena, a da pri tome uslov i nadalje ostane neistinit. Rad tajmera indicira TT bit. Naime, taj bit je postavljen na 1 za svo vreme za koje tajmer aktivno meri vreme (timer timing), i postavlja se na 0 kada tajmer ne radi. Konačno, kada je vrednost DN bita 0, onda to znači da je tajmer završio (done) svoj posao, tj. izmerio zadato vreme. Pri tome, DN bit ne govori o tome kada je časovnik završio sa poslom, jer će on ostati na vrednosti 0 sve dok uslov ne postane istinit. Vremenski dijagram rada časovnika ilustrovan je na Sl. 5-4.

- **Retentive Timer (RTO)**

RTO naredba, grafički simbol i položaj u rangu



Ovom naredbom se definiše treća vrsta tajmera i zauzimaju tri reči koje čine element e u datoteci časovnika f . Prilikom formiranja naredbe specificiraju se i vremenska baza (tx) i zadata vrednost (ns). Akumulirana vrednost se automatski postavlja na 0.



Sl. 5-5 Vremenski dijagram izvršavanja RTO naredbe

RTO naredba se razlikuje od TON naredbe samo po tome što se akumulirana vrednost ne resetuje, već zadržava i onda kada uslov postane neistinit. Drugim rečima, ovaj tajmer počinje da radi kada uslov postane istinit, i nastavlja sa radom povećavajući akumuliranu vrednost sve dok je uslov istinit. Kada uslov postane neistinit, tajmer prekida rad, ali se akumulirana vrednost pri tome ne menja. To znači da će kada uslov ponovo postane istinit, tajmer nastaviti sa radom i prethodno izmerenom vremenu (ACC) dodavati nove vrednosti. Na taj način ovaj tajmer omogućuje da se kumulativno mere intervali vremena u kojima je uslov bio istinit (Sl. 5-5).

Bitovi stanja RTO tajmera menjaju se u toku programskog sken ciklusa na sledeći način:

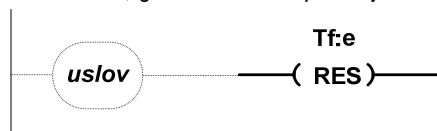
DN - Timer done bit se postavlja na 1 kada je $ACC \geq PRE$ (časovnik je izmerio zadato vreme). On se resetuje na 0 pomoću posebne RES naredbe.

EN - Timer enable bit se postavlja na 1 kada je uslov u rangu istinit (rad časovnika je omogućen) i resetuje na 0 kada uslov postane neistinit (rad časovnika je onemogućen).

TT - Timer timing bit se postavlja na 1 kada je uslov istinit i ako je $ACC \leq PRE$ (časovnik radi). On se resetuje na 0 kada uslov postane neistinit ili kada se DN bit postavi na 1 (časovnik prestaje sa radom).

- **Reset naredba (RES)**

RES naredba, grafički simbol i položaj u rangu



RES naredba je naredba akcije i koristi se za resetovanje tajmera. Kada je uslov istinit ova naredba se izvršava tako što se u tajmeru čija je adresa navedena u RES naredbi, resetuju na nulu bitovi DN, TT i EN, kao i akumulirana vrednost (ACC). S obzirom na način rada očigledno je da se RES naredbe ne sme koristiti za TOF tip časovnika.

5.1.3.1 Način rada tajmera

Sve dok časovnik radi, u svakom sken ciklusu povećava se akumulirana vrednost. Pri tome, iznos za koji će se povećati ACC vrednost zavisi od dužine trajanja sken ciklusa. Naime, kada se prilikom

obrade ranga ustanovi da su se stekli uslovi da časovnik počne sa radom onda se istovremeno startuje jedan interni časovnik, koji se ažurira preko prekida (interapta) na svakih 0,01 sec. Broj registrovanih vremenskih intervala se smešta u interni 8-bitni registar (bitovi 0-7 u prvoj reči). Ukoliko je u pitanju časovnik čija je vremenska baza 0,01 sec, onda se u sledećem programskom skenu, kada se naiđe na dati rang, vrednost internog registra, koja zapravo predstavlja interval vremena koji je protekao između dva sukcesivna sken-a, dodaje akumuliranoj vrednosti. Nakon toga se interni rgistar resetuje na 0 i počinje ponovo da meri vreme do sledećeg skena. Budući da je maksimalna vrednost koju može da ima interni registar oko 2,5 sec (255x0,01), može se očekivati da će tajmer raditi ispravno samo ako sken ciklus ne traje duže od 2,5 sekundi. Ukoliko se tajmer koristi u programu čiji sken ciklus traje duže, onda je neophodno da se ista naredba za časovnik postavi na više mesta u programu čime će se obezbediti da se rangovi koji sadrže taj časovnik obrađuju sa učestanošću koja nije veća od 2,5 sekundi.

Ukoliko časovnik radi sa vremenskom bazom od 1 sekunde obrada časovnika je donekle složenija. Ovde se, naime i dalje koristi interni časovnik koji se ažurira na svakih 0,01 sekundi, ali se pri tome u toku obrade ranga akumulirana vrednost ažurira samo ako je akumulirana vrednost veća ili jednaka od 1 sekunde. Pri tome se akumulirana vrednost uvećava za 1, dok se eventualni ostatak vremena pamti u internom brojaču i na njega se dodaju sledeći inkrementi od po 0,01 sekunde. Postupak ažuriranja akumulirane vrednosti je takav da se može očekivati da će časovnik raditi ispravno ako sken ciklus ne traje duže od 1,5 sekundi. Naravno, i ovde se problem ciklusa dužeg trajanja može prevazići stavljanjem naredbe časovnika na više mesta u programu.

Potrebno je da se naglasi da je pri korišćenju časovnika neophodno da se posebna pažnja posveti naredbama za skok. Naime, i ako je trajanje sken ciklusa u dozvoljenim granicama, može se desiti da se nekom od naredbi za skok u jednom ili više sukcesivnih sken ciklusa preskoči rang koji sadrži časovnik. Jasno je da se u tom slučaju neće vršiti ažuriranje akumulirane vrednosti. To nadalje znači da je neophodno da se obezbedi da u slučaju bilo kakvog programskog skoka, naredba za časovnik ne bude isključena iz obrade u periodu koji je duži od maksimalno dozvoljenog vremena.

5.2 Naredbe za prebrojavanje događaja

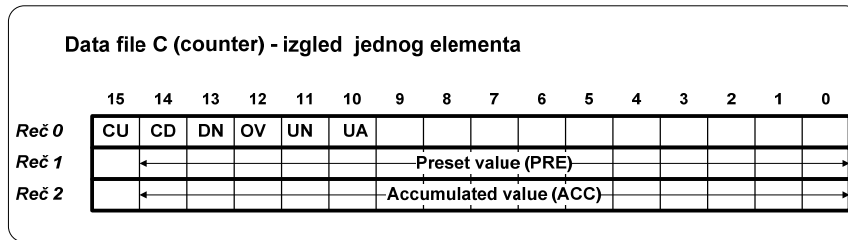
5.2.1 Parametri brojača

Svakom brojaču pridružena su sledeća dva parametra:

- **Zadata vrednost (*preset value* - PRE)** je vrednost kojom se definiše ukupni broj događaja koje brojač treba da registruje pre nego što se generiše signal koji označava da je brojač završio rad. Zadata vrednost za brojač može da se kreće u intervalu od 0 do +32767.
- **Akumulirana vrednost (*accumulated value* - ACC)** predstavlja broj događaja koje je brojač registrovao u nekom trenutku. Kada akumulirana vrednost postane veća ili jednaka od zadate vrednosti brojač, završavaju svoj rad – više ne raguje na događaje.

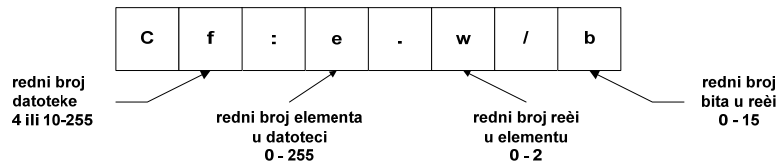
5.2.2 Datoteka podataka o brojaču (*counter data file*)

Budući da je brojač, isto kao i časovnik, realizovan softverski, parametri koji definišu njegov rad moraju biti smešteni u memoriji kontrolera. Za pamćenje podataka o brojačima koristi se datoteka podataka broj 5 (*counter file* – C). U ovoj datoteci može se definisati najviše 256 različitih brojača. Ukoliko je potrebno da se koristi veći broj brojača, korisnik može definisati i dodatne datoteke (*korisnički definisane datoteke*) čiji su brojevi od 9 do 255.



Adresibilni bitovi
 CU = bit 15 - counter up enable
 CD = bit 14 - counter down enable
 DN = bit 13 - done
 OV = bit 12 - overflow
 UN = bit 11 - underflow
 UA = bit 10 - update accumulated (HSC only)

Adresibilne reči
 PRE - preset value (zadata vrednost)
 ACC - accumulated value (akumulirana vrednost)



Sl. 5-6 Element datoteke brojača i adresiranje brojača.

Svakom brojaču pridružuju se po jedan *element* u odgovarajućoj datoteci (Sl. 5-6). *Osnovni element* ovih datoteka sastoji se od tri 16-bitne reči:

- **Reč 0** je kontrolna reč koja sadrži 6 bitova koji ukazuju na stanje brojača.
- **Reč 1** sadrži *zadatu vrednost* (PRE)
- **Reč 2** sadrži akumuliranu vrednost (ACC)

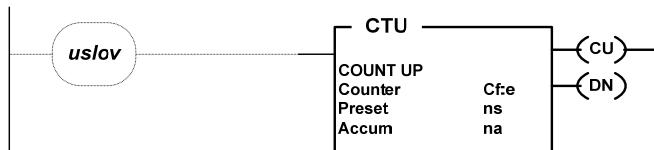
5.2.3 Naredbe brojača

Postoje dva osnovna tipa brojača *brojač unapred* (CTU – *count up*) i *brojač unazad* (CTD – *count down*). Obe naredbe su naredbe *akcije*, što znači da se smeštaju u desni deo ranga. Oba brojača broje promene vrednosti *uslova sa neistinit na isitinit* (uzlazna ivica). Pri svim ostalim vrednostima *uslova*, oni zadržavaju prebrojani iznos i čekaju sledeći prelaz. Drugim rečima, brojači se niti puštaju u rad, niti zaustavljaju. Oni neprekidno rade i beleže (broje) svaki prelaz *istinit/neistinit*. Dostizanje zadate vrednosti se signalizira postavljanjem odgovarajućeg bita – *done bit* (DN) – na 1, ali se brojanje i dalje nastavlja. Prebrojani iznos se može izbrisati jedino posebnom *RES* naredbom.

Jedina razlika između dva tipa brojača sastoji se u tome što prvi (CTU) broji unapred od 0 do 32767, i postavlja *overflow bit* (OV) na 1 kad pređe 32767, dok drugi (CTD) broji unazad, od 0 do –32767, i postavlja *underflow bit* (UN) kad pređe –32767.

- **Count up (CTU)**

CTU naredba, grafički simbol i položaj u rangu



Bitovi stanja brojača menjaju se u toku programskog sken ciklusa na sledeći način:

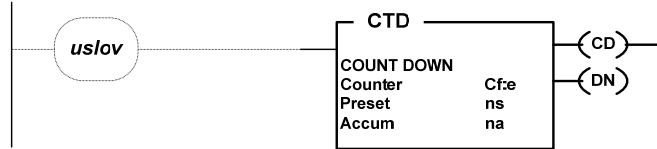
OV - Count up overflow bit se postavlja na 1 kada akumulirana vrednost (ACC) prelazi sa 32767 na –32768 (u binarnoj aritmetici drugog komplementa sa 16-bitnom reči važi: $32767+1=-32768$), i nastavlja brojanje unapredi.

DN - done bit se postavlja na 1 kada je $ACC \geq PRE$;

CU - Count up enable bit se postavlja na 1 kada je *uslov istinit*, a resetuje na 0 kada je *uslov neistinit* ili kada se aktivira odgovarajuća *RES* naredba.

- **Count down (CTD)**

CTD naredba, grafički simbol i položaj u rangu



Bitovi stanja brojača menjaju se u toku programskog sken ciklusa na sledeći način:

UN- Count down underflow bit se postavlja na jedan kada akumulirana vrednost (ACC) prelazi sa -32768 na 32767 (u binarnoj aritmetici drugog komplementa, sa 16-bitnom reči, važi: $-32768-1=32767$), i nastavlja da broji unazad od te vrednosti.

DN - done bit se postavlja na 1 kada je $ACC \leq PRE$;

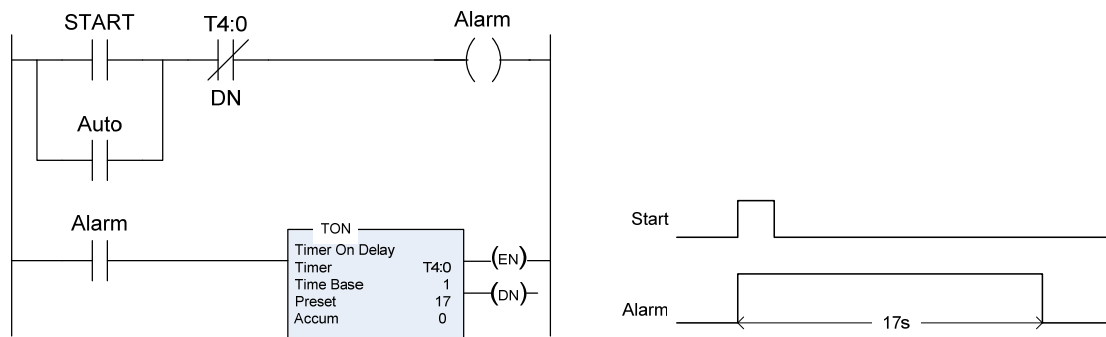
CD - Count down enable bit se postavlja na 1 kada je *uslov istinit*, a resetuje na 0 kada je *uslov neistinit* ili kada se aktivira odgovarajuća *RES* naredba.

5.3 Primeri korišćenja tajmera i brojača

Pr. 5-1 "One-shot" kolo

Zadatak: Ulaz u sistem je signal Start, a izlaz signal Alarm. Start=1 postavlja Alarm=1 u trajanju od 17s. (Start je impulsni signal čije trajanje može biti kraće ili duže od 17s.)

Rešenje: Sistem funkcioniše kao monostabilni multivibrator gde signal Start ima ulogu okidnog impulsa. Rešenje je dato na slici. U prvom rangu, Start=1 postavlja Alarm=1, što u drugom rangu omogućava tajmer koji se koristi za odmeravanje vremena od 17s. Po isteku 17s, tajmer setuje svoj DN bit, što raskida Start/Stop kolo iz prvog ranga i resetuje signal Alarm.

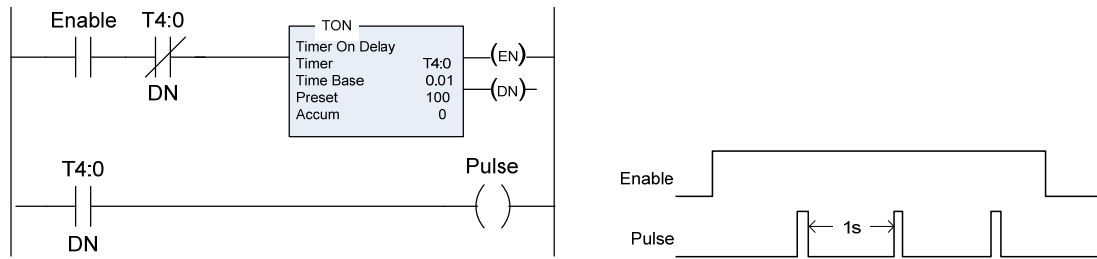


Sl. 5-7 Rešenje zadatka iz Pr. 5-1.

Pr. 5-2 Generator impulsa

Zadatak: Sve dok je ulazni signal Enable=1, na izlazu Pulse generišu se kratkotrajni impulsi periode 1s.

Rešenje: Rešenje sadrži dva ranga (Sl. 5-8). Tajmer iz prvog ranga je zadužen za odmeravanje vremena od 1s. Uvek kada završi odmeravanje jedne periode, tajmer resetuje samog sebe (bitom DN) i kreće iz početka (DN=1 resetuje tajmer, ali i postavlja DN=0, što ponovo omogućava rad tajmera). Dakle, DN=1 se javlja po isteku svake sekunde i traje tačno jedan sken ciklus. U drugom rangu, vrednost bita DN se prenosi na izlaz.

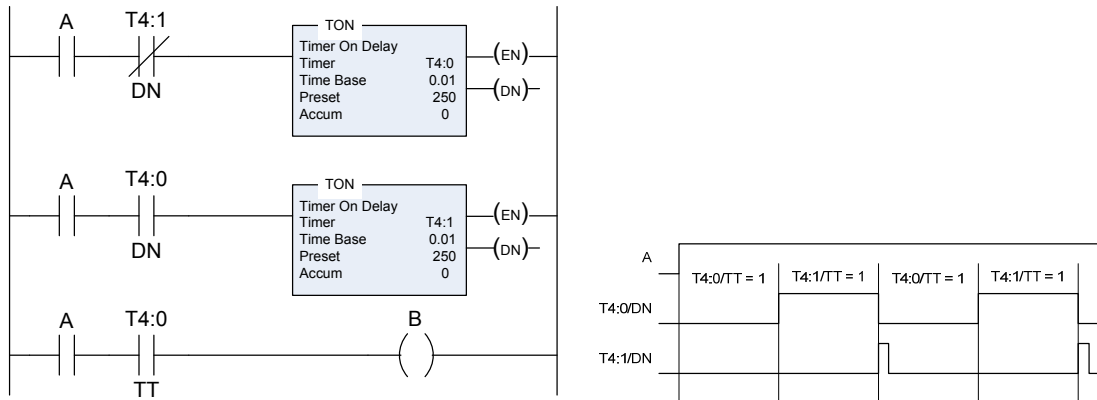


Sl. 5-8 Rešenje zadatka iz Pr. 5-2.

Pr. 5-3 Generator signala pravougaonog talasnog oblika

Zadatak: Sve dok je ulaz A=1, na izlazu B se generiše signal pravougaonog talasnog oblika, jednakog trajanja impulsa i pauze, periode 5s.

Rešenje: Rešenje koristi dva TON tajmera koji su međusobno povezani tako da naizmenično rade. Na početku, dok je A=0, oba tajmera su resetovana i onemogućena i oba Done bita imaju vrednost 0. Kada A postane 1, proradiće samo prvi tajmer (zbog T4:1/DN = 0), dok će drugi ostati zakočen (zbog T4:0/DN = 0). Kada prvi tajmer stigne do svoje preset vrednosti njegov Done bit se postavlja na DN=1, što omogućava drugi tajmer koji počinje da odbrojava osnovne vremenske jedinice. Uočimo da Done bit prvog tajmera ostaje 1, zbog uslova u prvom rangu koji je još uvek tačan. Kada drugi tajmer stigne do svoje zadate vrednosti, njegov Done bit se setuje, što ima za posledicu resetovanje prvog tajmera. Međutim, Done bit drugog tajmera ima vrednost 1 samo u toku jednog sken ciklusa, zato što već u sledećem sken ciklusu Done bit prvog tajmera dobija vrednost 0 i tako resetuje drugi tajmer. U ovom trenutku, uslovi su kao na samom početku: oba Done bita su 0; prvi tajmer počinje sa radom, dok drugi ostaje zakočen. Treći rang pobuđuje izlaz B: B ima vrednost 1 dok je A=1 i prvi tajmer je aktivan.



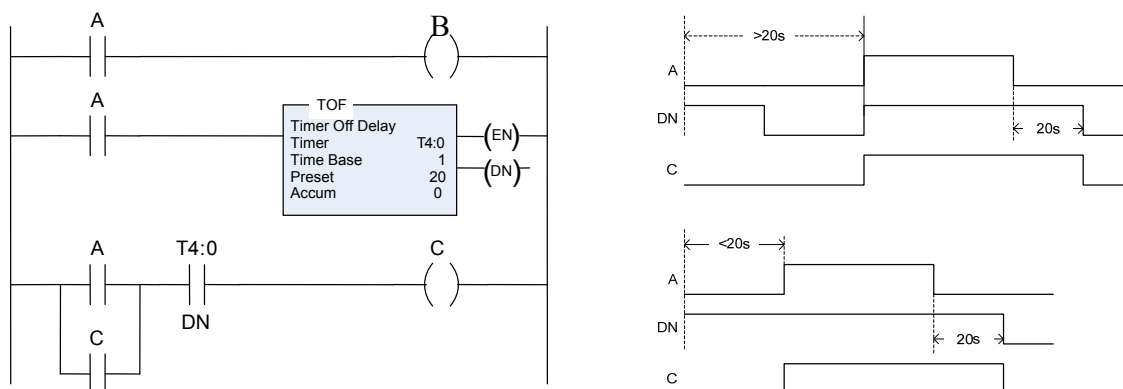
Sl. 5-9 Rešenje zadatka iz Pr. 5-3.

Pr. 5-4 Delay-off kolo

Zadatak: Postoji jedan ulaz, A, i dva izlaza, B i C. Izlaz B prati promene ulaza A, B=A. Izlaz C se uključuje kada i A=1, a isključuje 20s nakon A=0.

Rešenje: Ovaj zadatak je tipičan primer primene TOF tajmera (zakašnjene isključivanja). Za vreme dok je A=1, TOF tajmer iz drugog ranga je zakočen, a njegov Done bit ima vrednost DN=1 (Sl. 5-10). Tajmer počinje sa radom kada A=0. Po isteku zadatog vremena, tajmer se zaustavlja, a njegov DN bit dobija vrednost DN=0. Done bit zadržava vrednost 0 sve do sledećeg aktiviranja ulaza A. Done bit tajmera je iskorišćen u trećem rangu, zajedno sa ulazom A, za formiranje Start/Stop kola koje pobuđuje izlaz C. C se setuje ulazom A, tj. sa A=1, a resetuje Done bitom, tj. sa DN=0. Vremenski dijagrami sa Sl. 5-10 ilustruju dve moguće situacije koje nastaju kada je vreme od početka rada sistema pa do prvog aktiviranja ulaza A kraće, odnosno duže od 20s. Pošto na početku A ima vrednost 0, tajmer će biti omogućen, i ako se A ne postavi na 1 pre isteka 20s, tajmer će stići do kraja i resetovaće svoj Done bit (prvi slučaj). Međutim, ako je vreme do prvog aktiviranja ulaza A kraće od

20s, tajmer ne stiže do kraja, a njegov *Done* bit zadržava vrednost 1. Kao što vidimo, u oba slučaja, izlaz C, zahvaljujući Start/Stop kolu se ispravno generiše.

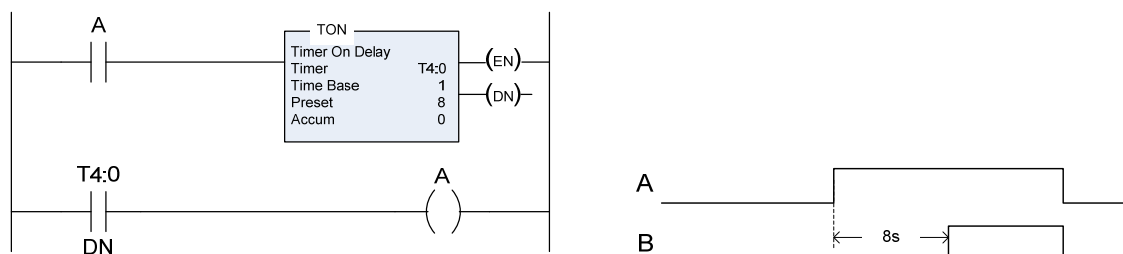


Sl. 5-10 Rešenje zadatka iz Pr. 5-4.

Pr. 5-5 Delay-on kolo

Zadatak: Jedan ulaz (A) i jedan izlaz (B). B se setuje 8s nakon što se A postavi na 1, a resetuje istovremeno kada A=0.

Rešenje: Ovaj zadatak ilustruje tipičnu primenu *delay-on* tajmera (odloženo uključenje). Rešenje sadrži samo dva ranga. Prvi rang sadrži tajmer koji se kontroliše ulazom A. Kada je A=0, tajmer je onemogućen i DN=0. Kada A postane 1, tajmer počinje da radi, a nakon odmerenog zadatog vremena njegov DN bit postaje 1. Drugi rang prenosi vrednost DN bitna na izlaz B.



Sl. 5-11 Rešenje zadatka iz Pr. 5-5.

Pr. 5-6 Dvostruko delay-off kolo

Zadatak: Jedan ulaz, A, i dva izlaza, B i C. Oba izlaza se uključuju kad i A. Izlaz B se isključuje sa zakašnjenjem od 10s, a C sa zakašnjenjem od 20s u odnosu na A=0.

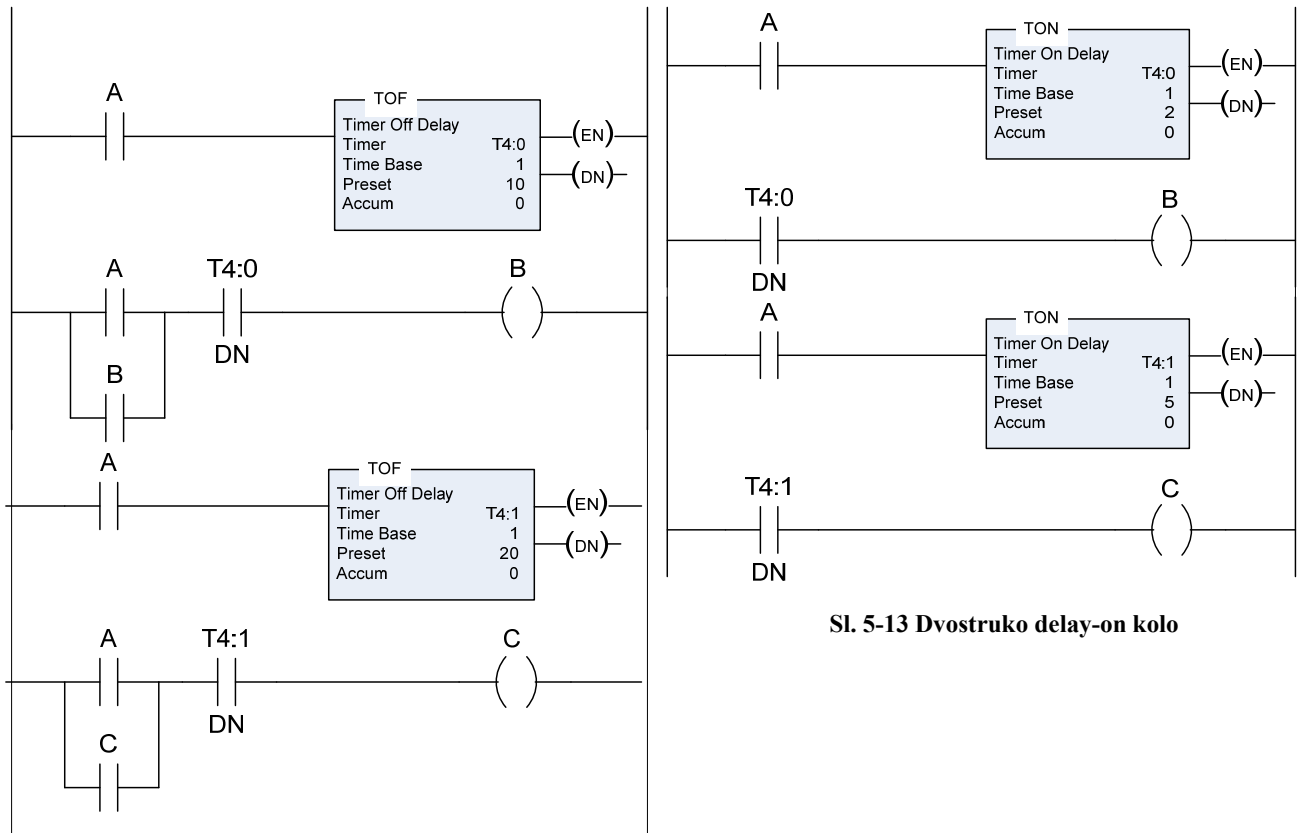
Rešenje: Zadatak možemo rešiti po ugledu na rešenje zadatka iz Pr. 5-4, s tom razlikom što sada treba koristiti dva para TOF tajmer - Start/Stop kolo. Oba tajmera se kontrolišu ulazom A. Tajmer za izlaz B je podešen na 10s, a tajmer za izlaz C na 20s. (Sl. 5-12)

Pr. 5-7 Dvostruko delay-on kolo

Zadatak: Jedan ulaz, A, i dva izlaza B i C. U odnosu na A, B se uključuje sa zakašnjenjem od 2s, a C sa zakašnjenjem od 5s. A=0, isključuje oba izlaza (B=C=0).

Rešenje: Zadatak možemo rešiti po ugledu na zadatak iz Pr. 5-5. Koristimo dva TON tajmera, jedan koji kontroliše izlaz B, a drugi izlaz C. Tajmer za B je podešen na 2s, a tajmer za C na 5s. (Sl. 5-13)

Drugo rešenje bi bilo da se tajmeri povežu "redno", tako da se tajmer za C omogućava DN bitom tajmera za B. Tajmer za B je podešen na 2s, a tajmer za C na 3s (5s-2s).

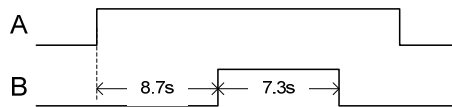


Sl. 5-13 Dvostruko delay-on kolo

Sl. 5-12 Dvostruko delay-off kolo

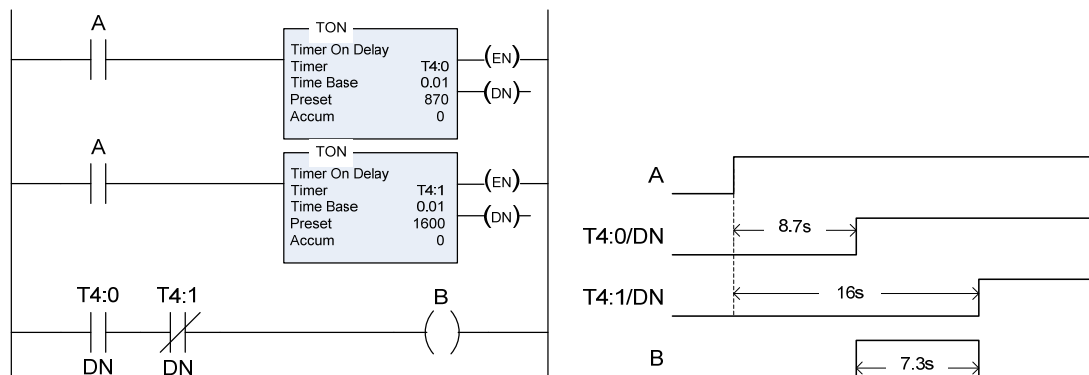
Pr. 5-8 "Pomereni" impuls

Zadatak: Jedan ulaz, A, jedan izlaz, B. $A=0 \Rightarrow B=0$. Za vreme dok je $A=1$, na izlazu B se generiše impuls koji počinje 8.7s nakon $A=1$ i traje 7.3s (Sl. 5-14).



Sl. 5-14 Vremenski dijagram za zadatak iz Pr. 5-8.

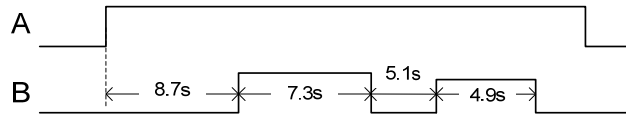
Rešenje: Potrebna su dva TON tajmera, prvi (T4:0) podešen na vreme 8.7s, a drugi (T4:1) na 16s (8.7 + 7.3). Oba tajmera se kontrolišu ulazom A (Sl. 5-15). Na vremenskom dijagramu sa Sl. 5-15 vidimo kako se menjaju *Done* bitovi tajmera: T4:0/DN postaje 1 8.7s nakon $A=1$, dok T4:1/DN postaje 1 16s nakon $A=1$. Lako je uočiti da izlaz B ima vrednost 1 kada je $T4:0/DN = 1$ i $T4:1/DN = 0$.



Sl. 5-15 Rešenje zadatak iz Pr. 5-8.

Pr. 5-9 Povorka impusla

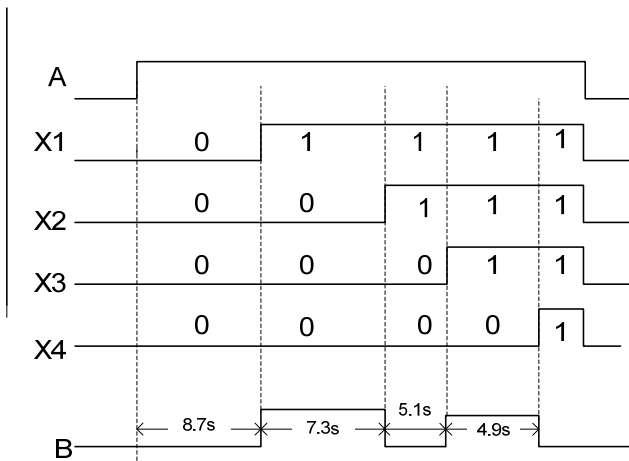
Zadatak: Jedan ulaz, A, i jedan izlaz, B. Za vreme dok je A=1, na izlazu B se generiše signal kao na Sl. 5-16.



Sl. 5-16 Vremenski dijagram za zadatak iz Pr. 5-9

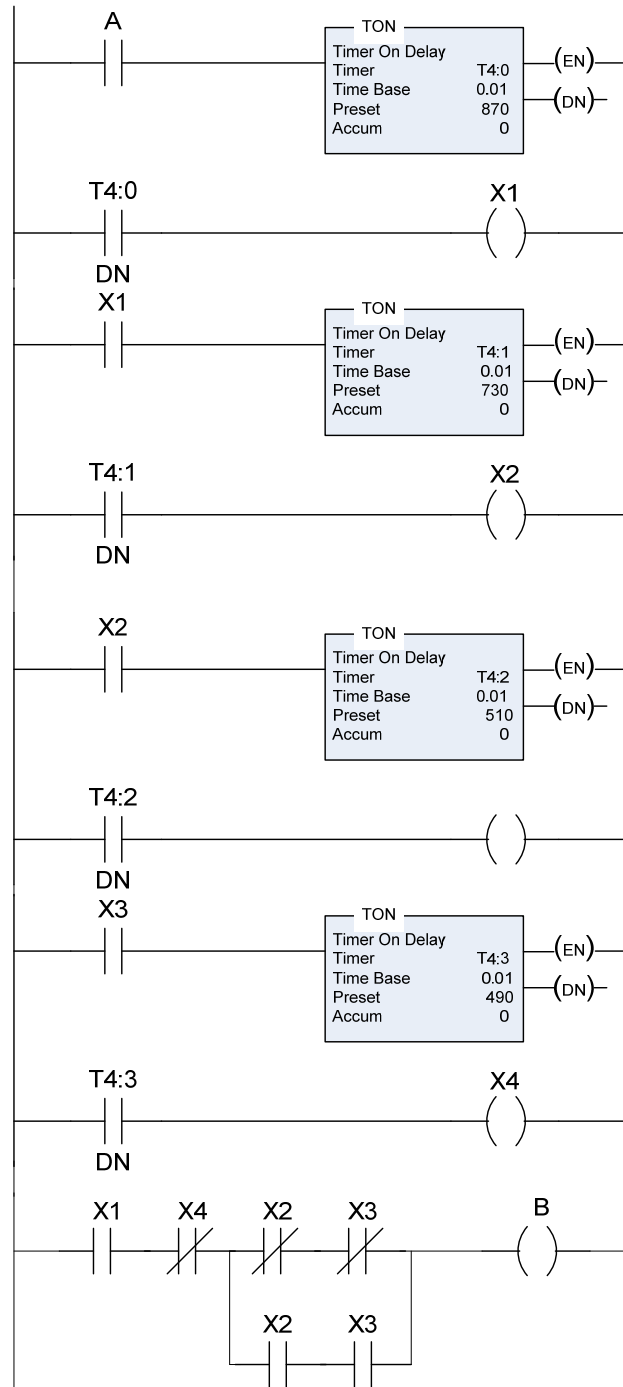
Rešenje:

Zadatak se rešava uvođenjem pomoćnih bitova X1, X2, X3 i X4, Sl. 5-17. Odgovarajući leder dijagram je prikazan na Sl. 5-18



$$B = X1\bar{X}2(\bar{X}1\bar{X}2 + X1X2)$$

Sl. 5-17 Talasni dijagram sa uvedenim pomoćnim bitovima

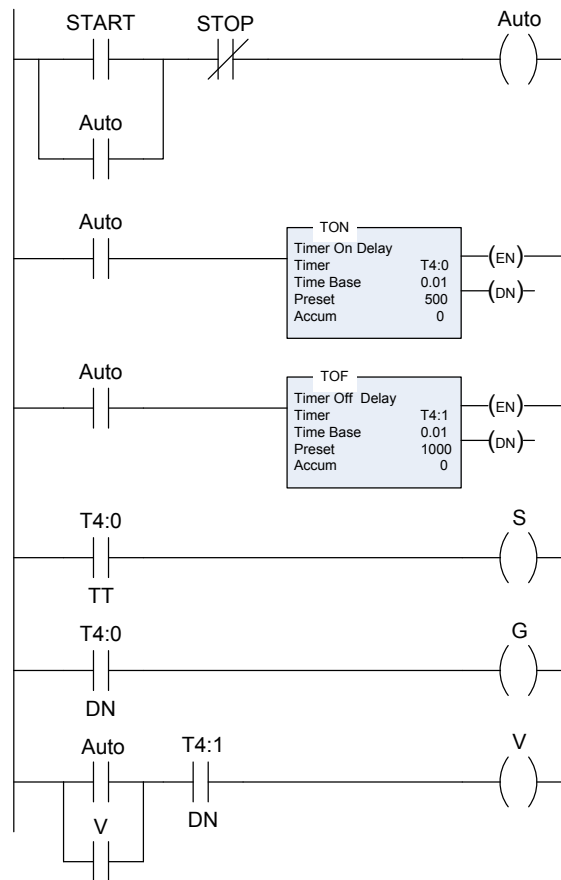


Sl. 5-18 Povorka impulsa

Pr. 5-10: Kontrola zagrevanja pećnice

Zadatak: Sistem čine: grejač, G, ventilator, V, sirena, S, i dva tastera, Start i Stop. Sistem se pušta u rad tasterom Start. Ventilator odmah počinje sa radom, a sirena emituje zvučni signal upozorenja u trajanju od 5s. Po isteku 5s, uključuje se i grejač. Pritiskom na taster Stop, grejač se isključuje, a ventilator nastavlja da radi još 10s.

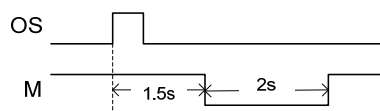
Rešenje: Jasno je da tastere Start i Stop treba povezati u Start/Stop kolo (Sl. 5-19). Neka je *Auto* bit za samodržanje (*Auto* se setuje sa Start, a resetuje sa Stop). U odnosu na bit *Auto*, uključivanje grajača kasni 5s, a isključivanje ventilatora za 10s. To znači da za upravljanje grejačem možemo iskoristiti TON kao u Pr. 5-5, a za upravljanje ventilatorom TOF tajmer, kao u Pr. 5-4. Leder dijagram je prikazan na Sl. 5-19. Uočimo da se sirena pobuđuje bitom TT TON tajmera, jer sirena treba da radi samo u prvih 5s nakon puštanja sistema u rad, tj. samo dok TON tajmer ne završi rad.



Sl. 5-19 Leder dijagram sistema za zagrevanje pećnice.

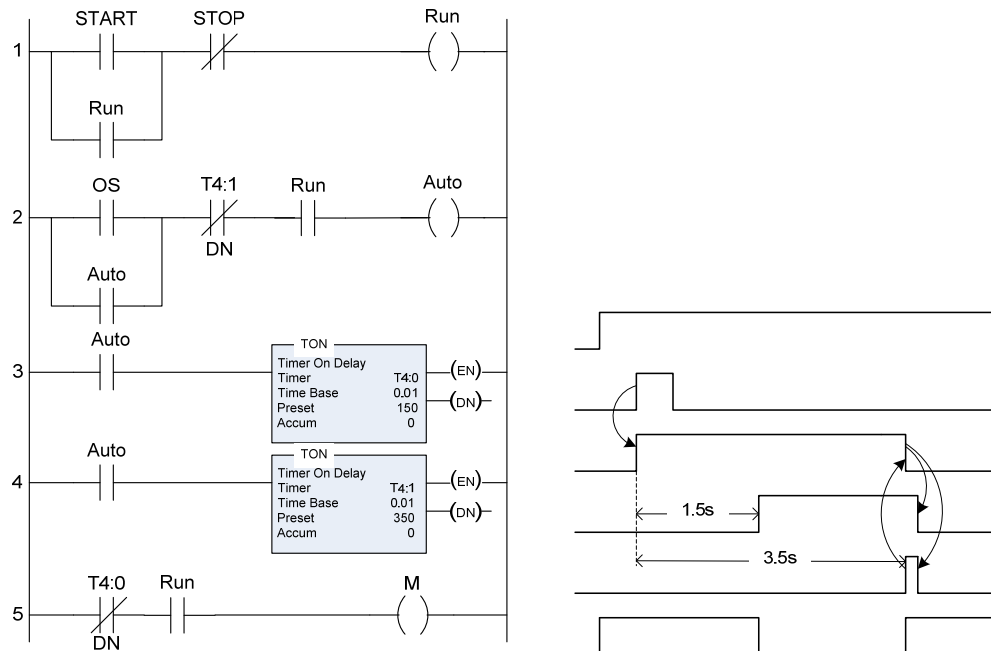
Pr. 5-11: Pozicioniranje predmeta na pokretnoj traci

Zadatak: Pokretna traka se pokreće motorom M. Motor se uključuje Start, a isključuje Stop tasterom. Za pozicioniranje predmeta koji se prenosi trakom koristi se optički senzor, OS. Traka se zaustavlja 1.5s nakon što OS detektuje prolazak predmeta, ostaje zaustavljena 2s i ponovo se pokreće (slika).



Sl. 5-20 Vremenski dijagram za zadatak Pr. 5-11

Rešenje: Koriste se dva TON tajmera, koja su slično kao u primeru "Pomereni impuls", podešeni na vremena 1.5s i 3.5s (1.5+2). Uvedena su dva pomoćna bita, Run i Auto. Bit Run, koji se postavlja Start/Stop kolom koje formiraju ulazi Start i Stop (rang 1), predstavlja dozvolu rada sistema. Bit Auto se postavlja Start/Stop kolom u rang 2. Auto se setuje signalom senzora OS, sa OS=1, a resetuje kada tajmer T4:1 odmeri zadato vreme (tj. nakon 3.5s). Auto se koristi kao dozvola rada tajmera i dodatno je uslovljen Ran bitom, tako da dok je sistem stopiran (Run=0) i Auto ostaje 0, a tajmeri su onemogućeni. Kada se na ulazu OS javi 1, tajmeri počinju da rade (zbog Auto=1). T4:0/DN postaje 1 nakon 1.5s, a T4:1/DN posle 3.5s. Pojava T4:1/DN=1 resetuje bit Auto (u rang 2), što ima za posledicu da se već u sledećem sken ciklusu resetuju i onemoguće oba tajmera. Na taj način su pripremljeni uslovi za novi ciklus rada. Na osnov u vremenskih dijagrama lako je uočiti da izlazni signal M treba da ima vrednost 0 kada je T4:0/DN = 1 uz dodatni uslov Run=1 - sistem je startovan (rang 5).

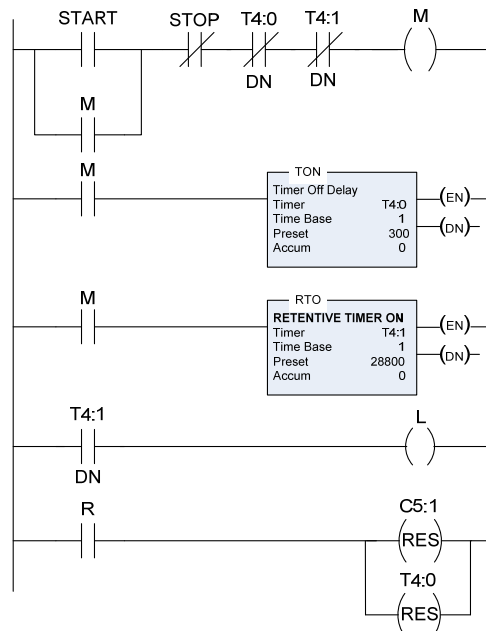


Sl. 5-21 Vremenski dijagram za zadatak iz Pr. 5-11

Pr. 5-12 RTO tajmer

Zadatak: Motor M se uključuje pomoću Start, a isključuje pomoću Stop tastera. Motor ne sme biti uključen duže od 5 min i automatski se isključuje ako se ovo vreme prekorači. Takođe, meri se i ukupno vreme rada motora. Nakon što se dostigne vreme od 8h rada, više nije moguće uključiti motor, a pali se signalna lampa L koja upozorava da treba obaviti servisiranje. Po obavljenom servisiranju, sistem se resetuje tasterom R.

Rešenje: U ovom primeru se koristi TON tajmer za merenje maksimalno dozvoljenog vremena neprekidnog rada motora i RTO tajmer za merenje ukupnog vremena rada motora. Kao što vidimo, oba tajmera se kontrolišu bitom M, koji pobuđuje motor; dok motor radi, rade i oba tajmera. Istek zadatog vremena bilo kog od dva tajmera trenutno isključuje motor (prvi rang). U slučaju da je prekoračeno vreme RTO tajmera, jedini način da se sistem ponovo pokrene je resetovanje tasterom R.

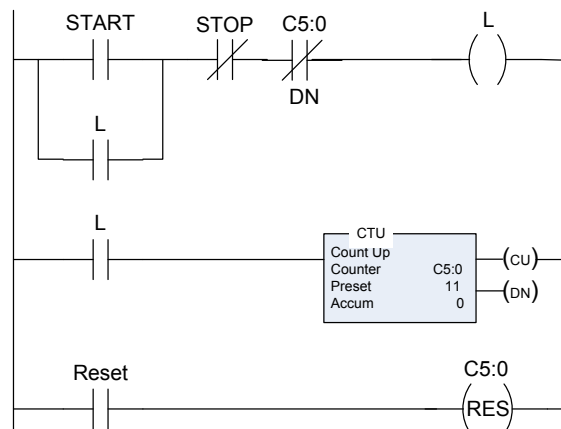


Sl. 5-22 Rešenje zadatka iz Pr. 5-12

Pr. 5-13 Ograničeni broj uključivanja

Zadatak: Sijalica S se pali pomoću Start, a gasi pomoću Stop tastera. Potrebno je ograničiti broj paljenja sijalice na 10. Postoji i taster Reset, koji resetuje broj uključivanja.

Rešenje: Preset vrednost brojača je 11, tako da 11. pokušaj paljenje sijalice dovodi do C5:5/DN = 1, što u prvom rangu onemogućava paljenje sijalice (Sl. 5-23). Reset=1, resetuje brojač, tj. postavlja C5:5.ACC=0 i C5:5/DN=0. Na taj način, omogućeno je paljenje sijalice, a brojanje paljenja kreće iz početka.

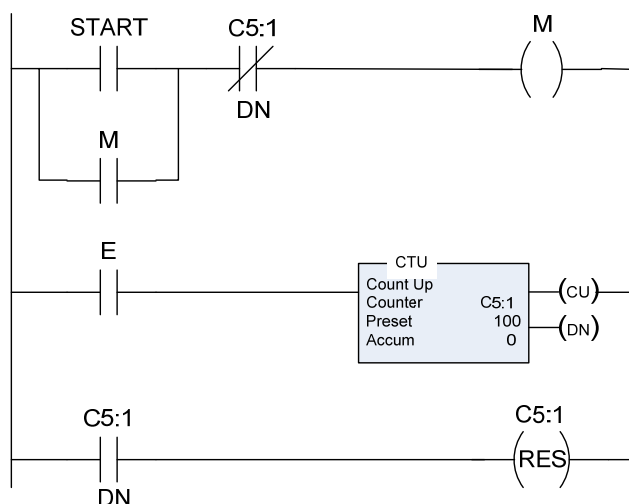


Sl. 5-23 Rešenje zadatka iz Pr. 5-13.

Pr. 5-14 Zadati broj okretaja motora

Zadatak: Pritiskom na taster Start pokreće se motor M. Nakon 100 rotacija motor se zaustavlja. Na osovini motora je ugrađen tzv. enkoder, E, koji registruje obrtaje i za svaki obrtaj daje jedan impuls.

Rešenje: Motorom se upravlja pomoću Start/Stop kola (prvi rang), kod koga taster Start setuje, a Done bit brojača resetuje izlazni bit M. Brojač broji impulse sa enkodera, a njegova preset vrednost je 100. Nakon 100 impulsa C5:1/DN postaje 1, što kao posledicu ima isključene motora (prvi rang) i resetovanje brojača (treći rang), čime je ujedno sistem pripremljen za novo startovanje.

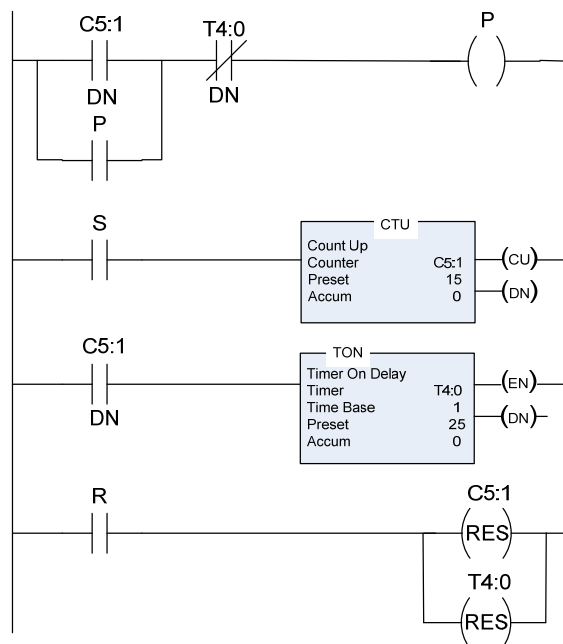


Sl. 5-24 Rešenje zadatka iz Pr. 5-14.

Pr. 5-15 Tajmer i brojač (1)

Zadatak: Nakon izbrojanih 15 impulsa sa senzora S, izlaz P se pobuđuje (P=1) u trajanju od 25s. Taster R služi za resetovanje sistema.

Rešenje: Rešenje (Sl. 5-25) koristi brojač za brojanje impulsa i tajmer za merenje vremena. Brojač broji unapred (CTU) i podešen je na zadatu vrednost 15. Tajmer je TON tipa i podešen je na zadato vreme od 25s. Izlaz P se setuje kada brojač dostigne zadatu vrednost (Done bitom brojača). Istovremeno, izlaz P se koristi kao dozvola rada tajmera. Izlaz P se resetuje kada tajmer dostigne zadatu vrednost (Done bitom tajmera). Taster R posredstvom RES naredbi, resetuje brojač i tajmer. Nakon resetovanja, sistem je spreman za novi ciklus rada.

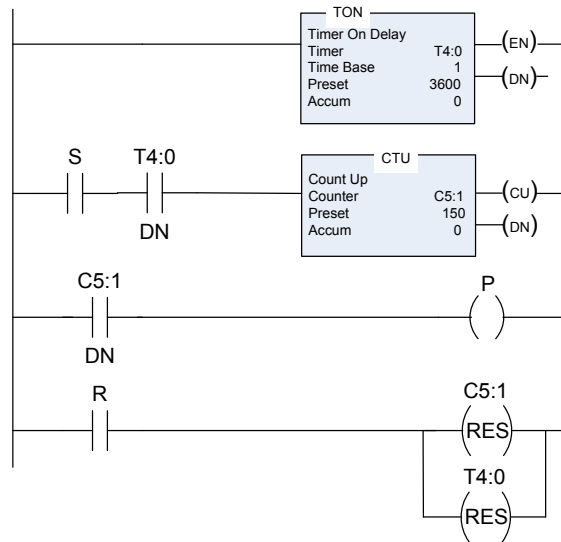


Sl. 5-25 Rešenje zadatka iz Pr. 5-15.

Pr. 5-16 Tajmer i brojač (2)

Zadatak: Brojanje impulsa koji potiču sa senzora S počinje 1h nakon startovanja sistema. Nakon izbrojanih 150 impulsa, aktivira se izlaz P. Taster R služi za resetovanje sistema.

Rešenje: U datom rešenju (slika), u rangu koji sadrži tajmer na postoji uslov. To znači da tajmer radi neprekidno (uslov je uvek tačan) i može se resetovati samo naredbom RES.



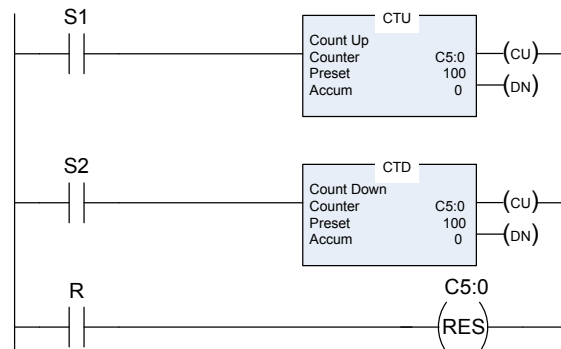
Sl. 5-26 Rešenje zadatka iz Pr. 5-16.

Pr. 5-17 Brojanje proizvoda na pokretnoj traci

Zadatak: Potrebno je doći do informacije o broju proizvoda koji se trenutno nalaze na pokretnoj traci. Postavljena su dva senzora za detekciju prolaska proizvoda: prvi, S1, na početku i drugi, S2, na kraju trake. S1 registruje proizvode koji se stavljaju na traku, a S2 proizvode koji napuštaju traku. Informacija koju tražimo je $P = N(S1) - N(S2)$, gde je $N(Si)$ broj proizvoda koje je registrovao senzor S_i . Kako će ta informacija dalje biti korišćena nije problem koji treba rešavati u ovom zadatku.

Rešenje: Koristimo dva brojača: jedan tipa CTU (broji napred) koji broji impulse sa senzora S1 i drugi tipa CTD (broji unazad) koji broji impulse sa senzora S2 (slika). Treba zapaziti da oba brojača dele isti element iz datoteke brojača, C5:0. To znači da će oba brojača ažurirati isti registar (ACC) kada odbrojavaju impulse. Svaki impuls sa senzora S1 uvećaće vrednost u C5:0.ACC za jedan, dok će svaki impuls sa senzora S2 umanjiti ovu vrednost za 1. Dakle, tražena informacija je prisutna u registru C5:0.ACC.

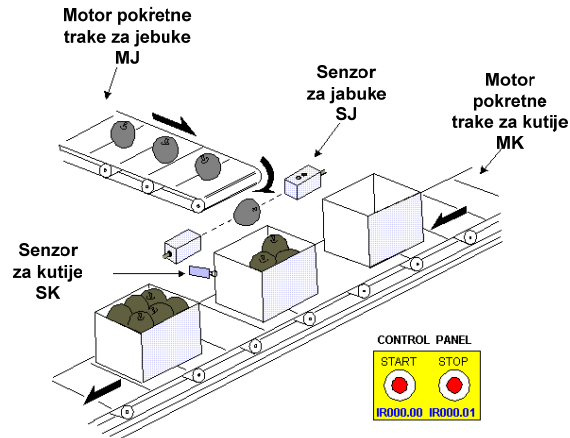
Napomenimo da u konfiguraciji "preklopljenih" brojača kao sa slike (kada dva brojača koriste istu adresu), preset vrednosti moraju biti iste, a DN bit se postavlja na 1 kada $ACC \geq PRE$ (tj. kao da se radi o CTU brojaču). U konkretnom primeru, DN bit se ne koristi, pa iz tog razloga preset (PRE) vrednost nije od značaja.



Sl. 5-27 Rešenje zadatka iz Pr. 5-17.

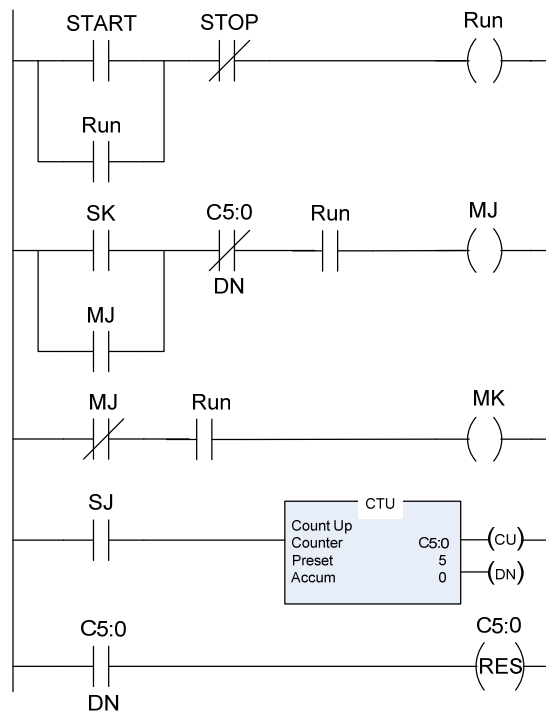
Pr. 5-18 Sistem za pakovanje jabuka u kutije

Zadatak: Na Sl. 5-28 je prikazan sistem za pakovanje jabuka u kutije. Sistem se pušta u rad tasterom Start, a zaustavlja tasterom Stop. Pokretna traka za kutije se pokreće motorom MK, a traka za jabuke motorom MJ. Prisustvo kutije na mestu za punjenje detektuje se senzorom SK, dok se za detekciju jabuka koristi senzor SJ. Kada se na mestu za punjenje detektuje kutija, traka za kutije se zaustavlja, a pokreće traka za jabuke. Nakon izbrojanih 5 jabuka, traka za jabuke se zaustavlja, a traka za kutije ponovo pokreće.



Sl. 5-28 Sistem za pakovanje jabuka u kutije

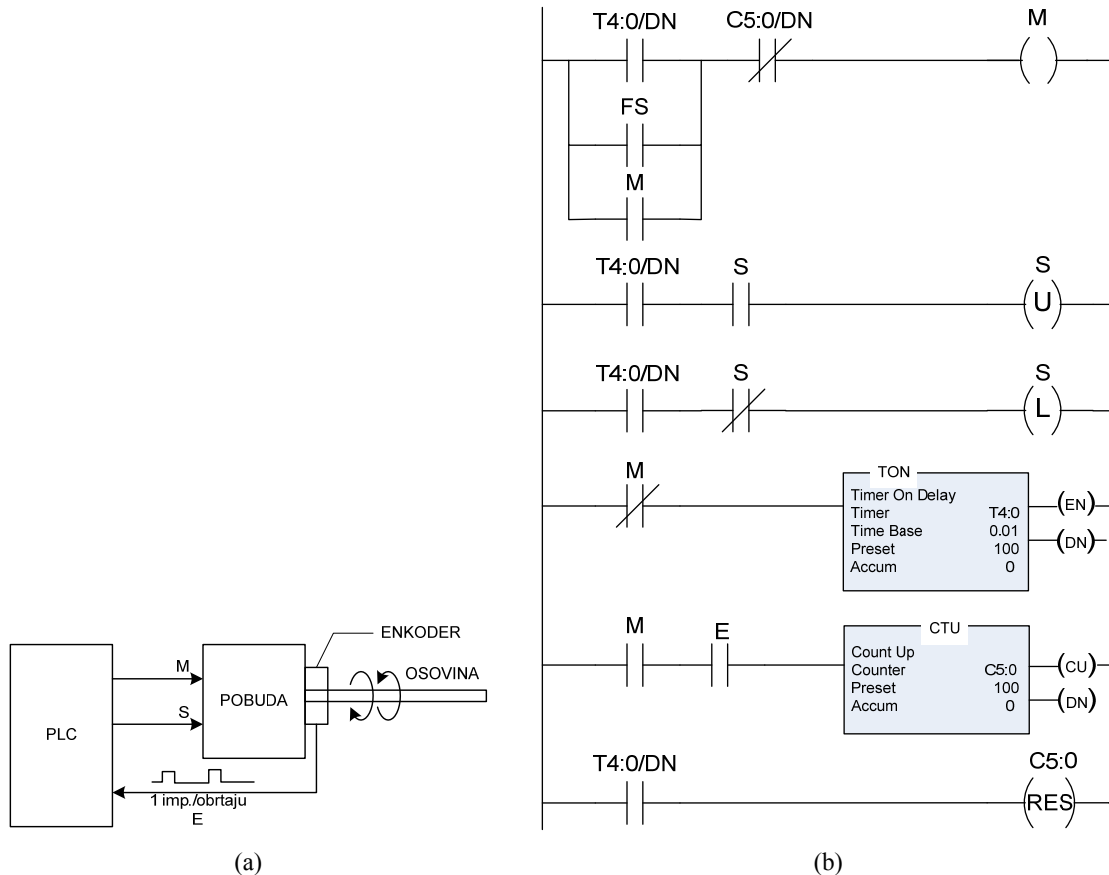
Rešenje: Jasno je da za brojanje jabuka treba koristiti brojač koji se taktuje signalom SJ. Takođe je jasno da dok radi MK ne radi MJ i obrnuto. Motorom MJ se može upravljati start/stop kolom kod koga će SK imati ulogu start, a Done bit brojača ulogu stop bita. Izlaz MK će biti komplement bita MJ. Naravno, Start i Stop tastere treba povezati u start/stop kolo, a pomoćni bit koji se koristi za samodržanje upotrebiti kako dodatni uslov za pobudu motora. Uz to, brojač jabuka treba resetovati uvek nakon izbrojanih 5 jabuka (Done bitom brojača) (Sl. 5-29).



Sl. 5-29 Leder dijagram za Pr. 5-18.

Pr. 5-19 Upravljanje rotacijom osovine

Zadatak: Na Sl. 5-30(a) prikazan je sistem za upravljanje rotacijom osovine koja je deo nekog složenijeg elektro-mehaničkog sistema. Blok POBUDA, koji sadrži motor i prateću elektroniku, okreće osovinu u zadanom smeru. Za $M=0$, osovina miruje; za $M=1$ i $S=0$, osovina se okreće u jednom, a za $M=1$ i $S=1$ u drugom smeru. Blok ENKODER registruje obrtaje osovine i generiše jedan impuls, E, za svaki obrtaj. Realizovati leder program za PLC kontroler koji će omogućiti da osovina naizmenično rotira po 100 obrtaja u oba smera sa pauzom od 1s pri svakoj promeni smera.



Sl. 5-30 Sistem za upravljanje rotacijom osovine: (a) blok dijagram sistema; (b) leder dijagram.

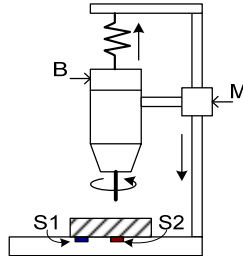
Rešenje: Sistem ne poseduje tastere za startovanje i zaustavljanje. Zato ćemo pretpostaviti da odmah nakon uključanja napajanja sistem počinje sa svojim normalnim radom. Za realizaciju leder dijagrama iskoristićemo brojač, za brojanje obrtaja, i tajmer za omeravanje pauze nakon završene rotacije u jednom smeru. Izlaz M, koji uključuje/isključuje motor, može se pobuđivati start/stop kolom, kod koga će Done bit tajmera imati ulogu start, a Done bit brojača ulogu stop uslova. (Kada brojač odbroji 100 obrtaja, motor se zaustavlja; kada tajmer odmeri 1s, motor se pokreće.) Da bi smo obezbedili da odmah po uključenu napajanja motor počne da se okreće, dodatni uslov za setovanje izlaza M biće "first scan" bit, FS. Izlazni bit S se menja pri svakom novom uključenu motora, odnosno uvek kada Done bit tajmera postane 1. Uslov za rad tajmera je "motor isključen", a uslov za dozvolu brojanja "motor uključen". Uslov za resetovanje brojača je, kao i za promenu smera, Done bit tajmera je 1.

Leder dijagram je prikazan na Sl. 5-30(b). Primetimo da $T4:0/DN=1$ traje tačno jedan sken ciklus. Da to nije tako, setovanje i resetovanje izlaznog bita S ne bi bilo regularno.

Studentima se predlaže da ovaj zadatak reše primenom metoda konačnih automata.

Pr. 5-20 Upravljanje automaskom bušilicom

Zadatak: Kreirati leder dijagram za PLC koji upravlja automaskom bušilicom (Sl. 5-31). Operacije i uslovi su sledeći: (1) Kada se na radnu površinu postavi komad za bušenje (detektuje se senzorom S1), uključuju se motor bušilice, B, i motor koji pokreće bušilicu naniže, M. (2) Kada je komad probušen (detektuje se senzorom S2), motor M se isključuje, a bušilica se pod dejstvom opruge vraća u polazni položaj. 3) motor bušilice B se isključuje 5s nakon završenog bušenja, pod uslovom da u međuvremenu na radnu površinu nije postavljen novi komad za bušenje. Zadatak rešiti primenom metoda konačnih automata.



Sl. 5-31 Automatska bušilica

Rešenje: Analizom problema možemo doći do zaključka da sistem poseduje sledeća četiri stanja:

F0 - inicijalno (neaktivno) stanje sistema, u kome se čeka na postavljanje komada na mesto za bušenje (na aktiviranje senzora S1).

F1 - bušenje, u kome su uključeni motor bušilice i motor za spuštanje bušilice ($M=B=1$). Ovo stanje se završava u trenutku kada burgija probije predmet koji se buši (aktiviran senzor S2).

F2 - čekanje da se probušeni predmet ukloni. U ovom stanju, motor bušilice je i dalje uključen ($B=1$). Motor za spuštanje je isključen ($M=0$) tako da se bušilica pod dejstvom opruge vraća u svoj gornji položaj. Takođe, u ovom stanju je omogućen tajmer koji odmerava vreme čekanja, podešeno na 5s. Usvojicemo da se rad tajmera kontroliše bitom TE. U ovom stanju važi $TE=1$. Stanje se završava kada se sa mesta za bušenje ukloni predmet (deaktivira senzor S1).

F3 - čekanje na novi predmet ili istek vremena čekanja. U ovom stanju, motor bušilice još uvek radi ($B=1$), a tajmer aktivno meri vreme ($TE=1$). Stanje se završava u momentu kada se predmet ukloni ($S1=1$) ili istekne zadato vreme (*Done* bit tajmera je aktiviran, $TDN=1$). U prvom slučaju, ($S1=1$), sistem se vraća u stanje F1, a u drugom ($TDN=1$) u stanje F0.

Odgovarajući ASM dijagram je prikazan na Sl. 5-32(a).

Jednačine stanja su:

$$F0 = (F0 + F3 \cdot \overline{S1} \cdot TDN) \cdot \overline{F0} \cdot \overline{S1} = (F0 + F3 \cdot \overline{S1} \cdot TDN) \cdot (\overline{F0} + \overline{S1}) + FS$$

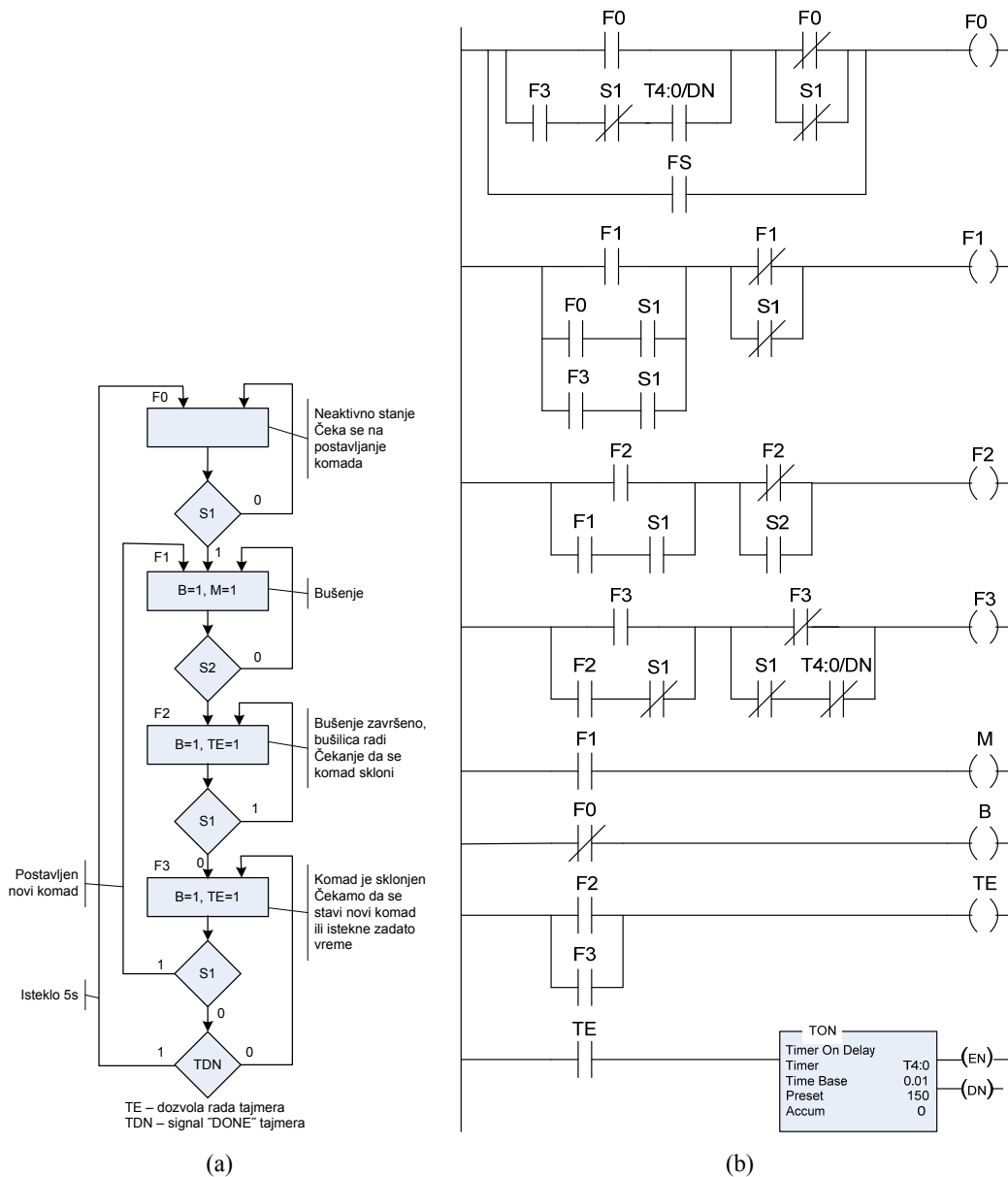
$$F1 = (F1 + F0 \cdot S1 + F3 \cdot S1) \cdot \overline{F1} \cdot \overline{S2} = (F1 + F0 \cdot S1 + F3 \cdot S1) \cdot (\overline{F1} + \overline{S2})$$

$$F2 = (F2 + F1 \cdot S2) \cdot \overline{F1} \cdot \overline{S2} = (F2 + F1 \cdot S2) \cdot (\overline{F2} + S2)$$

$$F3 = (F3 + F2 \cdot \overline{S1}) \cdot \overline{F3} \cdot \overline{S1} \cdot \overline{F3} \cdot \overline{S1} \cdot TDN = (F3 + F2 \cdot \overline{S1}) \cdot (\overline{F3} + \overline{S1} \cdot TDN)$$

Sistem poseduje dva fizička izlaza, M i B. Međutim, i bit za kontrolu tajmera, TE, se može smatrati izlazom. Primetimo da se tajmer, kao takav, ne pojavljuje u ASM dijagramu, već se tretira kao eksterna jedinica kojom se upravlja bitom TE. Shodno ASM dijagramu, bit M je aktivan u stanju F1, tj. $M=F1$; bit B je aktivan u stanjima F1, F2 i F3, tj. $B=F1+F2+F3=\overline{F0}$, dok je bit TE aktivan u stanjima F2 i F3, tj. $TE=F2+F3$.

Leder dijagram je prikazan na Sl. 5-32(b).

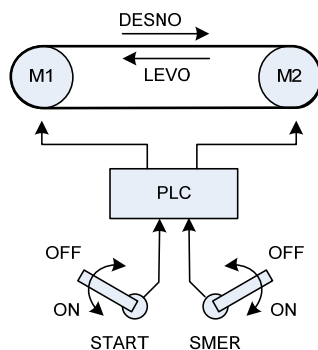


Sl. 5-32 Rešenje zadatka iz Pr. 5-20 : (a) ASM dijagram; (b) leder dijagram.

Pr. 5-21 Dvosmerna pokretna traka

Zadatak: Dvosmerana pokretna traka se pokreće motorima M1 i M2 (Sl. 5-33). Motorima se upravlja pomoću dva dvopoložajna (ON/OFF) prekidača, START i SMER. Za START=OFF, oba motora su isključena (traka se ne kreće). Za START=ON i SMER=OFF, motor M1 je uključen, a M2 isključen (traka se kreće u smeru LEVO); za START=ON i SMER=ON, M1 je isključen, a M2 uključen (traka se kreće u smeru DESNO). Realizovati leder program koji će obezbediti pauzu od 1s pri promeni smeru kretanja.

Na primer, ako se u toku kretanja trake promeni položaj prekidača SMER, traka se zaustavlja i nastavlja kretanje u suprotnom smeru posle 1s. Ako se pre isteka 1s prekidač SMER vrati u prvobitni položaj, traka trenutno nastavlja kretanje u istom smeru. Pauzu od 1s treba obezbediti i u sledećoj situaciji: kretanje trake je zaustavljeno prekidačem START (START=OFF), zatim je promenjen položaj prekidača SMER i prekidač START vraćen u položaj ON.



Sl. 5-33 Dvoscimerna pokretna traka.

Rešenje: Ulazi u sistem su stanja prekidača START i SMER, a izlazi signali za upravljenje motorima M1 i M2. Međutim, način na koji se (START, SMER) preslikava na (M1, M2), zavisi od tekućeg stanja sistema. Na primer, sistem na jedan način reaguje na prekidače ako se traka kreće, na drugi ako je od trenutka zaustavljanja trake prošlo manje od 1s, a na treći ako traka stoji duže od jedne sekunde.

Stanja sistema su:

S1 - neaktivno stanje; sistem je isključen i čeka se na START=1.

S2 - kretanje u smeru levo; motor M1 radi, a M2 je isključen. Ovo stanje se prekida kada se promeni položaj bilo kog prekidača (kada START=0, ili SMER=1).

S3 - pauza nakon prekida kretanja u smeru levo. Traka stoji i omogućen je tajmer za merenje 1s. Uslovi za napuštanje ovog stanja su: 1) prekidačima je ponovo postavljen smer kretanja u levo (START=1 i SMER=0); 2) istekla pauza.

S4 - kretanje u smeru desno. Motor M1 je isključen, a motor M2 radi. Ovo stanje se prekida kada se promeni položaj bilo kog prekidača (kada START=0, ili SMER=1).

S5 - pauza nakon prekida kretanja u smeru desno. Traka stoji i omogućen je tajmer za merenje 1s. Uslovi za napuštanje ovog stanja su: 1) prekidačima je ponovo postavljen smer kretanja u desno (START=1 i SMER=1); 2) istekla pauza.

Odgovarajući ASM dijagram je prikazan na Sl. 5-34. Nakon puštanja u rad (START=1), sistem prelazi u stanje S1 ili S2, zavisno od položaja prekidača SMER. U toku kretanja, svaka promena položaja bilo kog prekidača, START ili STOP, zaustavlja kretanje i omogućava rad tajmera koji se koristi za odmeravanje pauze od 1s (stanje S3 ili S5). Ako se za vreme pauze, prekidači postave u položaje koji odgovaraju smeru kretanju koji je važio neposredno pre zaustavljanja, sistem momentalno nastavlja kretanje u tom smeru. U suprotnom, ako se to ne desi, već istekne zadato vreme čekanja, sistem se vraća u početno stanje, S1. Može se desiti da se sistem vrati u početno stanje iako je START=1. U tom slučaju, sistem ostaje u stanju S1 samo jedan sken ciklus, da bi odmah zatim prešao u stanje S2 ili S3. Na Sl. 5-35 prikazan je odgovarajući leder dijagram.

Jednačine stanja su:

$$T = START * \overline{SMER}$$

$$K = START \cdot SMER$$

$$S1 = (S1 + S3 \cdot \overline{T} \cdot DN + S5 \cdot \overline{K} \cdot DN) \cdot \overline{S1} \cdot \overline{START} + FS = (S1 + DN \cdot (S3 \cdot \overline{T} + S5 \cdot \overline{K})) \cdot (\overline{S1} + \overline{START}) + FS$$

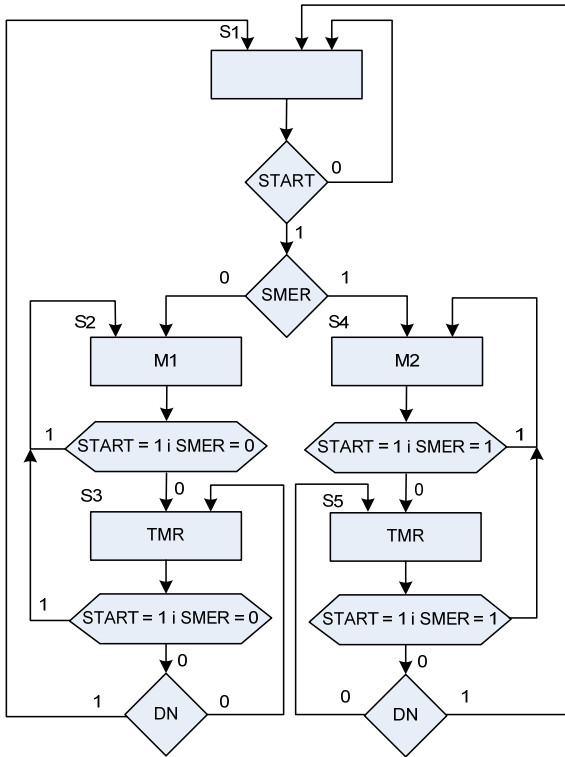
$$S2 = (S2 + T \cdot S1 + T \cdot S3) \cdot \overline{S2} \cdot \overline{T} = (S2 + T \cdot (S1 + S3)) \cdot (\overline{S2} + T)$$

$$S4 = (S4 + S1 \cdot K + S5 \cdot K) \cdot (S4 \cdot \overline{K}) = (S4 + K \cdot (S1 + S5)) \cdot (\overline{S4} + K)$$

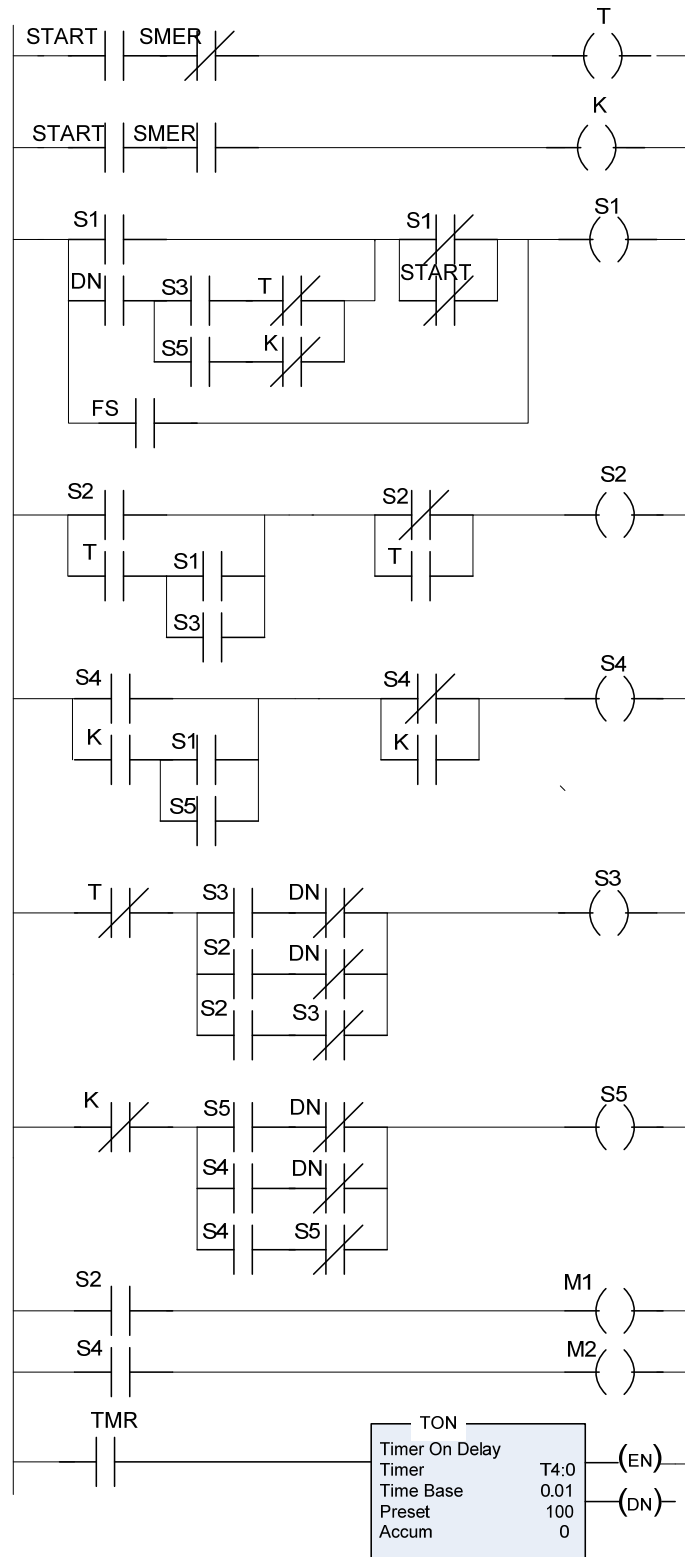
$$S3 = (S3 + S2 \cdot T) \cdot \overline{S3} \cdot T \cdot \overline{S3} \cdot \overline{T} \cdot DN = \overline{T} \cdot (DN \cdot S3 + \overline{DN} \cdot S2 + S2 \cdot \overline{S3})$$

$$S5 = (S5 + S4 \cdot \overline{K}) \cdot \overline{S5} \cdot \overline{K} \cdot \overline{S5} \cdot \overline{K} \cdot DN = \overline{K} \cdot (DN \cdot S5 + \overline{DN} \cdot S4 + S4 \cdot \overline{S5})$$

$M1 = S2$
 $M2 = S4$
 $TMR = S3 + S5$



Sl. 5-34 ASM dijagram za Pr. 5-21

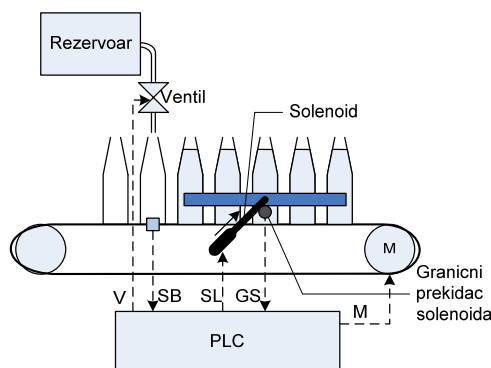


Sl. 5-35 Leder dijagram za Pr. 5-21

Pr. 5-22 Sistem za punjenje boca

Zadatak: Na Sl. 5-36 prikazan je sistem za punjenje boca. Sistem čine: (a) pokretna traka koja se pokreće motorom M; (b) rezervoar iz kojeg se tečnost ispušta ventilom sa ON/OFF upravljanjem, V;

(c) senzora SB koji detektuje prisustvo boce na mestu za punjenje; (d) elektro-mehaničkog mehanizma za sklanjanje napunjenih boca sa trake, koji se pokreće jednosmernim solenoidom, SL, i (e) PLC kontrolera. Kada boca stigne na mesto za punjenje, traka se zaustavlja i ventil rezervoara otvara. Punjenje traje 2s, a zatim se ventil zatvara. Na svakih 5 napunjenih boca, pokreće se mehanizam koji uklanja napunjene boce sa trake. Kreirati odgovarajući lader dijagram.



Sl. 5-36 Sistem za punjenje boca.

Rešenje: Detaljan opis rada sistema, u vidu ASM dijagrama, prikazan je na Sl. 5-37(a). PLC koristi tajmer za odmeravanje vremena punjenja i brojač za brojanje napunjenih boca. TMR je bit koji omogućava rad tajmera, a TDN *Done* bit tajmera. N je tekuća vrednost brojača boca. Sistem ostaje u stanju ST0 u kojem se traka kreće (M=ON) sve do aktiviranja senzora boce (SB). Kada senzor detektuje prisustvo boce (SB=ON), brojač boca se uvećava za jedan i sistem prelazi u stanje ST1. U ovom stanju dozvoljen je rad tajmera (TMR=1), a ventil rezervora je otvoren (V=1). Po isteku zadatog vremena punjenja (TDN=1), proverava se broj napunjenih boca. Ako je napunjeno manje od 6 boca, sistem se vraća u početno stanje, ST1, gde se traka ponovo pokreće. Inače, ako je izbrojano 6 boca, tada se brojač boca resetuje (N=0) i sistem prelazi u stanje ST2. U ovom stanju aktivna je pobuda solenoida (SL=1). Kada je solenoid izvučen do kraja (aktivan je granični prekidač - GS=1), sistem se vraća u početno stanje.

ASM dijagram sa Sl. 5-37(a) je specifičan po tome što sadrži "zaobljene pravougaonike", tj. blokove uslovnog izlaza. Kao što znamo, na ovaj način se predstavljaju tzv. Milijevi izlazi konačnog automata, tj. izlazni signali koji zavise ne samo od tekućeg stanja automata, već i od nekog dodatnog uslova. U konkretnom slučaju, inkrementiranje brojača se inicira kada je sistem u stanju ST0, ali samo pod uslovom ako je SB=ON. Slično, brojač se resetuje kada je sistem u stanju ST1 i pri tome je tajmer odmerio zadato vreme (TDN=1) i brojač odbrojao do zadatog broja.

Izvedimo, napre, jednačine izlaza. U sistemu postoje tri fizička izlaza, M, V i SL, i tri "virtuelna" izlaza: uslov za omogućavanje tajmera, TMR, uslov za taktovanje (inkrementiranje) brojača (označimo ga sa CNT_INC), i uslov za resetovanje brojača (uslov ćemo označiti sa CNT_RES, a *Done* bit brojača sa CDN). Na osnovu ASM dijagrama sledi:

$$M = ST0$$

$$V = TMR = ST1$$

$$SL = ST2$$

$$CNT_INC = ST0 \cdot SB$$

$$CNT_RES = ST2 \cdot TDN \cdot CDN$$

Uočimo da uslovni (tj. Milijevi) izlazi, CNT_INC i CNT_RES, traju tačno jedan sken ciklus, što je svakako dovoljno da obave svoju funkciju.

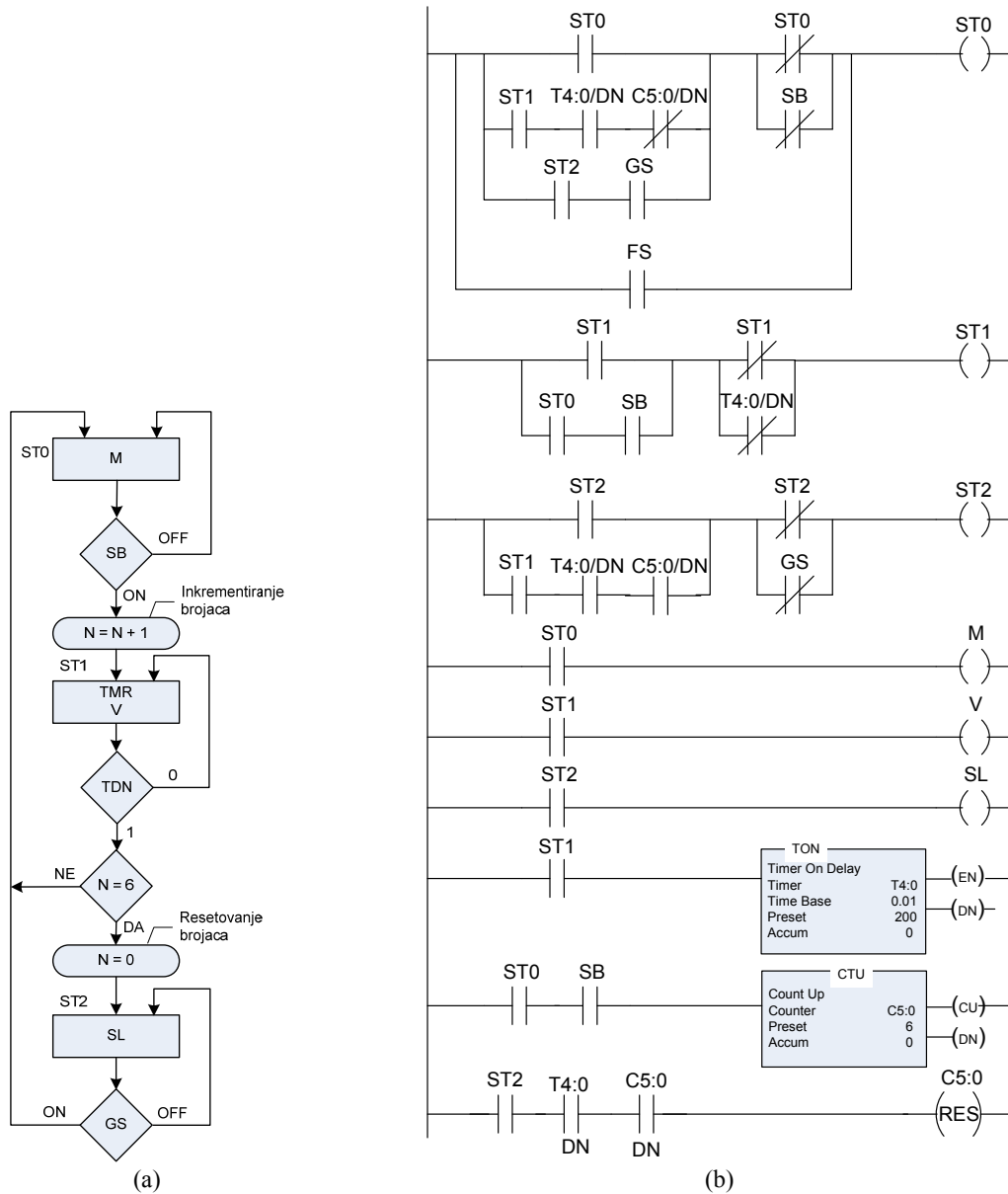
Jednačine stanja su:

$$ST0 = (ST0 + ST1 \cdot TDN \cdot \overline{CDN} + ST2 \cdot GS) \cdot \overline{ST0} \cdot \overline{SB}$$

$$ST1 = (ST1 + ST0 \cdot SB) \cdot \overline{ST1} \cdot \overline{TDN}$$

$$ST2 = (ST2 + ST1 \cdot TDN \cdot \overline{CDN}) \cdot \overline{ST2} \cdot \overline{GS}$$

Leder dijagram je prikazan na Sl. 5-37(b). Primitimo da dijagram može pojednostaviti ako se umesto bitova stanja ST0, ST1 i ST2 koriste bitovi M, V i SL, respektivno (s obzirmo na M=ST0, V=ST1 i SL=ST2). Takođe, primitimo da u leder dijagramu nisu predviđeni posebni bitovi za uslove TMR, CNT_INC i CNT_RES već su oni realizovani direktno u rangovima gde se koriste.



Sl. 5-37 Rešenje zadatka iz Pr. 5-22 : (a) ASM dijagram; (b) leder dijagram

6 Naredbe za operacije nad podacima

U realizaciji različitih algoritama često je potrebno da se izvrše određena izračunavanja, da se prenesu odgovarajuće poruke ili da se u zavisnosti od vrednosti nekih parametara promeni algoritam obrade. U osnovi svih navedenih aktivnosti nalaze se *promenljive – podaci* koji predstavljaju *operande* ili rezultate u različitim matematičkim ili logičkim *operacijama*.

- **Operandi**

Kao što je već rečeno, promenljive se u memoriji kontrolera pamte kao *numerički podaci* ili *alfanumerički podaci – stringovi*. Numerički podaci se pri tome mogu pamtititi kao *celobrojne vrednosti (integers)* ili *decimalni brojevi prikazani u formatu pokretnog zareza (floating point)*. Različiti tipovi numeričkih podataka smeštaju se u *datoteke podataka* odgovarajućeg tipa.

U principu, operandi mogu biti promenljive iz bilo koje datoteke. Potrebno je uočiti, međutim, da iako se dozvoljava korišćenje bit-adresibilnih datoteka (B,I,O), podaci smešteni u njima se u ovim operacijama mogu koristiti samo kao cele reči (elementi), što znači da se operacija ne može izvoditi nad pojedinim bitovima. Pored toga, u datotekama časovnika i brojača (T i C) mogu se kao operandi koristiti samo druga i treća reč elementa koje predstavljaju akumuliranu vrednost (ACC) i zadanu vrednost (PRE). Konačno, kao operandi se mogu javiti i neke promenljive iz kontrolne datoteke (R). O značenju i ulozi ovih promenljivih biće reči kasnije.

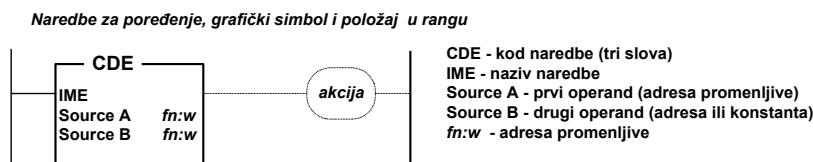
Pored promenljivih, operandi u pojedinim operacijama mogu biti i *programske konstante* – nepromenljive veličine koje se definišu eksplicitnim navođenjem vrednosti u okviru naredbe. Pri tome, nije dozvoljeno da oba operanda budu programske konstante. Samo se po sebi razume da se programska konstanta ne može koristiti kao rezultat.

- **Operacije**

Operacija koja treba da se izvrši nad operandima definiše se u okviru naredbe. Najveći broj ovih naredbi pojavljaju se kao *naredbe akcije*. Ovo je sasvim prirodno ako se ima u vidu da je glavna svrha ovih naredbi da se obavi neka aritmetička ili logička operacija nad operandima i dobijeni rezultat upamti kao odgovarajuća promenljiva. Drugim rečima, sam proces izračunavanja predstavlja jednu *akciju*, čije izvršavanje može biti uslovljeno istinosnom vrednošću nekog *uslova* koji se nalazi u levom delu ranga. Izuzetak su jedino *naredbe za poređenje*, koje opet, po svojoj prirodi, proveravaju da li je neka *relacija* između operanada ispunjena ili nije odnosno da li njena vrednost *istinita* ili *neistinita*. Shodno tome, takve naredbe moraju biti *naredbe uslova*, tako da je rezultat njihovog izvođenja.

6.1 Naredbe za poređenje

Naredbe za poređenje su naredbe *uslova*. U okviru ovih naredbi proverava se istinosna vrednost relacije između dva operanda. Kao rezultat provere naredba dobija vrednost *istinit* ili *neistinit*. Jedna grupa naredbi za poređenje ima oblik kao što je to prikazano na Sl. 6-1. U tabeli T. 6 dat je pregled svih naredbi za poređenje iz ove grupe. *Prvi operand* je uvek *promenljiva*, dok *drugi operand* može biti ili *promenljiva* ili *programska konstanta*.



Sl. 6-1 Opšti izgled naredbe za poređenje.

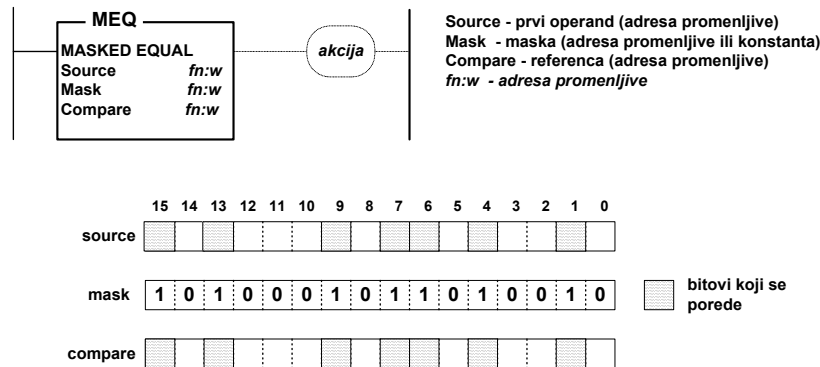
T. 6 Grupa naredbi za poredenje

Kod naredbe	Ime naredbe	relacija istinita ako je	neistinita ako je
EQU	Equal (jednako)	$A = B$	$A \neq B$
NEQ	Not equal (nejednako)	$A \neq B$	$A = B$
LES	Less than (manje)	$A < B$	$A > B$
LEQ	Less than or equal (manje ili jednako)	$A \leq B$	$A \geq B$
GRT	Greater than (veće)	$A > B$	$A < B$
GEQ	Greater than or equal (veće ili jednako)	$A \geq B$	$A \leq B$

Pored navedenih naredbe među naredbama za poredenje postoje i sledeće dve naredbe.

- **MEQ - masked comparison for equal (maskirano ispitivanje jednakosti)**

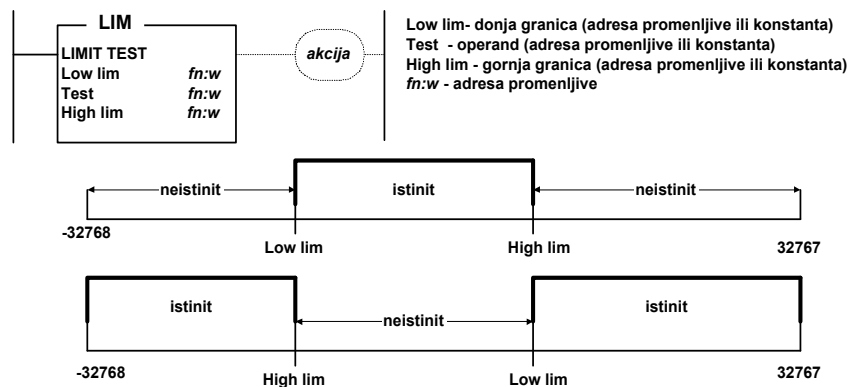
MEQ naredba, grafički simbol, položaj u rangu i realizacija



Ova naredba služi za poredenje delova pojedinih reči. Naime na položaju onih bitova koji ne učestvuju u poredenju (*maskirani bitovi*) u maski se stavljaju *nule*. Ostali bitovi maske, koji odgovaraju bitovima koji se porede (*nemaskirani bitovi*), se postavljaju na 1. Ukoliko su bitovi *operanda* i *reference* koji nisu *maskirani* međusobno jednaki naredba ima vrednost *istinit*. U protivnom ona ima vrednost *neistinit*. Pri definisanju maske, pogodno je koristiti heksadecimalnu konstantu ili promenljivu.

- **LIM – Limit test (ispitivanje granica)**

LIM naredba, grafički simbol, položaj u rangu i realizacija



LIM naredbom se proverava da li se vrednost operanda *Test* nalazi unutar datih granica. Ako je *donja granica* manja od *gornje granice*, vrednost naredbe je *istinita* ako operand pripada segmentu koji određuju granice. Potrebno je obratiti pažnju na činjenicu da “*donja granica*” može biti i veća od

“gornje granice”. U tom slučaju naredba je *istinita* ako se operand nalazi izvan granica ili na njima, a *neistinita* ako operand pripada intervalu koji određuju granice.

Ako je operand *test* konstanta, onda obe granice moraju biti adrese promenljivih. Međutim, ukoliko je *test* adresa promenljive, onda granice mogu biti bilo adrese promenljivih bilo konstante.

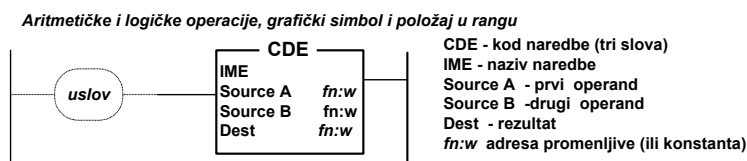
6.2 Matematičke naredbe

Kako im i samo ime kaže, *matematičke naredbe* služe za realizaciju različitih operacija nad operandima. Ove naredbe su naredbe akcije i u najvećem broju slučajeva imaju *dva operanda*. Izvršavanjem naredbe obavlja se zahtevana matematička operacija nad operandima i dobija rezultat čija se vrednost pamti. Operandi mogu biti programske promenljive ili konstante, s tim što oba operanda ne mogu biti konstante.

U odnosu na broj operanada i tip operacije koja se izvršava, matematičke naredbe se mogu podeliti u nekoliko grupa.

6.2.1 Aritmetičke i logičke binarne operacije

Opšti oblik naredbe za aritmetičke i logičke binarne operacije dat je na Sl. 6-2, dok je u tabeli T. 7 dat prikaz svih naredbi iz ove grupe.



Sl. 6-2 Opšti oblik naredbe za aritmetičke i logičke binarne operacije.

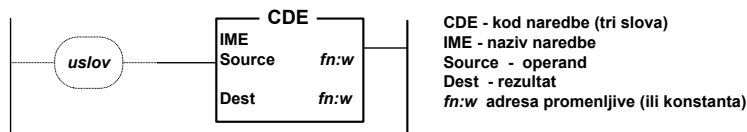
T. 7 Pregled naredbi za aritmetičke i logičke binarne operacije.

Kod naredbe	Ime naredbe	Operacija	Postavlja indikatorske bitove			
			C – bit	V – bit (ako je S:2/14=0)	Z – bit	S – bit
ADD	Add (sabiranje)	$d = a + b$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
SUB	Subtract (oduzimanje)	$d = a - b$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
MUL	Multiply (množenje)	$d = ab$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
DIV	Divide (deljenje)	$d = a/b$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
XPY	X to the power of Y	$d = x^y$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
AND	And (logičko “i”)	$d = a \wedge b$ bit po bit	uvek 0	uvek 0	1 za $d=0$	1 za $d<0$
OR	Or (logičko “ili”)	$d = a \vee b$ bit po bit	uvek 0	uvek 0	1 za $d=0$	1 za $d<0$
XOR	Exclusive OR (ekskluzivno “ili”)	$d = a \oplus b$ bit po bit	uvek 0	uvek 0	1 za $d=0$	1 za $d<0$

6.2.2 Unarne operacije

U grupu unarnih operacija svrstane su i aritmetičke i logičke unarne operacije. Sve ove naredbe imaju isti opšti oblik (Sl. 6-3). Pregled svih naredbi dat je u tabeli T. 8.

Unarne operacije i funkcije, grafički simbol i položaj u rangju



Sl. 6-3 Opšti oblik naredbe za unarne operacije.

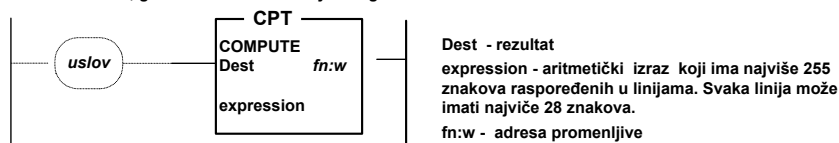
T. 8 Pregled naredbi za unarne operacije.

Kod naredbe	Ime naredbe	Operacija	Postavlja indikatorske bitove			
			C – bit	V – bit (ako je S:2/14=0)	Z – bit	S – bit
NEG	Negate (negacija)	$d = -a$	0 za $d=0$ ili $V=1$	1 za prekoračenje opsega (samo ako je $a=-32768$)	1 za $d=0$	1 za $d<0$
NOT	Not (komplement)	$d = \bar{a}$ bit po bit	uvek 0	uvek 0	1 za $d=0$	1 za $d<0$
DDV	Double divide (deljenje 32-bitnog celog broja iz mat. reg. sa 16-bitnim operandom)	$d = \text{reg}/a$ (rezultat zaokružen)	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
SQR	Square Root (kvadratni koren)	$d = \sqrt{a}$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
ABS	Absolute (apsolutna vrednost)	$d = a $	uvek 0	1 samo ako je $a=-32768$	1 za $d=0$	uvek 0
SIN	Sine	$d = \sin(a)$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
COS	Cosine	$d = \cos(a)$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
TAN	Tangent	$d = \text{tg}(a)$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
ASN	Arc Sine	$d = \arcsin(a)$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
ACS	Arc Cosine	$d = \arccos(a)$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
ATN	Arc Tangent	$d = \text{artg}(a)$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
LN	Natural log (prirodni logaritam)	$d = \ln(a)$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$
LOG	Log to the base 10 (dekadni logaritam)	$d = \log(a)$	uvek 0	1 za prekoračenje opsega	1 za $d=0$	1 za $d<0$

6.2.3 Složene matematičke naredbe

• CPT – Compute (izračunavanje aritmetičkog izraza)

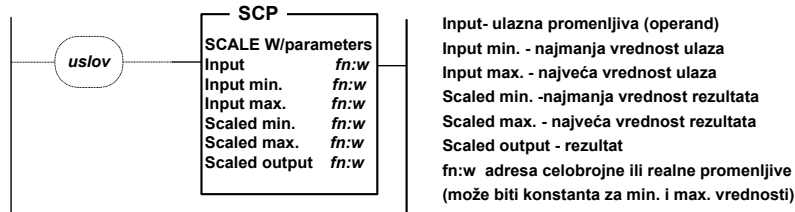
CPT naredba, grafički simbol i položaj u rangju



Pri formiranju izraza koriste se promenljive, konstante i sledeći operatori: – ili NEG (unarni minus), +, -, *, | ili DIV, ** ili XPY, SQR, ABS, SIN, COS, TAN, ASN, ACS, ATN, LN, LOG, NOT, AND, OR, XOR, TOD, FRD, DEG, RAD.

- **SCP – Scale with parameters (parametarsko skaliranje podatka)**

SCP naredba, grafički simbol i položaj u rangu



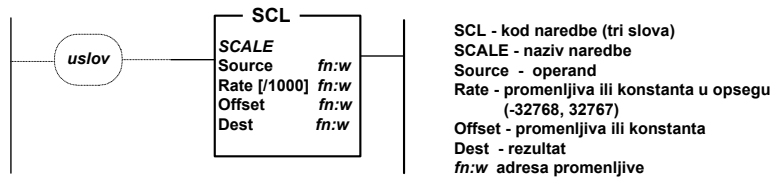
Ovom naredbom se ostvaruje linearno preslikavanje ulazne promenljive, prema relaciji

$$\text{scaled_output} = \frac{\text{scaled_max.} - \text{scaled_min.}}{\text{input_max.} - \text{input_min.}} \times (\text{input} - \text{input_min.}) + \text{scaled_min.}$$

Potrebno je da se istakne da se u ovoj naredbi pojmovi “najmanja i najveća vrednost ulaza i rezultata” zapravo koriste samo za određivanje dve tačke kroz koje se provlači prava linija koja određuje preslikavanje. To znači da vrednost ulazne promenljive ne mora biti unutar intervala određenog sa (*input_min.*, *input_max.*), niti da vrednost *input_min.* mora biti manja od *input_max.*

- **SCL - Scale data (skaliranje podatka)**

SCL naredba, grafički simbol i položaj u rangu



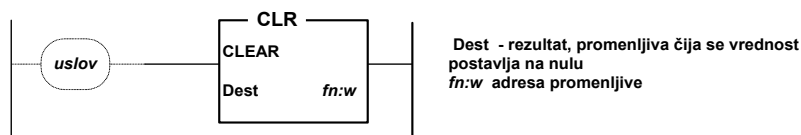
Ova naredba je slična naredbi SCP, jer se i pomoću nje ostvaruje linearno skaliranje promenljive. Razlika je samo u načinu definisanja parametara skaliranja. Skaliranje ulaza se izvodi prema sledećoj relaciji:

$$\text{Dest} = \frac{\text{Rate}}{1000} \times \text{Source} + \text{Offset}$$

6.2.4 Naredbe za manipulaciju sa numeričkim podacima

- **CLR – Clear (postavi na nulu)**

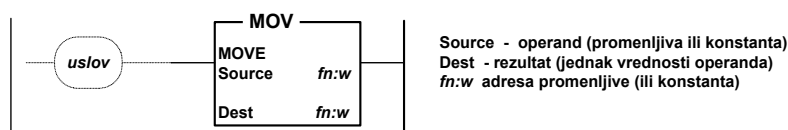
CLR naredba, grafički simbol i položaj u rangu



Adresirana promenljiva se postavlja na nulu.

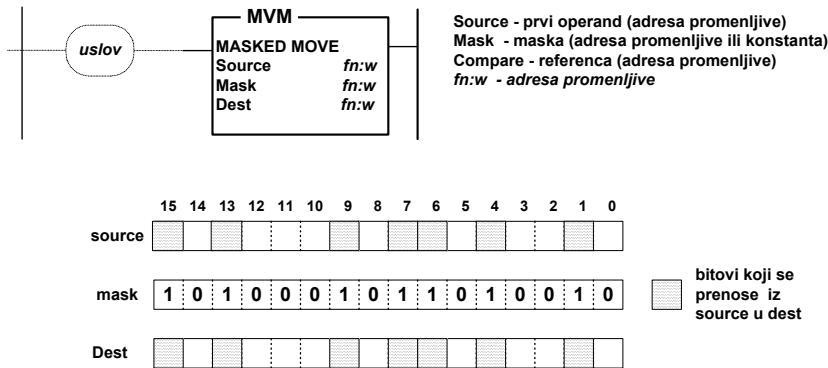
- **MOV – Move (postavljanje vrednosti promenljive)**

Mov naredba, grafički simbol i položaj u rangu



- **MVM – Masked move (postavljanje vrednosti pojedinih bitova)**

MVM naredba, grafički simbol, položaj u rangu i realizacija



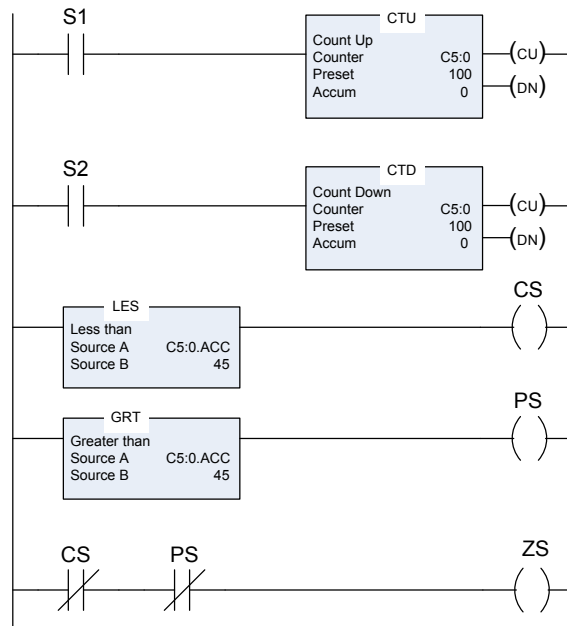
Maskiranim bitovima, koji se ne prenose u *dest* u maski odgovaraju vrednosti 0, dok nemaskiranim bitovima odgovaraju vrednosti 1.

6.3 Primeri

Pr. 6-1 Brojanje predmeta na pokretnoj traci (varijanta 1)

Zadatak: Senzor S1 se koristi za brojanje predmeta koji se stavljaju na početak pokretne trake, a senzor S2 predmete koji napuštaju pokretnu traku. Ako je na traci trenutno manje od 45 predmeta, svetli crvena svetiljka (CS), ako je na traci tačno 45 predmeta, svetli zelena (ZS) i ako je broj predmeta na traci veći od 45, svetli plava svetiljka (PS).

Rešenje: Koristimo dva brojača: jedan tipa CTU (broji napred) koji broji impulse sa senzora S1 i drugi tipa CTD (broji unazad) koji broji impulse sa senzora S2 (Sl. 6-4). Treba zapaziti da oba brojača dele isti element iz datoteke brojača, C5:0. To znači da će oba brojača ažurirati isti registar (ACC) kada odbrojavaju impulse. Svaki impuls sa senzora S1 uvećaće vrednost u C5:0.ACC za jedan, dok će svaki impuls sa senzora S2 umanjiti ovu vrednost za 1. Dakle, tražena informacija je prisutna u registru C5:0.ACC. Za poređenje koristimo naredbe LES (manje) i GRT (veće), koje porede sadržaj akumulatora brojača C5:0 sa 45. Kao što se može videti, umesto treće naredbe poređenja (EQU - jednako) koja bi pobuđivala zelenu sijalicu koristi se uslov da ako broj proizvoda na traci nije ni veći ni manji od 45, tada mora biti jednak 45.

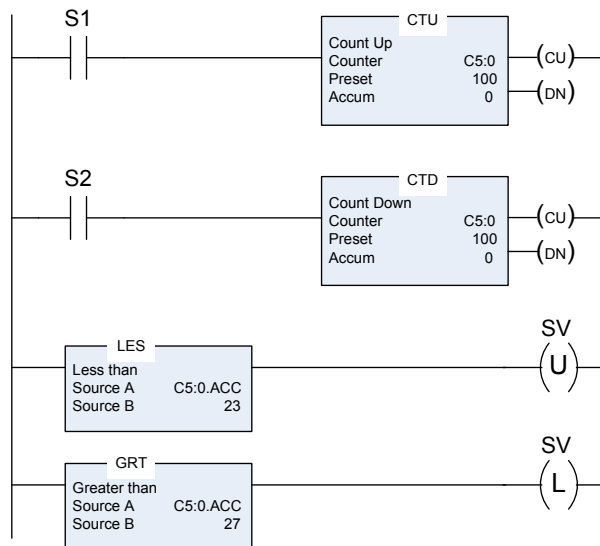


Sl. 6-4 Rešenje zadatka iz Pr. 6-1.

Pr. 6-2 Brojanje predmeta na pokretnoj traci (varijanta 2)

Zadatak: Senzor S1 se koristi za brojanje predmeta koji se stavljaju na početak pokretne trake, a senzor S2 predmete koji napuštaju pokretnu traku. Svetiljka SV se pali kada broj predmeta na traci postane veći od 27, a gasi kada postane manji od 23.

Rešenje: Za brojanje predmeta na traci koriste se dva brojača na identičan način kao u prethodnom primeru. Svetiljkom SV se upravlja pomoću naredbi za setovanje/resetovanje. Ako je broj predmeta na traci manji od 23, SV će biti resetovana (ugašena) naredbom LES, a ako je veći od 27, svetiljka će biti setovana (upaljena) naredbom GRT. Ako je broj predmeta između 23 i 27, svetiljka će zadržati svoje stanje (upaljeno/ugašeno) sve dok ovaj broj ne izađe iz datog opsega.

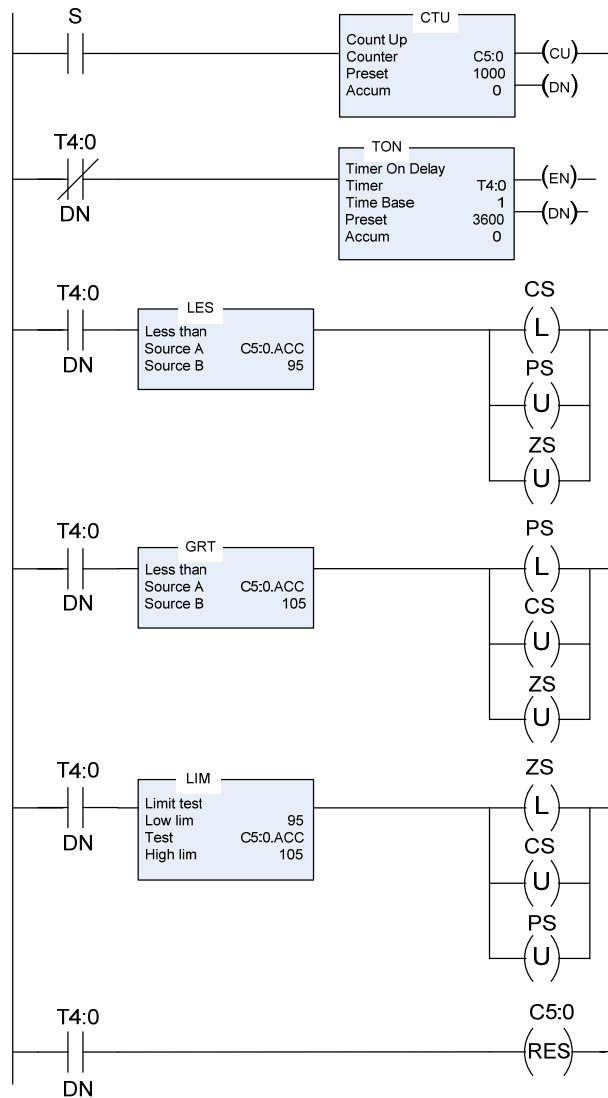


Sl. 6-5 Rešenje zadatka iz Pr. 6-2.

Pr. 6-3 Kolika je produktivnost?

Zadatak: Senzor S registruje predmete koji napuštaju pokretnu traku. Na svaki sat (1h), pali se jedna od tri sijalice, crvena (CS), zelena (ZS) ili plava (PS). Ako je broj predmeta koji su u prethodnom satu napustili traku manji od 95, pali se crvena, ako je u granicama između 95 i 105, pali se zelena i ako je veći od 105 pali se plava svetiljka.

Rešenje: Rešenje koristi tajmer za odmeravanje vremena od 1h, brojač za brojanje predmeta koji napuštaju traku i naredbe poređenja koje ispituju uslove za paljenje sijalica. U ovom primeru vidimo da se naredbe poređenja mogu kombinovati sa drugim naredbama uslova u istom rangu. U trenutku kada tajmer odmeri zadato vreme (DN=1), a u zavisnosti od trenutne vrednosti brojača, uslov u samo jednom od tri ranga koji ispituju vrednost brojača biće tačan, što će upaliti odgovarajuću svetiljku, a ugasiti ostale dve. Takođe, DN=1, resetuje brojač i tajmer čime započinje odmeravanje vremena i brojanje proizvoda za sledeći sat.

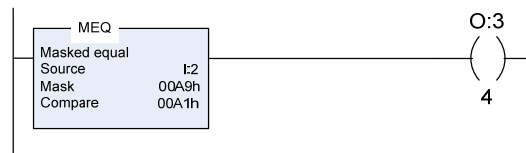


Sl. 6-6 Rešenje zadatka iz Pr. 6-3.

Pr. 6-4 Primena MEQ naredbe

Zadatak: Na pinove 0, 3, 5 i 7, ulaznog modul, smeštenog u slotu 2, povezana su četiri ON/OFF davača. Pobuditi svetiljku, povezanu na pin 4 izlaznog modula smeštenog u slotu 3, ako su stanja davača, redom, 1, 0, 1, 1. Koristiti naredbu MEQ.

Rešenje: Naredba za maskirano ispitivanje jednakosti (MEQ) omogućava da se uz pomoć samo jedne naredbe ispita složeniji logički uslov. Maska ukazuje na bitove iz izvorne reči koji se ispituju. Pošto ispitujuemo bitove 0, 3, 5 i 7, maska, u binarnom obliku je: 0000000010101001, odnosno 00A9 u heksadecimalnoj notaciji. Parametar *Compare* sadrži očekivane vrednosti bitova izabranih maskom, tj. imaće vrednosti 1, 0, 1, 1 na pozicijama 0, 3, 5 i 7, respektivno: 0000000010100001, odnosno 00A1h.



Sl. 6-7 Rešenje zadatka iz Pr. 6-4.

7 Analogni PLC moduli

7.1 Analogni ulazni moduli

Analogni ulazni moduli prihvataju analognu informaciju sa analognih senzora i konvertuju u digitalnu informaciju za dalju obradu u PLC kontroleru. Zato se zovi i A/D moduli. Postoje dva osnovna tipa A/D modula: sa strujnim i sa naponskim ulazom.

Analogni moduli sa naponskim ulazom dostupni su dve varijante: unipolarni i bipolarni. Unipolarni moduli privataju napon jednog polariteta, na primer od 0 do +10V. Bipolarni moduli mogu prihvatiti ulazni napon oba polariteta, na primer od -10V do +10V.

Analogni moduli sa strujnim ulazom, standardno prihvataju struju u opsegu od 4mA do 20mA, pri čemu ulazna struja od 4mA predstavlja najmanju, a ulazna struja od 20mA najveću ulaznu vrednost. (Uočimo da dozvoljeni opseg ulazne struje ne obuhvata 0A. To je učinjeno iz razloga kako bi se omogućila detekcija prekida veze između senzora i ulaznog modula. Ako veza postoji, između senzora i ulaznog modula uvek teče struja veća od 4mA. Ako je veza u prekidu, struje nema.)

Većina analognih ulaznih modula se može konfigurisati od strane korisnika. Konfiguracijom se podešava naponski ili strujni opseg. Postoje moduli i sa naponskim i sa strujnim ulazima, tako da korisnik može koristiti ulaz koji odgovara konkretnoj primeni.

Tipično, A/D moduli za industrijske primene imaju 12-bitnu rezoluciju, što znači da se puni opseg ulaznog napona ili struje preslikava na $4096 (=2^{12})$ digitalnih reči.

Tipično, analogni moduli imaju od jednog do osam nezavisnih analognih ulaza ili kanala.

Dostupni su i analogni moduli specijalne namene, kao što su moduli za spregu sa termoparovima. Radi se o A/D modulu koji je prilagođen za prihvatanje izlaza termopara. Izlazni signal termopara je veoma malog napona, što znači da mora biti pojačan pre nego što se obavi A/D konverzija. Konverzija napona termopara u temperaturu zahteva određene kompenzacije i korekcije, što je takođe funkcija koju obavlja ovaj specijalizovan analogni ulazni modul.

7.2 Analogni izlazni moduli

Namena analognih izlaznih modula je konverzija digitalne izlazne informacije koju generiše PLC kontroler u analogni napon ili struju koja se koristi za upravljanje nekog specifičnog izlazni uređaj. Na primer, analogni izlaz se može koristiti za upravljanje brzinom rotacije motora ili upravljanje strujom kroz grejač. Dostupni su analogni izlazni moduli sa naponskim ili strujnim izlazom. Analogni izlazni moduli generišu signale čiji napon ili struja može biti proizvoljan u okviru fiksnog opsega. Tipičan opseg izlaznog napona je 0 do 10V, a opseg struja 4mA do 20mA.

Pored analognih PLC modula koji su isključivo ulazni ili isključivo izlazni, u upotrebi su i kombinovani analogni U/I moduli, koji poseduju određeni broj analognih ulaza i analognih izlaza (npr. 2 ulaza i 2 izlaza).

7.3 SLC 500 analogni U/I moduli

Serijski analogni U/I moduli predviđenih za spregu sa PLC kontrolerima tipa SLC 500 nose oznaku 1746 i sadrži 9 različitih modula:

- NI4 ulazni
- NIO4I, NIO4V, FIO4I i FIO4V ulazno/izlazni moduli
- NO4I i NO4V izlazni moduli
- NT4 ulazni modul za spregu sa termoparom
- NR4 ulazni modul za spregu sa otpornim temperaturnim senzorima (RTD)

Prvo slovo u oznaci modula označava rezoluciju. Moduli sa oznakom N poseduju 16-bitnu, a moduli sa oznakom F 12-bitnu ulaznu (A/D) rezoluciju. Izlazna rezolucija (D/A) kod svih modula je 14-bitna. Drugo, odnosno drugo i treće slovo ukazuju na tip modula: I – ulazni, O- izlazni, IO-ulazno/izlazni modul, T-modul za spregu sa termoparovima, R-modul za spregu sa RTD senzorima. Broj koji sledi ukazuje na broj analognih kanala, koji je kod svih modula iz ove serije 4 (IO moduli imaju 2 ulazna i 2 izlazna kanala). Poslednje slovo ukazuje na tip analognih izlaza: V – naponski analogni izlazi; I – strujni naponski izlazi. Svaki ulazni analogni kanal se može konfigurisati kao naponski (opseg: -10V - +10V) ili strujni ulaz (-20mA - +20mA). Detaljna specifikacija A/D i D/A karakteristika analognih modula iz serije 1746 data je na Sl. 7-1 i Sl. 7-2.

NI4, NIO4I, & NIO4V Input Range	Decimal Range (input image table)	Number of Significant Bits	Nominal Resolution
±10V dc -1 LSB	-32,768 to +32,767	16	305.176 mV/LSB
0 to 10V dc -1 LSB	0 to 32,767	15	
0 to 5V dc	0 to 16,384	14	
1 to 5V dc	3,277 to 16,384	13.67	
±20 mA	±16,384	15	1.22070 mA/LSB
0 to 20 mA	0 to 16,384	14	
4 to 20 mA	3,277 to 16,384	13.67	

FIO4I and FIO4V Input Range	Decimal Range (input image table)	Number of Significant Bits	Nominal Resolution
0 to 10V dc -1 LSB	0 to 4095	12	2.4414 mV/LSB
0 to 5V dc	0 to 2047	11	
1 to 5V dc	409 to 2047	10.67	
0 to 20 mA	0 to 2047	11	9.7656 µA/LSB
4 to 20 mA	409 to 2047	10.67	

Sl. 7-1 A/D karakteristike ulaza analognih modula iz serije 1746. (Značenje kolona: Ulazni opseg, Decimalni opseg, Broj značajnih bita, Nominalna rezolucija).

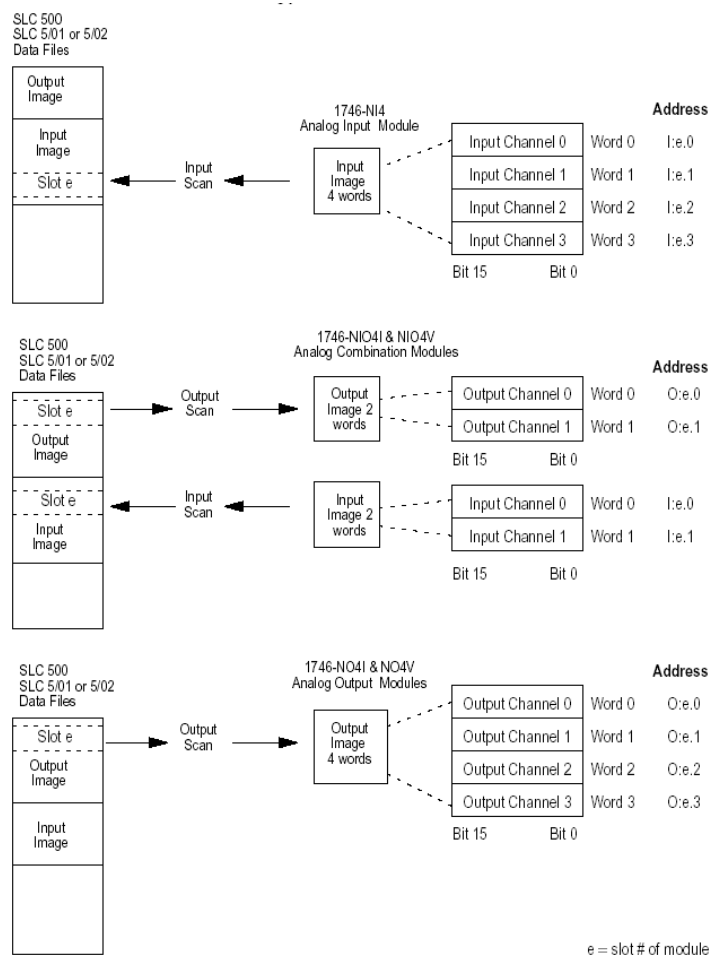
Module	Output Range	Decimal Range (output image table)	Significant Bits	Resolution
FIO4I	0 to 21 mA - 1 LSB	0 to 32,764	13 bits	2.56348 µA/LSB
NIO4I	0 to 20 mA	0 to 31,208	12.92 bits	
NO4I	4 to 20 mA	6,242 to 31,2089	12.6 bits	
FIO4V	±10V dc - 1 LSB	-32,768 to +32,764	14 bits	1.22070 mV/LSB
NIO4V	0 to 10V dc -1 LSB	0 to 32,764	13 bits	
NO4V	0 to 5V dc	0 to 16,384	12 bits	
	1 to 5V dc	3,277 to 16,384	11.67 bits	

Sl. 7-2 D/A karakteristike izlaza analognih modula iz serije 1746. (Značenje kolona: Modul, Izlazni opseg, Decimalni opseg, Broj značajnih bitova, Rezolucija).

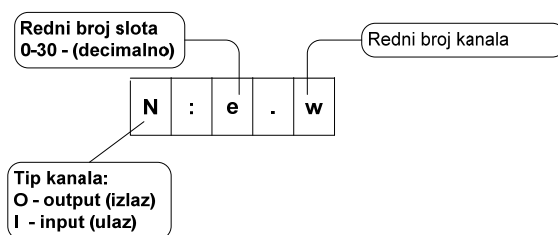
7.4 Sprezanje analognih modula sa kontrolerom

Analogni moduli se ugrađuju u slot reka i direktno su spregnuti sa PLC kontrolerom preko sistemske magistrale. Komunikacija između PLC kontrolera i analognih modula je pod kontrolom operativnog sistema i obavlja se automatski (Sl. 7-3). Svakom ulazom kanalu priključenog analognog modula odgovara jedna 16-bitna reč u slici ulaza, a svakom izlaznom kanalu jedna reč u slici izlaza PLC kontrolera. U svakom ulaznom skenu (*input scan*), konvertovane vrednosti analognih signala prisutnih na ulaznim kanalima (*input channels*) analognog modula se prenose na odgovarajuće mesto u slici ulaza (*input image*). U toku izlaznog skena (*output scan*) vrednosti iz odgovarajućih reči slike

izlaza (*output image*) se šalju analognim modulima gde se konvertuju u analogne signale. Način adresiranja pojedinačnih kanala analognog modula iustrovan je na Sl. 7-4. Na primer, izlazni kanal 0, modula NIO4I koji je smešten u slotu 3, adresira se sa: O:3.0.



Sl. 7-3 Sprega PLC i analognih modula.



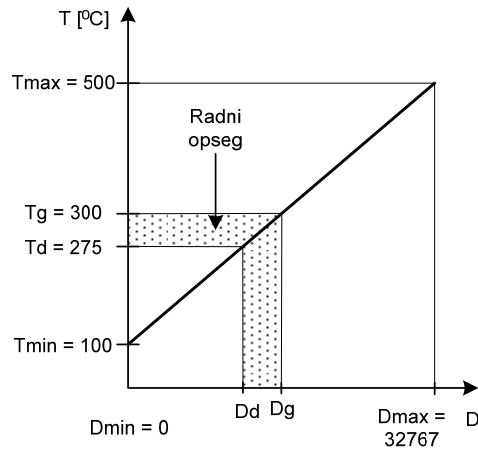
Sl. 7-4 Adresiranje analognih modula.

7.5 Primeri

Pr. 7-1 Adresiranje, ispitivanje opsega i skaliranje analognog ulaza.

Zadatak: Analogni modul NI4 smešten je u slotu 1 modularnog PLC kontrolera. Temperaturni senzor sa naponskim izlazom opsega 0-10V povezan je na drugi ulazni kanal analognog modula. Izlazni napon temperaturnog senzora direktno je srazmeran temperaturi iz opsega 100°C (0V) – 500°C (10V). Temperatura procesa mora ostati u opsegu 275°C-300°C. Ako je temperatura izvan ovog opsega, treba aktivirati alarm koji je vezan za pin 0 diskretnog izlaznog modula smeštenog u slotu 2. Vrednost temperature izraženu u fizičkim jedinicama (stepenima Celzijusa) treba smestiti u reč N7:0 radi dalje obrade.

Rešenje: Analogi modul NI4 obavlja 16 bitnu A/D konverziju za opseg ulaznog napona -10V - +10V. Rezultujuća izlazna digitalna reč dostupna je u formatu dvojičnog komplementa. Pošto je u konkretnom primeru ulazni napon unipolaran, opseg izlaznih reči biće: od 0 (za 0V) do $32767=2^{15}-1$ (za 10V). Između digitalnih reči i temperature postoji linearna zavisnost, kao što je prikazano na Sl. 7-5. Na dijagramu sa Sl. 7-5 takođe je naznačen radni temperaturni opseg. Najnižoj temperaturi radnog opsega odgovara digitalna reč D_d (donja granica), a najvišoj digitalna reč D_g (gornja granica).



Sl. 7-5 Zavisnost između ulazne digitalne reči i temperature.

Vrednosti D_d i D_g možemo izračunati na sledeći način:

$$T = \frac{T_{\max} - T_{\min}}{D_{\max} - D_{\min}} (D - D_{\min}) + T_{\min} \text{ - jednačina prave } T(D)$$

za konkretne vrednosti dobijamo:

$$T = \frac{500 - 100}{32767 - 0} (D - 0) + 100 = \frac{400}{32767} D + 100 [^{\circ}\text{C}]$$

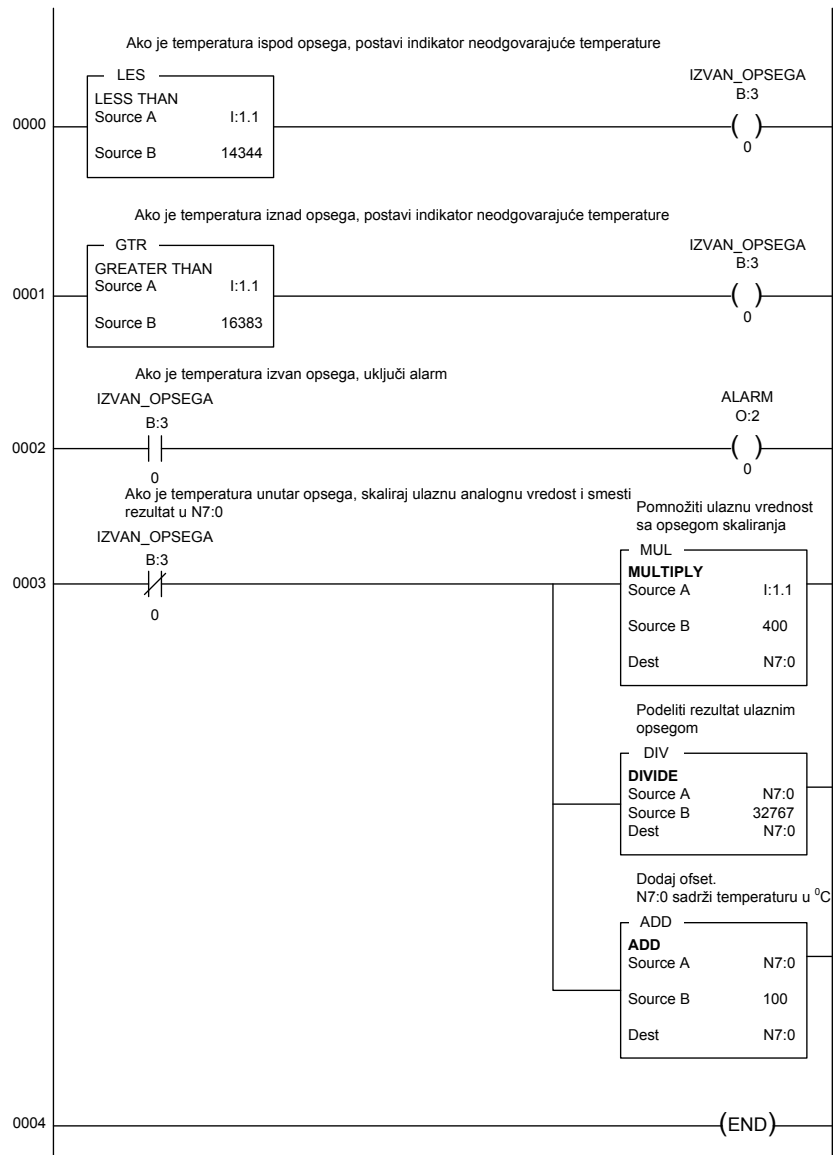
odnosno:

$$D = (T - 100^{\circ}\text{C}) \frac{32767}{400}, \text{ odakle sledi}$$

$$D_d = D(275) = 14344$$

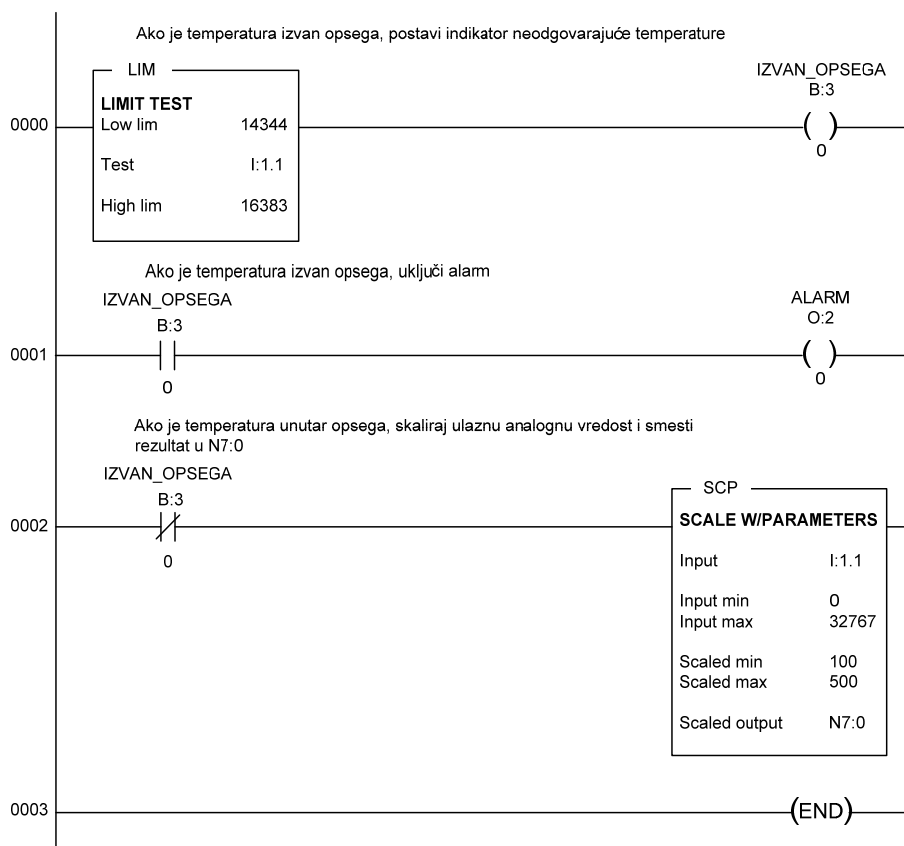
$$D_g = D(300) = 16383$$

Na Sl. 7-6 prikazan je lader program koji ispituje da li ulazna digitalna reč pripada radnom opsegu; aktivira alarm ako je temperatura izvan opsega, odnosno, obavlja konverziju ulazne digitalne reči u $^{\circ}\text{C}$, ako je temperatura u radnom opsegu. Lader program sa Sl. 7-6 koristi naredbe poređenja i standardne matematičke naredbe.



Sl. 7-6 Skaliranje analogne ulazne vrednosti (relizacija pomoću standardnih aritmetičkih naredbi).

Leder program se može pojednostaviti ako se koristi naredba za ispitivanje granica (LIM) i naredba za linearno parametarsko skaliranje podataka (SCP), kao što je prikazano na Sl. 7-7.



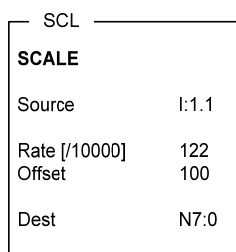
Sl. 7-7 Skaliranje analogne ulazne vrednosti (relizacija pomoću LIM i SCP naredbi).

Umesto naredbe SCP, za linearno skaliranje ulazne analogne vrednosti možemo koristiti i naredbu SCL. Prethodno je potrebno odrediti parametre **Rate** i **Offset** naredbe SCL:

$$Rate = (400/32767) \times 10000 = 122$$

$$Offset = 100$$

Na Sl. 7-8 je prikazan blok naredbe SCL sa uvršćenim konkretnim parametrima.



Sl. 7-8 SCL naredba koja odgovara SCP naredbi iz leder dijagrama sa Sl. 7-7

Pr. 7-2 Skaliranje izlazne analogne veličine

Zadatak: Neka je u slotu 2 modularnog PLC kontrolera prisutan analogni modul NIO4I. Aktuator električnog ventila povezan je na izlazni kanal 0 analognog modula. Aktuator se pobuđuje strujom opsega 4mA – 20mA. Otvor ventila je proporcionalan struji, tako da struja od 4mA potpuno zatvara ventil (otvorenost ventila od 0%), a struja od 20mA potpuno otvara ventil (otvorenost ventila od 100%). Aktuator ventila ne sme biti pobuđivan strujom koja je izvan zadatog opsega. Nacrtati leder dijagram za određivanje izlazne analogne vrednosti, pod pretpostavkom da je procenat otvorenosti ventila (0-100%) dostupan u registru N7:0.

Rešenje: Analogni modul NIO4I poseduje dva analogna strujna izlaza. Opseg izlazne struje je -21mA - +21mA. Ulazne digitalne vrednosti su 16 bitne u formatu dvojičnog komplementa. U konkretnom primeru, izlazna struja je ograničena na pozitivan podopseg 4mA-20mA. Granicama dozvoljenog opsega izlazne struje odgovaraju sledeće digitalne vrednosti:

$$4\text{mA} \rightarrow (32767 / 21) \times 4 = 6242 \quad (\text{scaled_min})$$

$$20\text{mA} \rightarrow (32767 / 21) \times 20 = 31208 \quad (\text{scaled_max})$$

Zavisnost skalirane izlazne vrednosti, *scaled_value* (digitalna vrednost koja se šalje analognom modulu) od ulazne vrednosti, *input_value*, (procenat otvorenosti ventila) prikazana je na Sl. 7-9, a određena je sledećom jednačinom:

$$\text{scaled_value} = (\text{input_value} \times \text{slope}) + \text{offset},$$

gde je *slope* nagib prave određene krajnjim tačkama ulaznog i izlaznog opsega:

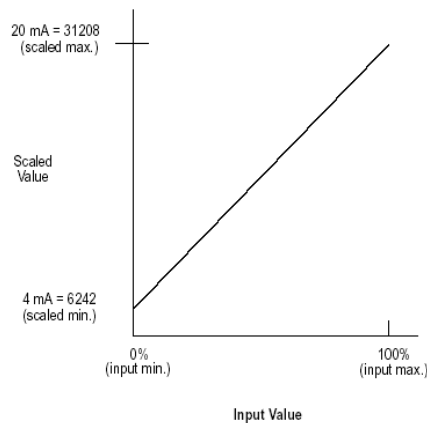
$$\text{slope} = (\text{scaled_max} - \text{scaled_min}) / (\text{input_max} - \text{input_min})$$

$$= (31208 - 6242) / (100 - 0) = 24966 / 100$$

dok je *offset*:

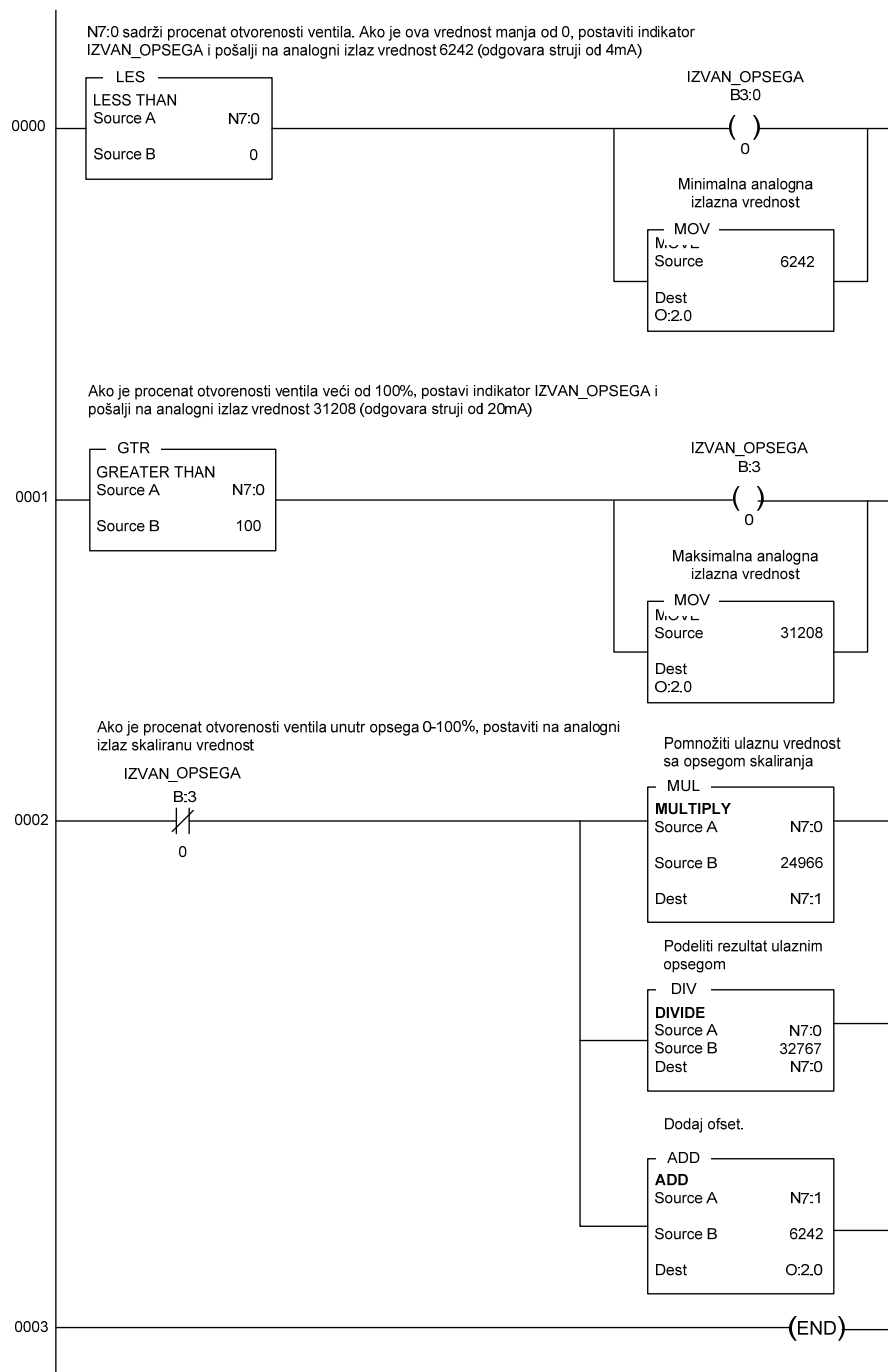
$$\text{offset} = \text{scaled_min} - (\text{input_min} \times \text{slope}) = 6242 - (0 \times (24966 / 100)) = 6242$$

Dakle, ako je *input_value* u opsegu dozvoljenih vrednosti (0-100), *scaled_value* (vrednost koja se šalje analognom modulu) određuje se shodno izvednoj jednačini. Ako je *input_value* manje od 0% (podkoračenje), *scaled_value* se postavlja na 6242 (*scaled_min*). Ako je *input_value* veće od 100% (prekoračenje), *scaled_value* se postavlja na 31208 (*scaled_max*).

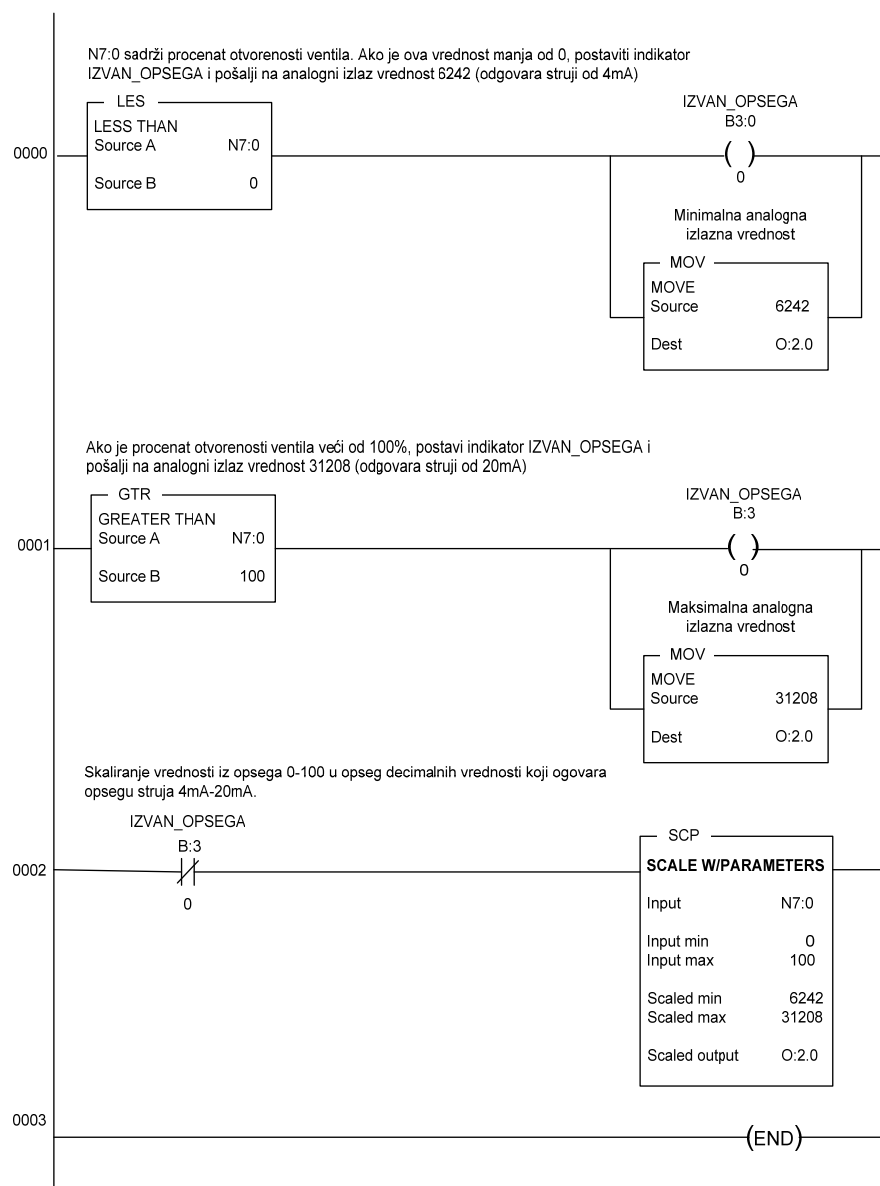


Sl. 7-9 Zavisnost izlazne vrednosti od zadate procentualne otvorenosti ventila.

Treba uočiti da obe vrednosti *slope* i *offset* pripadaju dozvoljenom opsegu celih brojeva (integers), (-32768 - +32767), te je stoga za izračunavanje skalirane izlazne vrednosti moguće direktno primeniti standardne matematičke naredbe. Leder program koji koristi standardne matematičke naredbe i naredbe poredjenja prikazan je na Sl. 7-10. Na Sl. 7-11 je prikazan leder program kod koga se za skaliranje izlazne vrednosti koristi SCP naredba.



Sl. 7-10 Leder program za pobudu aktuatora ventila (varijanta sa matematičkim naredbama).



Sl. 7-11 Leder program za pobudu aktuatora ventila (varijanta sa SCP naredbom).

Pr. 7-3 Skaliranje za ofset veći od 32767 ili manji od -32768

Zadatak: Neka su uslovi iz prethodnog primera promenjeni tako da opsegu struja 4mA-20mA odgovara otvorenost ventila od 90-100%. Analogni modul NIO4I postavljen je u slot 2, a aktuator električnog ventila povezan je na izlazni kanal 0 analognog modula.

Rešenje: Na Sl. 7-12 prikazana je zavisnost skalirane izlazne vrednosti od ulazne vrednosti. Kao što se može videti, minimalna ulazna vrednost (*input_min*) iznosi 90% i njoj odgovara minimalna skalirana izlazna vrednost *scaled_min*=6242 (za struju od 4mA). Maksimalna ulazna vrednost iznosi *input_max*=100% i njoj odgovara maksimalna izlazna skalirana vrednost *scaled_max*=31208 (za struju od 20mA). Na osnovu ovih podataka možemo odrediti jednačinu skaliranja:

$scaled_value = (input_value \times slope) + offset$, gde je:

$$slope = \frac{scaled_max - scaled_min}{input_max - input_min} = \frac{31208 - 6242}{100 - 90} = 24966/10$$

$$offset = scaled_min - (input_min \times slope) = 6242 - (90 \times (24966/100)) = -218452, \text{ odnosno}$$

$$scaled_value = (input_value) \times (24966/10) - 218452$$

Uočimo da je vrednost *offset*-a manja od -32768, odnosno manja od najmanje negativne vrednosti koja se može predstaviti sa 16 bita što onemogućava direktnu primenu matematičkih naredbi.

Da bi se pojednostavilo izračunavanje, linearna zavisnost se može translirati duž ose ulaznih vrednosti ka koordinatnom početku, kao na Sl. 7-13. Ova zavisnost se dobija nakon uvedene smene

$$input_value = input_value - input_min.$$

S obzirom na to, jedančina skaliranja sada glasi:

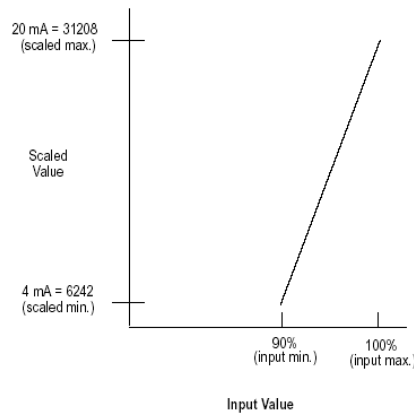
$$scaled_value = ((input_value - input_min) \times slope) + offset, \text{ gde je:}$$

$$slope = (scaled_max - scaled_min) / (input_max - input_min) = (31208 - 6242) / (100 - 90) = 24966/10$$

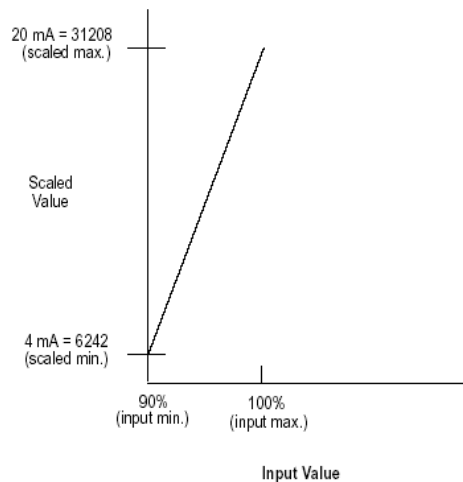
$$offset = scaled_min = 6242$$

$$scaled_value = (input_value - 90) \times (24966/10) + 6242$$

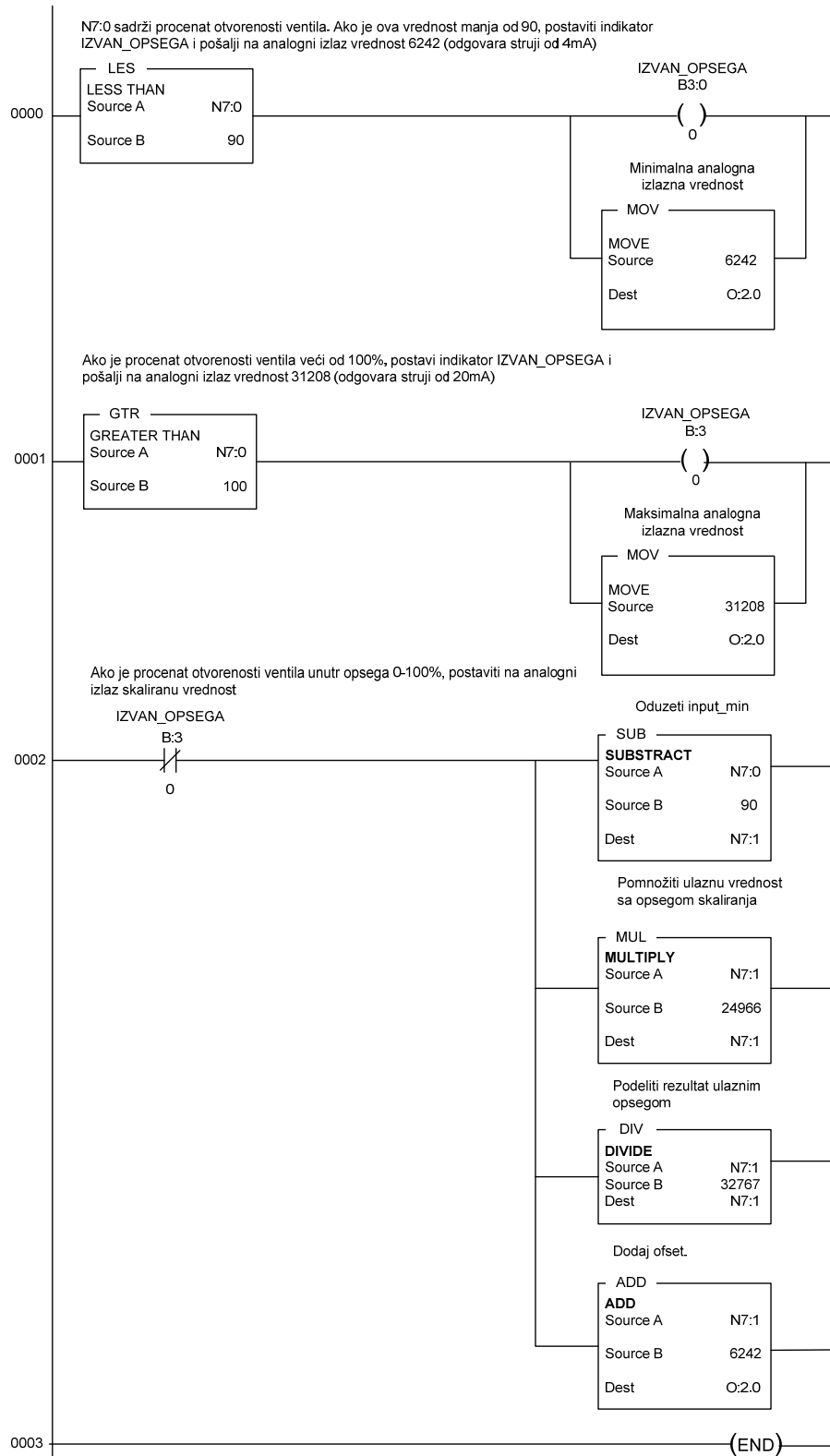
Očigledno, *offset* je smanjen i može se predstaviti kao integer. Odgovarajući leder program prikazan je na Sl. 7-14. Kao i u prethodnim primerima, umesto standardnih matematičkih naredbi možemo koristiti naredbu za skaliranje podatak SCP ili SCL.



Sl. 7-12 Zavisnost skaliranja sa velikom negativnom vrednošću *offset*-a.



Sl. 7-13 Translirana zavisnost skaliranja.



Sl. 7-14 Leder program za transliranu zavisnost skaliranja.

Pr. 7-4 Analogni ulaz/izlaz

Zadatak: U slotu 1 modularnog PLC kontrolera smešten je analogni modul NIO4V. Senzor pritiska za opseg 0 – 200 psi (paskala) povezan je na ulazni kanal 0 analognog modula. Senzor pritiska daje struju 4mA-20mA, tako da struja od 4mA odgovara pritisku od 0psi, a struja od 20mA odgovara

pritisku od 200psi. Ulaznu vrednost dobijenu od analognog modula treba najpre testirati kako bi se proverilo da li pripada dozvoljenom opsegu, a zatim skalirati na opseg napona 0-2.5V i proslediti voltmetru koji je povezan na izlazni kanal 0 analognog modula. Ako je očitana ulazna vrednost izvan dozvoljenog opsega, treba postaviti indikator IZVAN_OPSEGA.

Rešenje: Opseg pune skale ulaznog kanala analognog modula NIO4V, konfigurisanog kao strujni ulaz, iznosi +/-20mA. Opsegu pune skale odgovara decimalni opseg +/-16384. Na osnovu ovih podataka, lako dolazimo do graničnih vrednosti decimalnog opsega koji odgovara ulaznoj struji od 4mA do 20mA:

$$input_min = (16384/20mA) \times 4mA = 3277$$

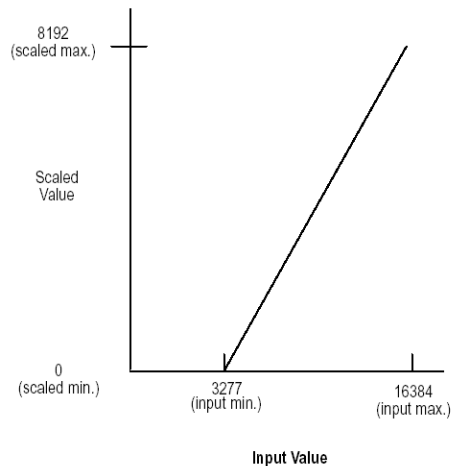
$$input_max = (16384/20mA) \times 20mA = 16384$$

Izlazni kanal analognog modula NIO4V, na koji je povezan volmetar, je naponskog tipa sa opsegom pune skale +/-10V. Opsegu pune skale odgovara decimalni opseg -32768 - +32764. Na osnovu ovih podataka možemo odrediti decimalni opseg koji odgovara željenom naponskom opsegu od 0-2.5V:

$$scaled_min = 0$$

$$scaled_max = (32764/10V) \times 2.5V = 8192$$

Na Sl. 7-15 je prikazana zavisnost između decimalnih vrednosti ulazne veličine (*input_value*) i rezultujućih skaliranih vrednosti izlazne veličine (*scaled_value*).



Sl. 7-15 Zavisnost skaliranja.

Jednačina skaliranja je oblika:

$scaled_value = (input_value \times slope) + offset$, gde je:

$$slope = (scaled_max - scaled_min) / (input_max - input_min)$$

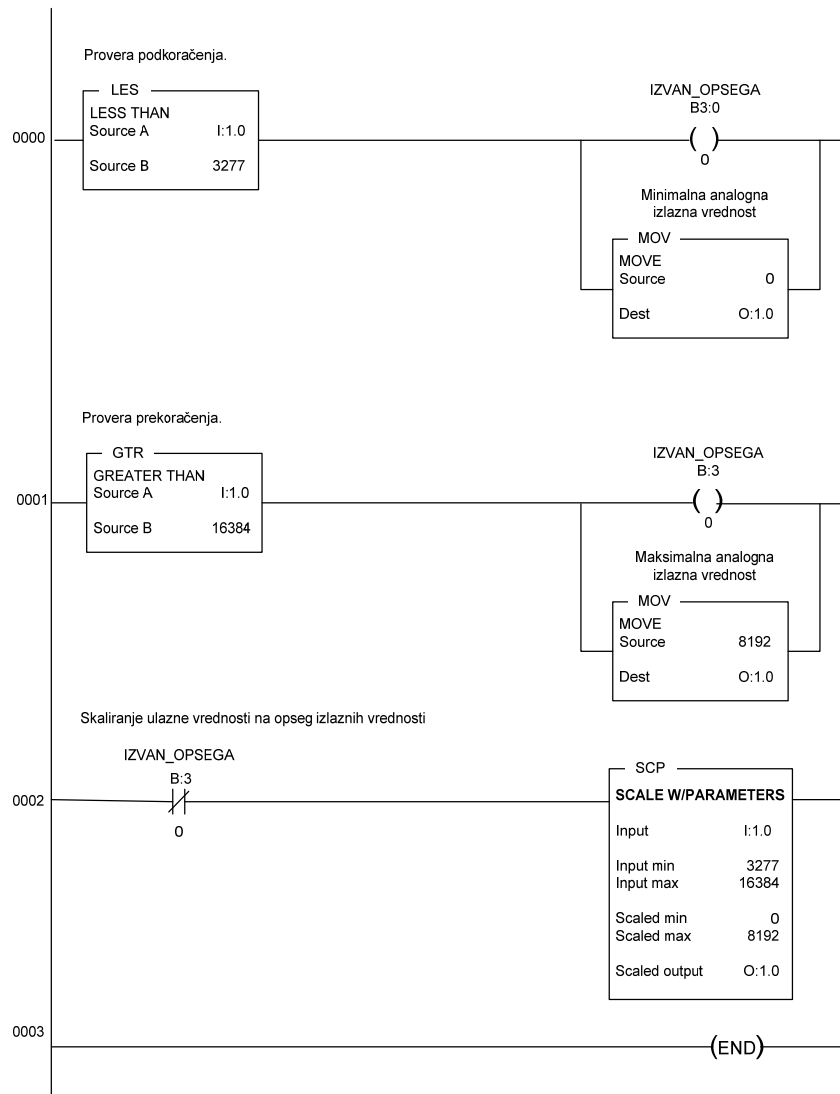
$$= (8192 - 0) / (16384 - 3277) = 8192 / 13107$$

$$offset = scaled_min - (input_min \times slope)$$

$$= 0 - (3277 \times (8192 / 13107)) = -2048, \text{ odnosno}$$

$$scaled_value = (input_value) \times (8192 / 13107) - 2048$$

Na Sl. 7-16 je prikazan odgovarajući leder program. Program, najpre ispituje da li očitana ulazna vrednost pripada dozvoljenom opsegu. Ako je ulazna vrednost izvan dozvoljenog opsega, postavlja se indikator IZVAN_OPSEGA. Ako je ulazna vrednost pripada dozvoljenom opsegu, izračunava se izlazna decimalna vrednost i prosleđuje analognom modulu. Skaliranje ulazne vrednosti se na zadati opseg izlaznih vrednosti obavlja se uz pomoć naredbe SCP.



Sl. 7-16 Leder program

Pr. 7-5 Ventilator i grejač

Zadatak: Dat je sistem koji sadrži PLC kontroler, senzor temperature za opseg 0 – 100⁰C i ventilator sa dve brzine okretanja. Senzor temperature daje struju 4-20mA, tako da struja od 4mA odgovara temperaturi od 0⁰C, a struja od 20mA temperaturi od 100⁰C. Senzor temperature je povezan na kanal 0 analognog modula (nalazi se u slotu 1) koji je konfigurisan kao strujni ulaz za opseg struja +/-20mA i decimalni opseg +/-16384. Ventilatorom se upravlja pomoću dva ON/OFF signala: U i B. Signal U uključuje (U=1) i isključuje (U=0) ventilator, dok B reguliše brzinu okretanja (B=0 - sporo, B=1 - brzo).

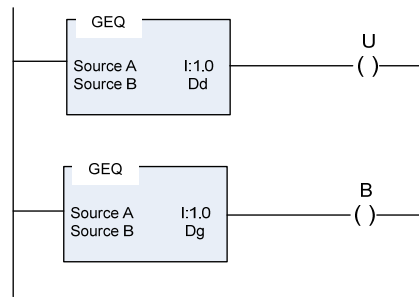
Kreirati leder program tako da je ventilator isključen za temperaturu manju od 18⁰C, okreće se manjom brzinom, za temperaturu višu od 18⁰C i manju od 25⁰C, odnosno većom za temperaturu višu od 25⁰C.

Rešenje: Upravljački signali ventilatora, U i B, zavise od temperature na sledeći način:

$$U = \begin{cases} 0 & \text{za } T < 18^{\circ}C \\ 1 & \text{za } T \geq 18^{\circ}C \end{cases}, \quad B = \begin{cases} 0 & \text{za } T < 25^{\circ}C \\ 1 & \text{za } T \geq 25^{\circ}C \end{cases}$$

PLC kontroleru je dostupna digitalizovana vrednost temperature. Neka je D_d decimalna vrednost koja odgovara temperaturi od T=18⁰C, a D_g decimalna vrednost koja odgovara temperaturi od T=25⁰C.

Odgovarajući leder dijagram prikazan je na Sl. 7-17. Za pobudu izlaza U koristi se prva naredba GEQ koja ispituje uslov $T \geq 18^{\circ}\text{C}$, a za pobudu izlaza B druga naredba GEQ koja ispituje uslov $T \geq 25^{\circ}\text{C}$.



Sl. 7-17 Leder dijagram za zadatak iz Pr. 7-5.

Da bi zadatak bio kompletan potrebno je na osnovu raspoložih podataka o karakteristikama temperaturskog senzora i analognog modula izračunati decimalne vrednosti D_d i D_g .

Zavisnost decimalne vrednosti od struje analognog modula:

Error! Objects cannot be created from editing field codes.

Zavisnost struje od temperature temperaturskog senzora:

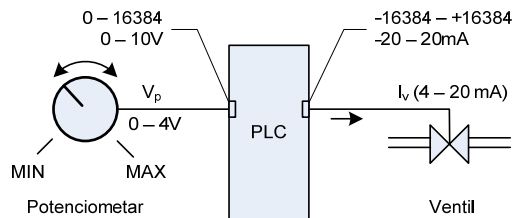
Error! Objects cannot be created from editing field codes.

Za temperaturu $T=18^{\circ}\text{C}$, senzor daje struju $I(18^{\circ}\text{C})=6.88\text{mA}$, koja se u analognom modulu digitalizuje na decimalnu vrednost $D_d=D(6.88\text{mA})=5636$.

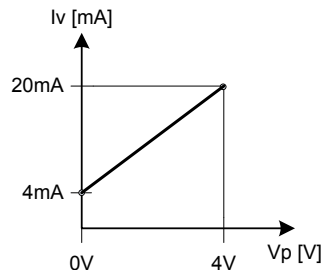
Temperaturi $T=25^{\circ}\text{C}$ odgovara struja $I(25^{\circ}\text{C})=8\text{mA}$, odnosno decimalna vrednost $D_g=6554$.

Pr. 7-6: Potenciometar i ventil

Zadatak: Sistem se sastoji od: ventila sa električnim upravljanjem, PLC kontrolera i potenciometra za regulaciju otvorenosti ventila (vidi sliku). Ventil se upravlja strujom opsega 4-20mA (za 4mA ventil je zatvoren, a za 20mA u potpunosti otvoren) i povezan je na analzni izlaz PLC kontrolera strujnog tipa sa decimalnim opsegom od -16384 do +16384 i analognim opsegom -20 - +20mA. Napon na potenciometru se menja od 0V (za poziciju MIN) do 4V (za poziciju MAX). Potenciometar je povezan na anlogni ulaz PLC kontrolera naponskog tipa sa decimalnim opsegom 0-16384 i analognim opsegom 0 - 10V. Realizovati leder dijagram tako da otvorenost ventila prati poziciju potenciometra.



Rešenje: Zadatak PLC-a je da konvertuje napon potenciometra (V_p) u odgovarajuću struju za pobudu ventila (I_v). Zavisnost struje ventila od napona potenciometra:



ili, u analitičkom obliku:

$$I_V = \frac{20mA - 4mA}{4V} V_p + 4mA = (4V_p + 4)[mA] \quad (1)$$

U ulaznom analognom modulu, napon potencijometra, V_p , se konvertuje u digitalnu veličinu (D_p), zatim se skalira na digitalnu vrednost izlazne struje (D_V), koja se u izlaznom analognom modulu konvertuje u struju I_V .

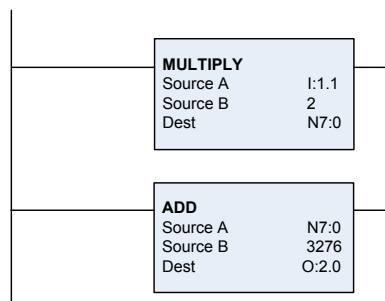
$$D_p = \frac{16384}{10V} V_p, \text{ odnosno } V_p = \frac{10V}{16384} D_p \quad (2)$$

$$I_V = \frac{20mA}{16384} D_V, \text{ odnosno } D_V = \frac{16384}{20mA} I_V \quad (3)$$

Smenom (2) u (1), a potom (1) u (3), dolazimo do funkcije skaliranja:

$$D_V = 2D_p + 3276 \quad (4)$$

Funkcija (4) se realizuje pomoću dve matematičke naredbe, kao što je prikazano na Sl. 7-18. Pretpostavka je da su adrese priključaka za potencijometar i ventil I:1.1 i O:2.0, respektivno.



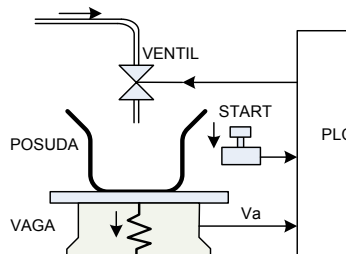
Sl. 7-18 Leder dijagram za zadatak iz Pr. 7-6.

Pr. 7-7 Punjenje posude

Zadatak: Na Sl. 9 je prikazan sistem za punjenje posuda zadatom količinom tečnosti. Sistem se sastoji iz: (a) ventila sa ON/OFF upravljanjem za dovod tečnosti u posudu; (b) elektronske vage koja za oseć težina 0-100kg daje analogni napon, V_a , iz opsega 0-10V; (c) posude koja se ručno postavlja na vagu; (d) tastera START za startovanje procesa punjenja i (e) PLC kontrolera. Izlaza elektronske vage povezan je na analogni PLC modul za opseg napona 0-10V i decimalni opseg 0-16384. Kreirati leder program prema sledećim zahtevima:

Nakon pritiska na taster START, PLC odmerava i pamti težinu posude, W_p . Ako je $W_p < 1$ kg, PLC ignoriše zahtev za punjenje (smatra da posuda nije prisutna). Ako je $W_p > 1$ kg, PLC otvara ventil i prati porast težine. PLC zatvara ventil kada težina postane veća od $W_p + W_T$, gde je W_T zadata težina tečnosti.

Usvojiti da je zadata težina tečnosti, W_T , izražena u kilogramima, dostupna u promenljivoj N7.1.



Rešenje:

V_a - analogni napon elektronske vage (0-10V)

D_a - decimalni ekvivalent analognog napona (0-16384) (dostupan preko I:1.1)

W - izmerena (trenutna) težina (0-100kg), (biće smeštena u N7.2), $W = (16384/100) * D_a$ [kg]

Decimalni ekvivalent 1kg iznosi 164 (= 16384/100)

W_T - zadata težina tečnosti (dostupna u N7.1)

W_P - težina posude (biće smeštena u N7.3)

Dijagram stanja

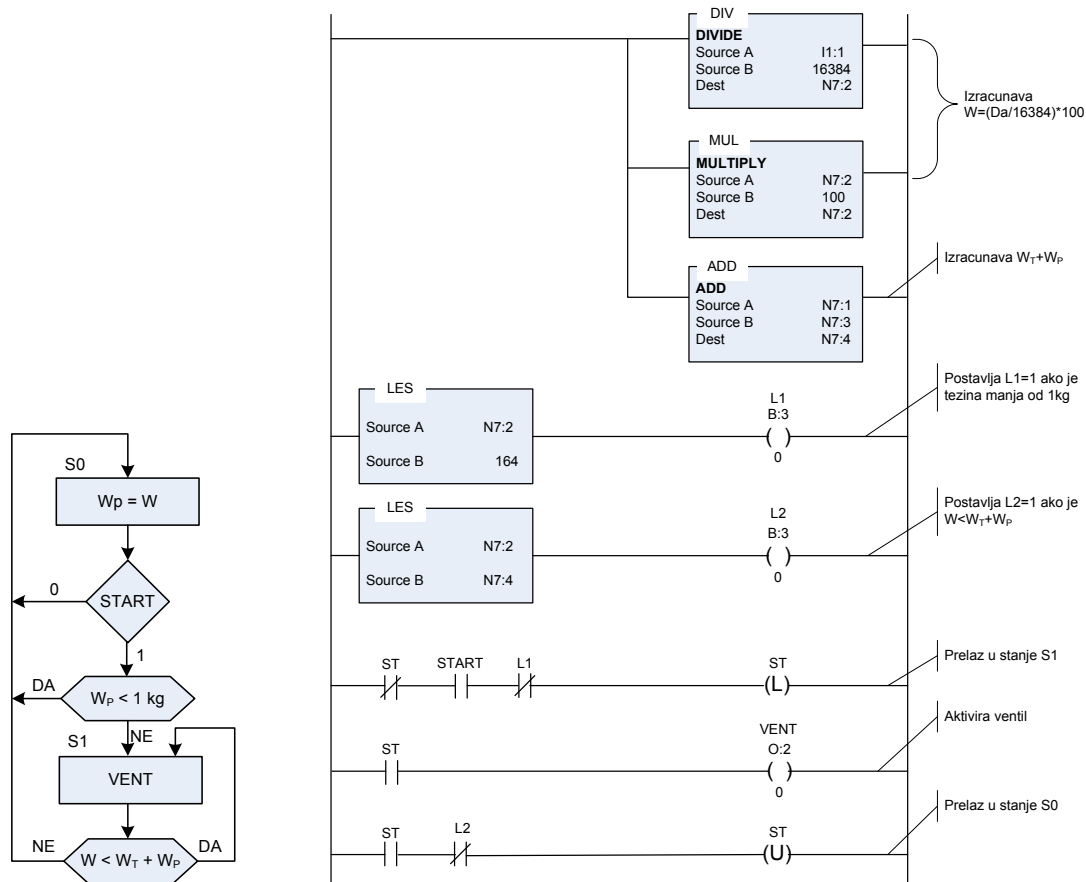
Dijagram stanja sistema prikazan je na Sl. 7-19(a). U stanju S0 sistem neprekidno očitava težinu sa vage, W_p , i nadgleda taster START. Nakon pritiska tastera (START=1), sistem proverava da li je trenutna težina veća od 1 kg i ako jeste, prelazi u stanje S1. U stanju S1, aktivan je signal VENT, koji otvara ventil. Takođe, u ovom stanju sistem prati porast težine i napušta stanje S1 (čime se ventil zatvara) onda kada težina, W , postane veća od $W_T + W_P$.

Leder dijagram

Stanje sistema se predstavlja jednim pomoćnim bitom, ST. Za ST=0, sistem je u stanju S0, a za ST=1 u stanju S1 (Sl. 7-19(b))

Leder program neprekidno, nezavisno od tekućeg stanja: (a) očitava analogni napon sa vage i konvertuje ga u težinu (W). (b) izračunava $W_T + W_P$ (c) postavlja pomoćni bit L1 ako je izmerena težina manja od 1kg i (d) postavlja pomoćni bit L2 ako je težina manja od $W_T + W_P$.

Poslednja tri ranga regulišu prelaze između stanja i aktiviranje ventila.



Sl. 7-19 Rešenje zadatka iz Pr. 7-7: (a) ASM dijagram; (b) leder dijagram.

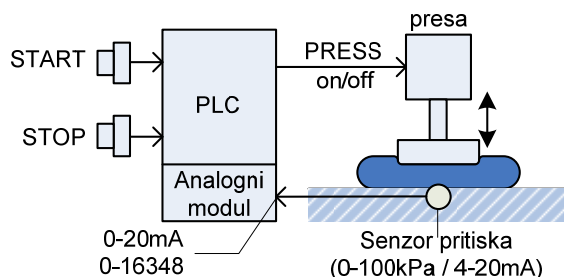
Pr. 7-8 Grejanje vode u bojleru

Realizovati leder dijagram za PLC koji održava temperaturu u bojleru u opsegu 60-70°C. Ulaz u sistem je napon temperaturskog senzora, a izlaz ON/OFF signal za upravljanje grejačem. Grejač se uključuje ako je temperatura niža od 60°C, a isključuje ako je viša od 70°C. Temperaturski senzor

konvertuje temperaturu iz opsega $0 - 100^{\circ}\text{C}$ u napon $0-5\text{V}$, i povezan je na analogni PLC modul sa naponskim ulazom za opseg napona $\pm 10\text{V}$ i decimalni opseg ± 16384 . U sistemu, takođe, postoje i dve sijalice S1 i S2. Sijalica S1 je upaljena ako je temperatura viša od 60°C . Sijalica S2 je upaljena za vreme dok je grejač uključen.

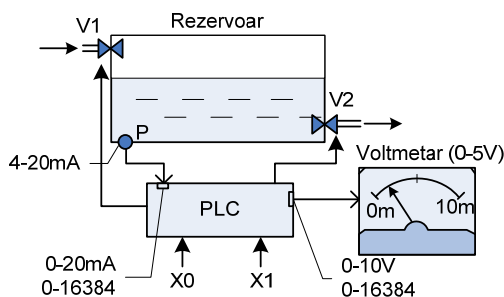
Pr. 7-9 Presovanje

Realizovati leder dijagram upravljačke jedinice prese. Presa se pušta u rad pritiskom na taster START (uslovljava $\text{PRESS}=\text{ON}$). Od tog momenta klip prese počinje da se spušta delujući na objekat koji se presuje silom koja se postepeno povećava. Pritisak prese na objekat se meri senzorom pritiska za opseg $0 - 100\text{ kPa}$ i strujnim izlazom opsega $4-20\text{ mA}$. Senzor je povezan na analogni PLC modul za strujni opseg $0-20\text{ mA}$ i decimalni opseg $0-16384$. Presovanje se završava ($\text{PRES}=\text{OFF}$) kada pritisak prese postane jednak ili veći od 50 kPa ili kada istekne maksimalno dozvoljeno vreme presovanja od 5 s . Takođe, rad prese se može prekinuti u bilo kom trenutku pritiskom na taster STOP.



Pr. 7-10 Punjenje rezervoara

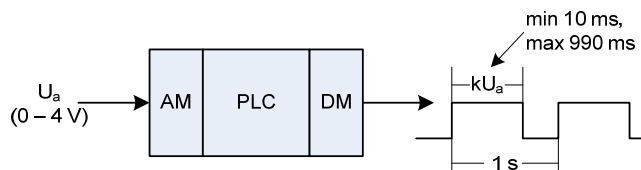
Na slici je prikazan sistem za kontrolisano dolivanje/ispuštanje vode u/iz rezervoara. Voda se sipa u rezervoar kroz ventil V1, a ispušta kroz ventil V2, oba sa ON/OFF upravljanjem. Nivo vode se meri posredno, pomoću senzora pritiska sa analognim strujnim izlazom ($4-20\text{mA}$) koji je postavljen na dno rezervoara i priključen na analogni PLC modul sa strujnim ulazom za opseg struja $0-20\text{mA}$ i decimalni opseg $0-16384$. Ako je rezervoar prazan, senzor P daje struju od 4mA ; ako je rezervoar pun (nivo vode je 10m), senzor P daje struju od 10mA . Nivo vode se prati na voltmetru sa opsegom pune skale od 5V čija je skala baždarena u metrima ($0-10\text{m}$). Voltmetar je priključen na analogni izlazni PLC modul sa naponskim izlazom za opseg napona $0-10\text{V}$ i decimalni opseg $0-16384$. Taster X0 služi za dolivanje, a X1 za ispuštanje vode. Pri tome, treba onemogućiti dolivanje vode ako je rezervoar pun. Realizovati odgovarajući leder dijagram.



Pr. 7-11 Impulsno-širinski modulator

Razviti leder dijagram koji će u PLC sistemu sa Sl. 1 realizovati funkciju impulsno-širinskog modulatora. PLC sistem je opremljen: (a) ulaznim analognim modulom (AM) sa naponskim ulazima za opseg napona $\pm 10\text{V}$ i decimalni opseg ± 16384 i (b) izlaznim digitalnim modulom (DM). Na ulaz analognog modula dovodi se analogni napon U_a iz opsega $0-4\text{ V}$, dok se na izlazu digitalnog modula generiše signal pravouganog talasnog oblika fiksne periode $T = 1\text{ s}$ (1000 ms), ali promenljivog trajanja impulsa. Trajanje impulsa je srazmerno ulaznom

analognom naponu, tako da je pri $U_a = 0 \text{ V}$ trajanje impulsa najkraće i iznosi 10 ms, a pri $U_a = 4 \text{ V}$ najduže i iznosi 990 ms.

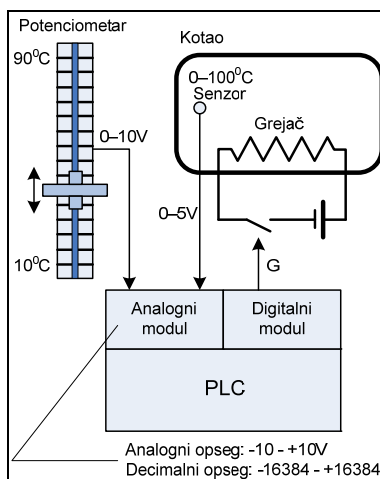


Sl. 1.

Pr. 7-12 Grejanje tečnosti u kotlu

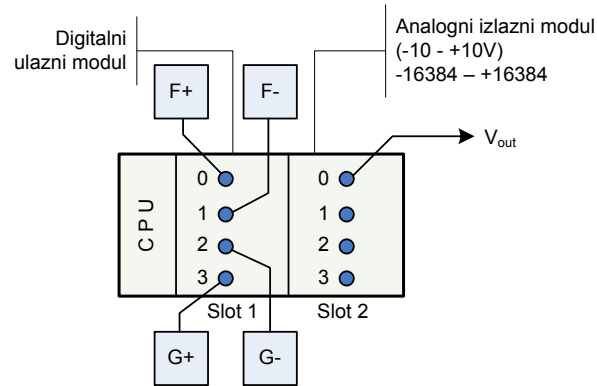
Realizovati leder dijagram za PLC sa slike čiji je zadatak održavanje zadate temperature tečnosti u kotlu. Temperatura tečnosti se meri pomoću temperaturskog senzora, koji temperaturu iz opsega $0 - 100^{\circ}\text{C}$ konvertuje u napon $0-5\text{V}$. Tečnost se zagreva pomoću grejača sa ON/OFF upravljanjem. Ciljna temperatura se postavlja pomoću kliznog, linearnog potencijometra koji je baždaren u stepenima celzijusa sa skalom od $10 - 90^{\circ}\text{C}$. Kada je klizač u položaju 10°C , na izlazu potencijometra je napon 0V , a kada je u položaju 90°C , napon na izlazu potencijometra je 10V . Temperaturski senzor i potencijometar su povezani na analogni PLC modul sa naponskim ulazima za opseg napona $\pm 10\text{V}$ i decimalni opseg ± 16384 .

Neka je T_Z zadata, a T_M izmerena temperatura tečnosti. Grejač se uključuje kada razlika između zadate i izmerene temperature postane veća od 1°C ($T_Z - T_M > 1^{\circ}\text{C}$), a isključuje kada razlika između izmerene i zadate temperature postane veća od 1°C ($T_M - T_Z > 1^{\circ}\text{C}$).



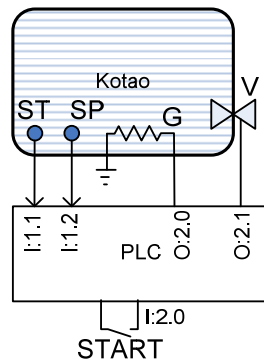
Pr. 7-13 Podešavanje izlaznog napona

PLC sistem čine: digitalni ulazni modul i analogni izlazni modul (Sl. 2). Karakteristike analognog izlaznog modula su: naponski izlaz sa opsegom $-10 - +10\text{V}$ i decimalni opseg: $-16384 - +16384$. Napon sa izlaza 0 analognog modula, V_{out} , se koristi za upravljanje nekim uređajem, kao što je npr. ventil. Zahtev je da radni opseg ovog napona bude u granicama $0 - 5\text{V}$ sa mogućnošću manualnog podešavanje u koracima od po 0.1V . Za manualno podešavanje napona V_{out} koriste se dva para tastera povezanih na digitalni ulazni modul. Tasteri G+ i G- se koriste za grubo, a tasteri F+ i F- za fino podešavanje napona V_{out} . Svaki pritisak na taster G+ (G-) uvećava (umanjuje) napon V_{out} za 1V , dok svaki pritisak na taster F+ (F-) uvećava (umanjuje) napon V_{out} za 0.1V . Realizovati odgovarajući leder dijagram.



Pr. 7-14 Kotao

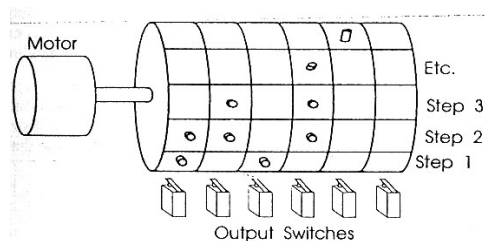
Na slici 10 prikazani su elementi sistema za upravljanje zagrevanjem tečnosti u kotlu. Sistem se sastoji od grejača, G, analognog senzora temperature, ST, analognog senzora pritiska, SP, sigurnosnog ventila sa ON/OFF upravljanjem, V i start tastera, START. Senzor ST konvertuje temperaturu iz opsega 0 – 300⁰C u struju 4-20mA, dok senzor SP konvertuje pritisak iz opsega 0 – 200psi u struju 4-20mA. Senzori ST i SP su povezani na analogni PLC modul sa strujnim ulazima za opseg ±20mA i decimalni opseg ±16384. Nacrtati leder dijagram po sledećim zahtevima: 1) Nakon pritiska na taster START, uključuje se grejač G. 2) Kada temperatura dostigne vrednost TG ⁰C, grejač se isključuje; 3) Ako u toku zagrevanja pritisak u kotlu postane veći od PG psi, grejač se isključuje i ventil V otvara. 4) Ventil se ponovo zatvara kada pritisak padne ispod vrednosti PD psi. Usvojiti da su vrednosti TG, PG i PD smeštene u registrima N7:0, N7:1 i N7:2, respektivno.



8 Sekvencijalno upravljanje

Jedan od čestih zadataka pri upravljanju procesima je *sekvencijalno upravljanje*. Ovim upravljanjem se izvršnim uređajima na procesu zadaje niz naredbi binarnog tipa (uključi/isključi, napred/nazad, kreni/stani i sl.) koje se smenjuju u vremenu, pri čemu svaka aktivnost traje određeni, unapred definisani interval vremena, ili dok se ne detektuje nastanak nekog događaja. Zamislimo, na primer, proizvodnu liniju za punjenje boca nekom tečnošću. Proces se odvija tako što se boce najpre stavljaju na liniju, a zatim peru, pune, zatvaraju, proveravaju i konačno pakuju – dakle, niz (sekvencija) uvek istih operacija. Mašina za pranje rublja je dobar primer sekvencijalnog upravljanja. Bitno obeležje sekvencijalnog načina upravljanja je da je sekvencija unapred potpuno određena i da se niz aktivnosti može definisati kao sukcesivan niz binarno kodiranih reči, kod kojih se svaki bit odnosi na pojedini izvršni organ, koji je vezan za kontroler preko odgovarajućeg digitalnog izlaznog modula. Kako se proces odvija, tako se na izlazni modul prenosi reč po reč iz upravljačke sekvence.

U prošlosti, za realizaciju sekvencijalnog upravljanja korišćen je doboš-programator. Doboš je oblika cilindra sa udubljenim rupama, koje su ravnomerno raspoređene duž većeg broja paralelnih, kružnih staza. U svaku rupu se može postaviti trn, a raspored trnova duž jedne staze odgovara binarnoj sekvenci za pobudu jednog izvršnog organa (Sl. 8-1). Neposredno uz doboš postavljen je niz prekidača, pri čemu svaki prekidač upravlja jednim izvršnim uređajem. Kako se doboš okreće, tako trnovi zatvaraju prekidače na čijim se izlazima generiše željena sekvencija binarno kodiranih upavljačkih reči. Doboš okreće motor, čija brzina rotiranja može da se reguliše. Doboš-programator se lako programira. Korisnik najpre formira tabelu koja za svaki korak (fazu procesa), pokazuje koji izvršni uređaji su aktivni u tom koraku (Sl. 8-2). Međutim, doboš-programator ima i niz nedostataka. Na primer, iako brzina rotiranja doboša, a time i brzina odvijanja procesa može da se reguliše uz pomoć motora, ne postoji mogućnost programiranja trajanja pojedinih koraka – svi koraci imaju isto trajanje. Takođe, doboš-programator nije u stanju da ispituje ulazne informacije o tekućem stanju procesa i da na osnovu tih informacija, eventualno promeni radnu sekvenciju.



Sl. 8-1 Doboš-programer. (*Output Switches* – izlazni prekidači; *Step 3-1* – koraci 3-1, svaki red trnova odgovara jednom koraku, tj. fazi)

Step	Input Pump	Heater	Add Cleaner	Sprayer	Output Pump	Blower
1	on		on			
2	on	on		on		
3		on		on		
4					on	
5						on

Sl. 8-2 Primer izlazne sekvence doboš-programatora sa Sl. 8-1. (*Input Pump* – ulazna pumpa; *Heater* – grejač; *Add Cleaner* – dodaj sredstvo za čišćenje; *Sprayer* – prskalica; *Output Pump* – izlazna pumpa)

Mnoga ograničenja karakteristična za doboš-programator mogu se prevazići ako se za sekvencijalno upravljanje koristi PLC kontroler. Za razliku od doboš-programatora, kod koga svi koraci imaju isto trajanje, kod PLC realizacije, s obzirom na daleko veću fleksibilnost programiranja, uobičajeno se koristi pristup kod koga je prelazak sa jedne na drugu aktivnost uslovljen stanjem u pojedinim delovima procesa. To znači da je neophodno da se, pod određenim uslovima, očitavaju stanja indikatora na procesu i poredbe sa unapred definisanim stanjima. U zavisnosti od rezultata poređenja, odlučuje se da li je došlo vreme za sledeću aktivnost. Kada je odgovor potvrđan, onda je izvesno da

proces ulazi u sledeću fazu, te da se nadalje stanje mora porediti sa drugim nizom vrednosti koji ukazuje na završetak sledeće faze. Dakle, moguće je da se svi parametri koji učestvuju u poređenju, urede u jedan niz binarno kodiranih reči, i da se stanje procesa, koje se učitava preko digitalnih ulaznih modula poredi sa odgovarajućom reči iz niza.

Nema nikakve sumnje da bi se opisane operacije mogle izvesti kombinovanjem naredbi za unošenje i iznošenje digitalnih podataka i naredbi za poređenje. Međutim, pošto je potreba za ovim operacijama izuzetno izražena, predviđene su posebne naredbe kojima se one u celosti mogu realizovati.

8.1 Naredbe za sekvencijalni rad sa datotekama

U okviru ovih naredbi bar jedan od operandi je datoteka u kojoj se nalazi niz podataka. Pri tome se dozvoljava rad samo sa onim datotekama čiji elementi su dužine jedne reči. Adrese pojedinih podataka određuju se pomoću *bazne adrese* koja se definiše u naredbi i *pointera* koji predstavlja upravljački parametar, čija se vrednost menja u toku ponovljenih izvršavanja naredbe. Pri tome se adresa operanda dobija kao zbir *bazne adrese* i vrednosti *pointera*. U naredbama se definiše početna vrednost pointera kao i ukupna dužina niza.

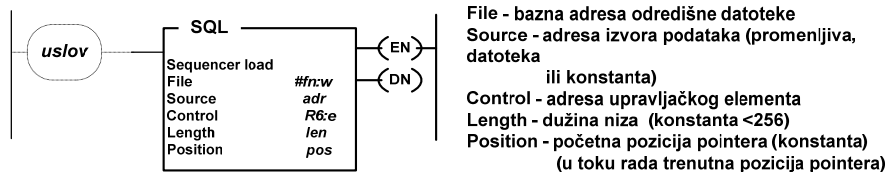
Ako neka aplikacija zahteva da se sekvencijalna obrada izvrši nad podacima koji su duži od 16 bitova, onda se ti podaci moraju podeliti na više datoteka. Tada se u svakom rangu na izlazu kao naredbe *akcije* mogu paralelno staviti više istih naredbi kojima se adresiraju sve definisane datoteke.

Svakoј datoteci koja se specificira u okviru neke sekvencijalne naredbe pridružuje se po jedan element upravljačke datoteke *R*. U okviru ovog elementa pamte se indikatorski bitovi, kao i vrednost pointera i dužina same datoteke. O formatu jednog elementa ove datoteke biće kasnije više reči.

Samo kada se *uslov* u rangu menja sa *neistinit* na *istinit* menja se vrednost pointera i on ukazuje na drugi podatak. Međutim, ako *uslov* posle toga ostane i dalje *istinit*, pointer ne menja vrednost već se naredba izvršava sa podatkom koji je uzet pri poslednjoj promeni pointera.

• SQL – Sequencer Load (sekvencijalno punjenje datoteke)

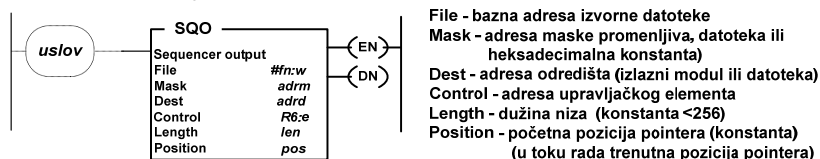
SQL naredba, grafički simbol i položaj u rangu



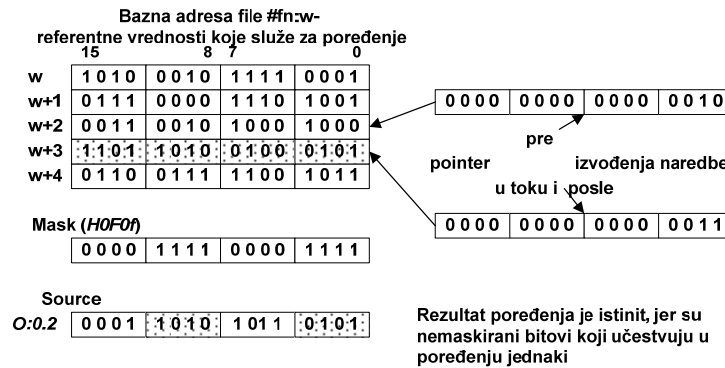
Svaki put kada se *uslov* menja sa *neistinit* na *istinit*, ova naredba se izvršava tako što se vrednost pointera poveća za 1 i podatak koji je određen kao *source* prenese u datoteku *file* na onu adresu na koju pokazuje pointer. Na taj način se pri svakom sledećem izvršavanju naredbe menja sadržaj sledeće reči u nizu. Ukoliko se kao *source* adresa navede konstanta onda se ceo niz postavlja na istu vrednost. Ako je *source* adresa promenljiva (*fn:s*), onda svaka reč niza dobija vrednost koju promenljiva ima u trenutku izvođenja naredbe. Međutim, ako se kao *source* adresa navede datoteka (*#fn:s*), onda se ta adresa uzima kao *bazna adresa izvorne datoteke*, što znači da se pri izvođenju naredbe podatak uzima sa one adrese na koju u izvornoj datoteci pokazuje pointer. Pri tome se podrazumeva da obe datoteke imaju istu dužinu, definisanu kao *length*. Pri sledećim sken ciklusima, za sve vreme za koje *uslov* ostaje *istinit*, vrednost pointera se ne menja, već se isti, prethodno određeni, podatak prenosi u promenljivu označenu sa *dest*.

• SQO – Sequencer output (sekvencijalno upravljanje)

SQO naredba, grafički simbol i položaj u rangu



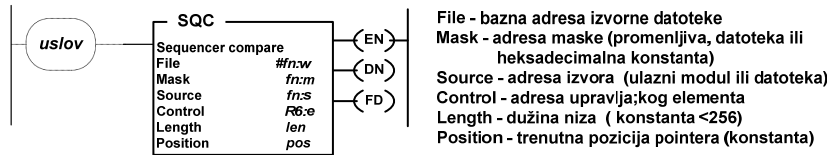
Svaki put kada se uslov menja sa *neistinit* na *istinit*, ova naredba se izvršava tako što se vrednost pointera (*position*) poveća za 1 i uzme ona reč iz datoteke *file* (*#fn:w*) na koju pokazuje pointer. Ta reč se filtrira kroz masku *mask* i rezultat filtracije se prenosi u promenljivu označenu sa *dest*. Ako je kao *dest* navedena datoteka *#fn:d* onda će se rezultat upisati u onu reč te datoteke na koju pokazuje pointer (Sl. 8-3). Isto tako, ako je kao *mask* navedena datoteka *#fn:m* onda i maska prestaje da bude fiksna, već se svaki put kao maska uzima ona reč iz datoteke na koju pokazuje pointer. Potrebno je zapaziti da se u reči koja označena sa *dest* menjaju samo oni bitovi koji su nemaskirani (odgovarajući bitovi maske su postavljeni na 1). Pri sledećim sken ciklusima, za sve vreme za koje *uslov* ostaje *istinit*, vrednost pointera se ne menja, već se isti, prethodno određeni, podatak prenosi u promenljivu označenu sa *dest*.



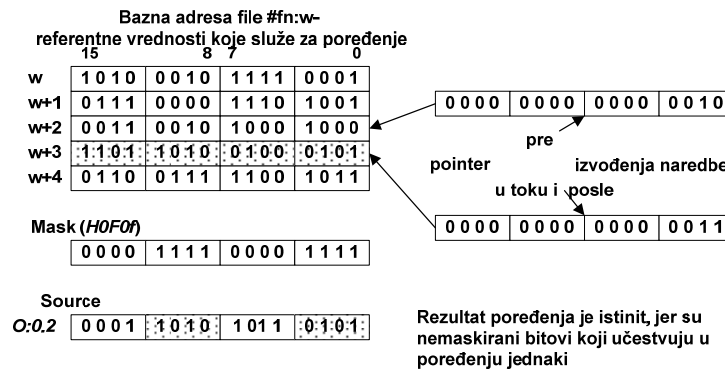
Sl. 8-3 Ilustracija izvršavanja SQO naredbe.

• **SQC – Sequencer compare (sekvencijalno poređenje)**

SQC naredba, grafički simbol i položaj u rangu



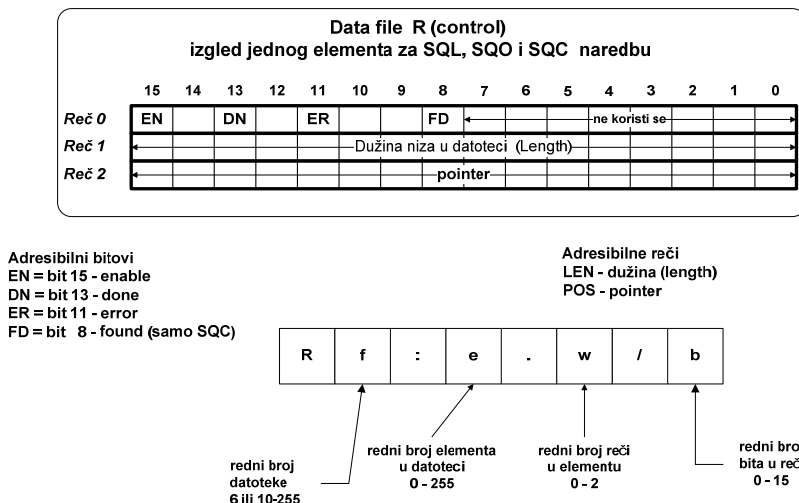
Svaki put kada se uslov menja sa *neistinit* na *istinit*, SQC naredba se izvršava tako što se vrednost pointera (*position*) poveća za 1 i uzme ona reč iz datoteke *#fn:w* na koju pokazuje pointer. Ta reč se poredi sa filtriranim podatkom koji sadrži promenljiva označena kao *source* i rezultat poređenja se upisuje u odgovarajući indikatorski bit. Filtracija podatka vrši se pomoću maske *mask* i to tako da u poređenju učestvuju samo oni bitovi kojima u maski odgovara vrednost bita 1 (nemaskirani bitovi). Ako je kao *source* navedena datoteka *#fn:s* onda će se podaci koji učestvuju u poređenju uzimati iz one reči te datoteke na koju pokazuje pointer (Sl. 8-4). Isto tako, ako je kao *mask* navedena datoteka *#fn:m* onda i maska prestaje da bude fiksna, već se svaki put kao maska uzima ona reč iz datoteke na koju pokazuje pointer. Pri sledećim sken ciklusima, za svo vreme za koje *uslov* ostaje *istinit*, vrednost pointera se ne menja, već se isti, prethodno određeni, podatak (*source*) uzima kao podatak za poređenje.



Sl. 8-4 Ilustracija izvršavanja SQC naredbe.

Datoteka R – Control

Naredbama za sekvenciranje pridružuju se indikatorski bitovi i upravljački parametri. Ove informacije se smeštaju u *upravljačku datoteku* tipa R. Pri tome se može koristiti sistemska upravljačka datoteka broj 6, ili korisnička datoteka (brojevi od 9 do 255). Jedan elementat ove datoteke, koji se odnosi na SQO i SQC naredbu ima izgled kao na Sl. 8-5.



Sl. 8-5 Element upravljačke datoteke za SQO i SQC naredbu.

Bitovi stanja u elementu datoteke R menjaju se na sledeći način:

EN – Enable bit se postavlja na 1 kada *uslov* prelazi sa *neistinit* na *istinit*. Postavljanje ovog bita prouzrokuje da se izvrši naredba i vrednost pointera poveća za 1. Pri svakom sledećem prolazu kroz ovaj rang, sve dok je uslov *istinit*, EN bit zadržava vrednost 1, ali se vrednost pointera ne menja, već se naredba izvršava sa istom vrednošću pointera. Kada *uslov* postane *neistinit*, EN bit se resetuje na 0.

DN – Done bit se postavlja na 1 kada vrednost pointera, posle niza izvođenja SQL, SQO ili SQC naredbe, dođe do kraja niza u zadanoj datoteci. Ovaj bit će biti resetovan na 0 tek u onom sken ciklusu u kome *uslov*, pošto je prethodno postao *neistinit*, ponovo postaje *istinit* (kada se EN bit ponovo postavi na 1).

ER – Error bit se postavlja na 1 kada se u programu detektuje negativna vrednost pointera, ili negativna ili nulta vrednost dužine niza. Ako se ovaj bit ne resetuje pre kraja sken ciklusa nastaje *značajna greška*.

FD – Found bit se postavlja na 1 ako je rezultat poređenja u SQC naredbi *istinit*. Drugim rečima ovaj bit ukazuje na to da su nemaskirani bitovi podataka jednaki odgovarajućim bitovima u datoteci referentnih vrednosti.

Length i position

Promenljive *length* se pamti u prvoj reči datoteke R i predstavlja broj reči koji se nalazi u nizu u jednoj sekvencijalnoj datoteci. Maksimalna vrednost dužine je 255. Pri definisanju dužine, potrebno je voditi računa o činjenici da navedena adresa reči *w* u datoteci *#fn:w* zapravo predstavlja nultu, početnu poziciju. To znači da se za datu dužinu *l* u datoteci koristi zapravo *l+1* reč. Ovo se naravno odnosi i na *mask*, *source* i *dest* ukoliko su u naredbi specificirane kao datoteke.

Vrednost pointera, označena kao *position*, pamti se u drugoj reči datoteke R. Početna vrednost pointera se definiše pri specifikaciji naredbe. Vrednost pointera se kreće od 1 do *l* i ukazuje na reči u datoteci od *fn:(w+1)* do *fn:(w+l+1)*. Kada pointer stigne do poslednje reči u datoteci, postavlja se DN-bit na 1 i pri tome se u prvom sledećem sken ciklusu u kome *uslov* ima prelaz sa *neistinit* na *istinit* (isiti ciklus u kome se resetuje DN bit) vrednost pointera automatski vraća na 1. Pri definiciji početne vrednosti pointera potrebno je obratiti pažnju na činjenicu da se ona poveća za 1 pre prvog izvođenja naredbe.

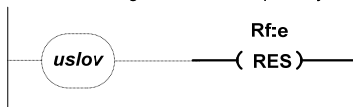
Ako se kao početna vrednost pointera definiše 0, onda će pri izvođenju SQC naredbe obrada početi od reči u datoteci čija je adresa $fn:(w+1)$. Međutim kod SQO naredbe način izvođenja operacije zavisi od istinitosti *uslova* u prvom sken ciklusu. Ako je *uslov istinit* naredba se izvršava počev od nulte reči, čija je adresa $fn:(w+0)$. Međutim, ako je uslov *neistinit*, izvršavanje naredbe se odlaže sve dok uslov ne postane *istinit* i tada se uzima prva reč, čija je adresa.

Konačno, važno je da se istakne da se prilikom eventualne programske promene dužine i pozicije mora voditi računa da se ne prekorači veličina definisanog niza u datoteci.

Resetovanje parametara

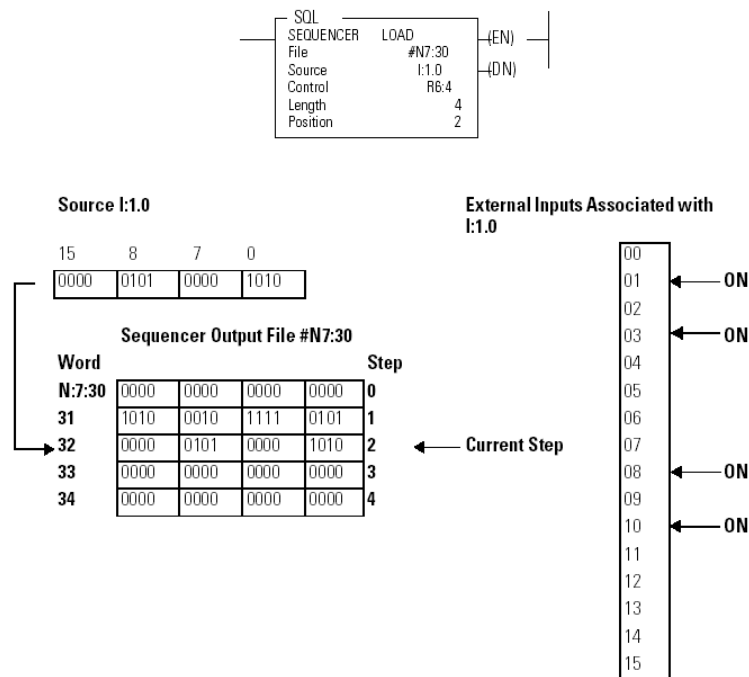
Ukoliko se iz nekog razloga želi prekinuti sekvencijalno upravljanje ili poređenje, to se može ostvariti pomoću *RES* naredbe u kojoj se navodi adresa nulte reči elementa datoteke R koji je vezan za naredbu čiji se rad želi resetovati *Rn:e*. *RES* naredbom se vrednosti svih indikatorskih bitova, izuzev *FD*-bita, postavljaju na 0. Istovremeno se i vrednost pointera postavlja na 0 (ova vrednost će se povećati na 1 pre prvog sledećeg izvođenja naredbe).

RES naredba, grafički simbol i položaj u rangu



8.1.1 Korišćenje SQL naredbe

Na Sl. 8-6 je prikazan primer SQL naredbe. Parametri SQL naredbe su tako podešeni da je ulazna reč I:1.0 izvor podataka, a da se pročitani podaci smeštaju u odredišnu datoteku #N7:30 veličine 4 reči. Kada se uslov ranga promeni sa *neistina* na *istina*, pointer SQL naredbe se uvećava za jedan i podatak očitani sa ulaza I:1.0 se upisuje na poziciju u odredišnom fajlu na koju ukazuje pointer. SQL naredba nastavlja da prenosi podatke sa ulaza na istu poziciju odredišne datoteke sve dok je uslov ranga istinit. Nakon što je korak 4 završen, setuje se bit DN. Sledeća promena uslova sa *neistina* na *istina*, vraća pointer na poziciju 1 (element N7:31). Da je umesto ulazne reči I:1.0, kao izvor navedena interna datoteka, npr. #N7:40, tada bi obe datoteka odredišna i izvršna bile iste dužine (4), a SQL naredba bi, praktično, kopirala sadržaj izvorne datoteke u odredišnu.

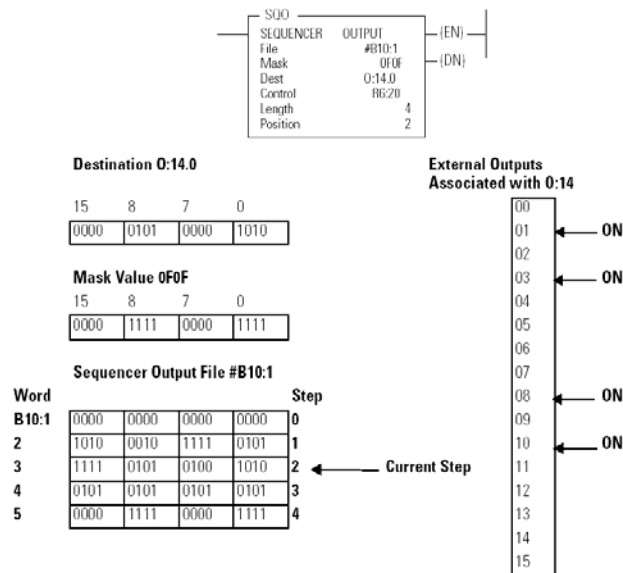


Sl. 8-6 Primer SQL naredbe.

8.1.2 Korišćenje SQO naredbe

SQO naredba čita reči specificiranog fajla čiji su bitovi namenjeni upravljanju izvršnih organa. Kada uslov ranga koji sadrži SQO naredbu postane istinit, naredba čita sledeću reč iz fajla. Pročitana reč prolazi kroz masku i prenosi se na odredišnu adresu. Samo oni bitovi pročitane reči kojima u maski odgovara bit 1 se prenose. Bit DONE postaje 1 nakon što je iz fajla pročitana poslednja reč. Sledeći promena uslova ranga sa *neistina* na *istina* usloviće da SQO naredba krene iz početka, sa pointerom koji je resetovan na početnu vrednost 1.

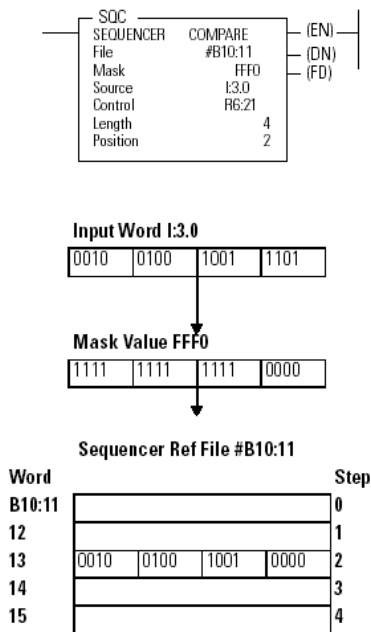
Na Sl. 8-7 je ilustrovan rad SQO naredbe. SQO naredba sa slike definiše sekvencu od 5 reči smeštenu u datoteci B10, počev od reči B10:1. Početna vrednost pointera je 2, što znači da će kao prva reč iz izvorišnog fajla biti pročitana reč B10:3. Odredišna adresa je prva reč izlazne datoteke O:14. (Dugim rečima, odredište pročitanih reči iz ulaznog fajla je izlazni digitalni modul sa adresom 14.) Naredba SQO definiše masku čija je heksadecimalna vrednost 0F0F. To znači da se na odredište ne prenose svi bitovi pročitanih reči, već samo osam bita, naznačenih jedinicama u maski (to su bitovi 0-3 i bitovi 8-11). Naravno, sadržaj izvorišnog fajla mora biti unapred pripremljen, prema zahtevima aplikacije, i upisan u odgovarajuće reči naznačene datoteke. Očigledno, SQO naredba predstavlja softverski ekvivalent doboš-programatora.



Sl. 8-7 Primer SQO naredbe

8.1.3 Korišćenje SQC naredbe

SQC naredba radi tako što poredi sve ne-maskirane bitove iz izvorišne reči sa odgovarajućim bitovima referencirane reči. Ako na svim bitskim pozicijama postoji slaganje, naredba SQC postavlja bit FD kontrolne reči na 1. U suprotnom, ako postoji neslaganje barem na jednoj poziciji, bit FD se postavlja na 0. Na Sl. 8-8 je prikazan primer SQC naredbe. Ova SQC naredba definiše referentni fajl dužine 5 reči šmešten u datoteci B10, počev od reči 11 ove datoteke. Izvorišna adresa (I:3.0) ukazuje na prvu reč ulazne datoteke I3 (odgovara ulaznom digitalnom modulu sa adresom 3). Tekuća pozicija pointera je 2, što znači da se vrednost pročitana sa ulaza I:3.0 poredi sa rečju B10:13. Ne poredi se celokupne reči već samo viših 12 bita, što je definisano maskom 0FFF. Pošto su u konkretnom primeru svi odgovarajući ne-maskirani bitovi identični, SQC naredba setuje FD bit, tj. bit R6:21/FD.



Sl. 8-8 Primer SQC naredbe.

8.2 Primeri

Pr. 8-1 Primena SQO naredbe

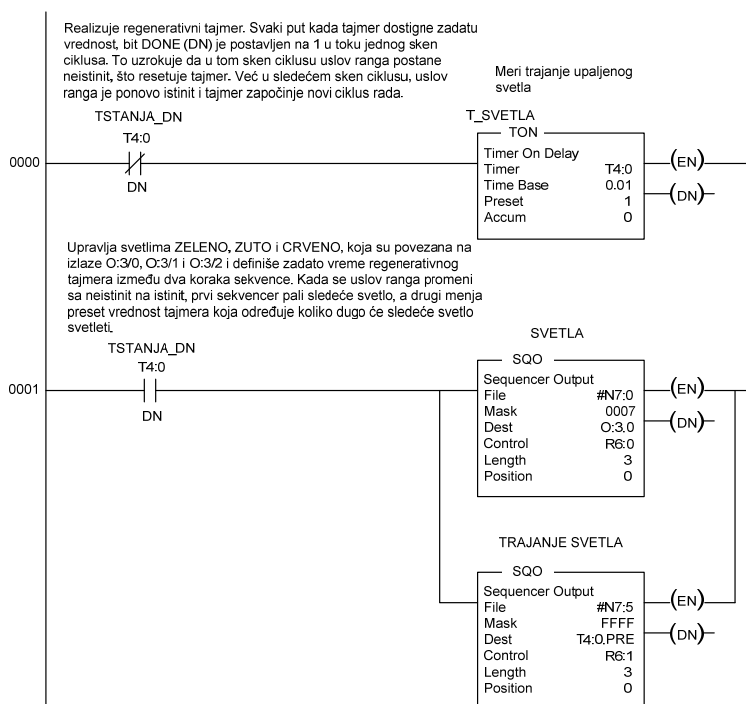
Zadatak: Realizovati leder program za upravljanje semaforom. Semafor ima tri sijalice: ZELENO, ZUTO i CRVENO, koje naizmenično svetle u trajanju od:

ZELENO	60s
ZUTO	15s
CRVENO	30s

Rešenje: Usvajamo da su semaforska svetla povezana na izlazni modul na sledeći način:

ZELENO	O:3/0
ZUTO	O:3/1
CRVENO	O:3/2

Leder program je prikazan na Sl. 8-9. Leder program koristi TON naredbu za merenje vremena i SQO naredba za realizaciju sekvencijalnog upravljanja. Uvek kada tajmer dostigne zadatu vrednost, izvršavaju se dve SQO naredbe. Rang 0 realizuje regenerativni tajmer. Po dostizanju zadate vrednosti, bit DONE (DN) tajmera postaje 1. S obzirom da se bit DN koristi kao uslov u rang 1, dostizanje zadate vrednosti tajmera inicira izvršenje dve SQO instrukcije. Prva SQO naredba čita sledeću reč iz svog izvorišnog fajla i prenosi je na izlazni modul koji pobuđuje sijalice. Sadržaj izvornog fajla prve SQO naredbe prikazan je na Sl. 8-10(a). Druga SQO naredba čita sledeću reč iz svog izvorišnog fajla i upisuje je u PRE registar tajmera, čime se definiše vreme u toku koga će upravo upaljeno svetlo ostati upaljeno. Sadržaj izvornog fajla druge SQO naredbe prikazan je na Sl. 8-10(b).



Sl. 8-9 Leder dijagram kontrolera jednostavnog semafora.

Adresa	Podatak (binarno)			
	15-3	2	1	0
N7:1	0...0	0	0	1
N7:2	0...0	0	1	0
N7:3	0...0	0	0	1

(a)

Adresa	Podatak (decimalno)
N7:6	6000
N7:7	1500
N7:8	3000

(b)

Sl. 8-10 (a) sadržaj izvorišne datoteke sekvencera SVETLA; (b) sadržaj izvorne datoteke sekvencera TRAJANJE SVETLA.

Pr. 8-2 Realizacija sekvence

Zadatak: Dat je sistem koji sadrži jedan jednosmerni solenoid (A) i dva dvosmerna (B i C). Potrebno je realizovati sledeću sekvencu pomeranja klipova: A+ B+ C+ B- A- C- . Pri tome se predpostavlja da su u početnom trenutku svi klipovi uvučeni. Granični prekidači koji indiciraju uvučenost klipa A i B su *normalno zatvoreni*, dok su svi ostali granični prekidači *normalno otvoreni*. Sistem se pušta u rad pomoću pritiska na taster i prestaje sa radom kada se jedanput izvrši zahtevana sekvenca

Rešenje: Zadatak će biti rešen korišćenjem naredbi SQO i SQC. Sistem ima 7 ulaza, SWA-, SWA+, SWB-, SWB+, SWC-, SWC+ i START i 5 izlaza, MOVE_A+, MOVE_B+, MOVE_B-, MOVE_C+ i MOVE_C-. U početnom stanju (korak 0), klipovi sva tri solenoida su uvučeni, a da bi se otpočelo sa radom čeka se pritisak na prekidač START. Kada se pritisne prekidač START, sistem prelazi na korak 1. U ovom koraku, aktivan je izlaz MOVE_A+ (klip solenoida A se izvlači) i čeka se da klip dođe u krajnji izvučeni položaj (SWA+ = 1). U koraku 2, koji počinje kada klip solenoida A dostigne krajnji izvučeni položaj, obavlja se izvlačenje klipa solenoida B: MOVE_B+ = 1 i čeka se na SWB+ = 1. Pri tome, zadržava se pobuda solenoida A kako bi klip solenoida A ostao u krajnjem izvučenom položaju (tj. MOVE_A+ = 1). Dakle, sistem prolazi kroz sekvencu koraka pri čemu su za svaki korak definisani aktivni izlazi kao i stanje ulaza na koje se čeka kako bi sistem prešao na sledeći korak. Za postavljanje izlaza biće korišćena naredba SQO, a za ispitivanje uslova za prelaz na sledeći korak naredba SQC. Izvorne datoteke obe naredbe imaju istu dužinu, jednaku broju koraka sekvence uvećanom za 1. Ove dve naredbe rade u paru, sinhronizovano, tako da pointeri u izvornim datotekama obe naredbe ukazuju na iste pozicije. Par reči sa istih pozicija dve izvorne datoteke odgovara jednom koraku. Reč iz izvorne datoteke naredbe SQO definiše stanje izlaza, a reč iz izvorne datoteke naredbe SQC na uslov prelaska na novi korak. Najjednostavni način za sinhronizaciju dve naredbe je da one

dele isti kontrolni registar. Kao što se vidi sa Sl. 8-5, kontrolni registar naredbe sekvenciranja pored indikatorskih bitova sadrži i tekuću poziciju pointera na izvornu datoteku. Ako dve naredbe dele isti kontrolni registar, onda će one deliti zajednički pointer, tj. uvećanje pointera u okviru jedne naredbe značiti i uvećanje pointera one druge naredbe. Pri tome, leder dijagram će biti tako konstruisan da je naredba SQC uvek aktivna, a da se naredba SQO aktivira samo kada je indikator FD postavljen na 1, tj. kada očitani ulaz postane jednak očekivanoj vrednosti. S obzirom da naredba SQO/SQC obavlja inkrementiranje pointera uvek kada se uslov ranga promeni sa *neistinit* na *istinit*, inkrementiranje pointera obavljaće samo naredba SQO. Leder program je prikazan na Sl. 8-11. Rang 0 resetuje sekvencer u prvom sken ciklusu. Rang 1 sadrži SQC naredbu koja je uvek aktivna (izvršava se u svakom sken ciklusu). Rang 2 sadrži SQO naredbu koja se izvršava samo u onim sken ciklusima kada je FD=1, tj. kada treba preći na sledeći korak. Sadržaj izvornih datoteka naredbi SQC i SQO definisan je tabelama T. 9, T. 10 i T. 11. Datoteka maski naredbe SQC definiše bitove ulazne reči koji će biti poređeni sa bitovima odgovarajuće reči iz ulazne datoteke podataka. Na primer, u prvom koraku, od interesa je samo stanje prekidača START. Zato maske za prvi korak ima vrednost 1 na poziciji bita 6 (odgovara ulazu START) i sve nule na ostalim pozicijama.

T. 9 Sadržaj datoteke podataka za poređenje naredbe SQC

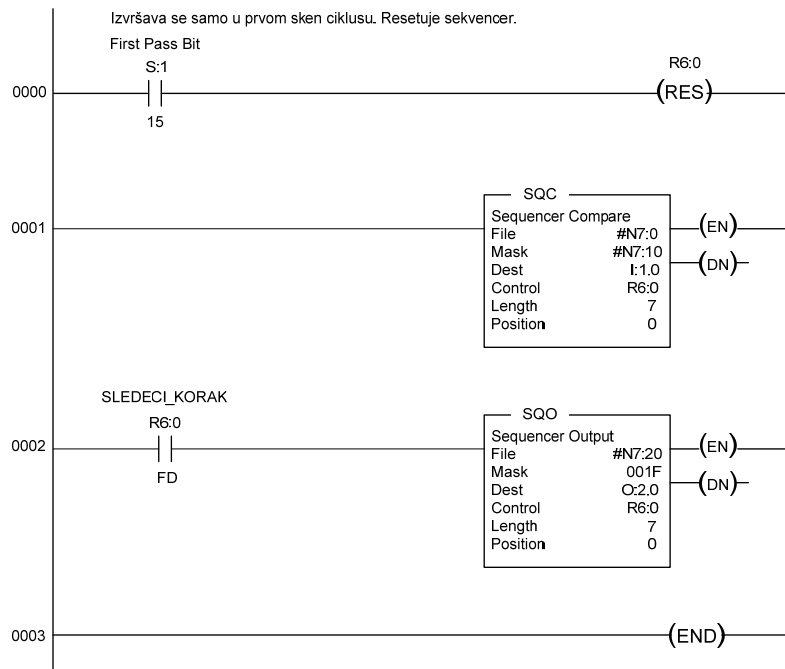
Adresa	15..7	6 START	5 SWC+	4 SWC-	3 SWB+	2 SWB-	1 SWA+	0 SWA-
N7:1	0..0	1	0	0	0	0	0	0
N7:2	0..0	0	0	0	0	0	1	0
N7:3	0..0	0	0	0	1	0	0	0
N7:4	0..0	0	1	0	0	0	0	0
N7:5	0..0	0	0	0	0	1	0	0
N7:6	0..0	0	0	1	0	0	0	0
N7:7	0..0	0	0	0	0	0	0	1

T. 10 Sadržaj datoteke maski naredbe SQC

Adresa	15..7	6 START	5 SWC+	4 SWC-	3 SWB+	2 SWB-	1 SWA+	0 SWA-
N7:11	0..0	1	0	0	0	0	0	0
N7:12	0..0	0	0	0	0	0	1	0
N7:13	0..0	0	0	0	1	0	0	0
N7:14	0..0	0	1	0	0	0	0	0
N7:15	0..0	0	0	0	0	1	0	0
N7:16	0..0	0	0	1	0	0	0	0
N7:17	0..0	0	0	0	0	0	0	1

T. 11 Sadržaj izvorne datoteke naredbe SQO.

Adresa	15..5	4 MOVE C-	5 MOVE C+	4 MOVE B-	3 MOVE B+	2 MOVE A+
N7:21	0..0	0	0	0	0	0
N7:22	0..0	0	0	0	0	1
N7:23	0..0	0	0	0	1	1
N7:24	0..0	0	1	0	1	1
N7:25	0..0	0	1	1	0	1
N7:26	0..0	1	0	1	0	1
N7:27	0..0	1	0	1	0	0



Sl. 8-11 Leder dijagram sekvencera koji koristi naredbe SQC i SQO.

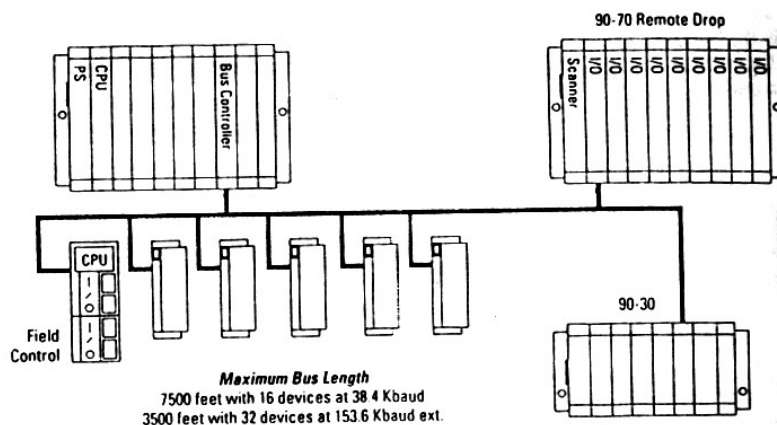
9 PLC moduli specijalne namene

9.1 Izdvojeni ulazno-izlazni moduli

Upravljanje pojedinim procesima zahteva da U/I moduli budu locirani na različitim lokacijama. U nekim slučajevima mašine su fizički udaljene. U takvim slučajevima poželjno je pozicionirati U/I modul izvan PLC kontrolera, što bliže mašini ili procesu kojom upravljaju ili koga nadgledaju. Za spregu PLC kontrolera sa izdvojenim U/I modulima koriste se specijalizovani moduli (adapteri) koji se smešaju u rek PLC kontrolera.

Veza između PLC kontrolera i izdvojenih U/I modula može biti ostvarena na različite načine. Po pravilu se koristi neka vrsta magistale za prenos podataka. Preko magistrale PLC kontroler i izdvojeni U/I moduli razmenjuju poruke. Dva uobičajena metoda za fizičko povezivanje su kablovi sa upletenim provodnicima i optički kablovi. Kablovi sa upletenim provodnicima (*twisted-pair*) su ekonomičnije rešenje. Dva provodnika su upletena jedan oko drugoga i povezana između PLC kontrolera i izdvojenog U/I modula. Upletanje smanjuje električnu interferenciju (šum). Sa druge strane, fiber-optički kablovi su imuni na šum, zato što se podaci prenose putem svetlosti. Takođe, brzina prenosa podataka fiber-optičkim kablom je daleko veća, kao i rastojanje maskimalno rastojanje između dva uređaja.

Na Sl. 9-1 je prikazan primer U/I magistrale firme GE Func. U/I magistrala čini kabl sa upletenim provodnicima koji povezuje kontroler magistrale najviše 31 izdvojeni U/I modul. Sistemi upravljanja zasnovani na konceptu izdvojenih U/I modula predstavljaju svaremeni trend u industrijskoj automatizaciji.



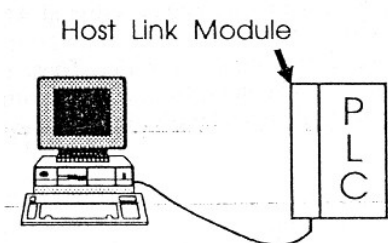
Sl. 9-1 Primer sprege PLC kontrolera sa izdvojenim U/I modulima: *GE Func Genious I/O bus*.

9.2 Komunikacioni moduli

Komunikacija postaje sve značajniji zadatak PLC kontrolera. U jednom integrisanom sistemu, javlja se potreba razmene podataka između različitih komponenti sistema. PLC kontroleri moraju biti u stanju da komuniciraju sa računarima, CNC mašinama, robotima, pa čak i sa drugim PLC kontrolerima. Na primer, u jednom fleksibilnom sistemu, PLC može da pošalje program CNC mašini. CNC mašina prima program i izvršava ga. Originalni PLC kontroleri nisu bili projektovani da obavljaju ovakve zadatke, dok su danas na raspolaganju brojni tipovi komunikacionih modula.

Razlikuju se dva tipa komunikacionih modula: *host-link* i *peer-to-peer* (od tačke ka tački).

Host-link moduli se koriste za komunikaciju PLC kontrolera sa host (glavnim) računarom. Host može biti računar ili drugi PLC kontroler (Sl. 9-2). Većina PLC kontrolera poseduju ugrađen softver koji omogućava prenos programa od hosta u PLC kontroler.

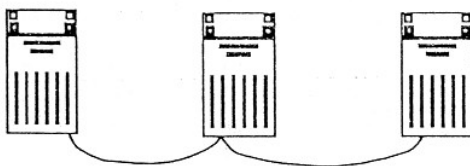


Sl. 9-2 Sprega host računara i PLC posredstvom Host-Link modula.

Tipična primena host-link modula je u realizaciji tzv. integrisanih proizvodnih ćelija. Zamislimo promenljivu u lader dijagramu. Promenljiva može da sadrži broj komada koje treba proizvesti u ćeliji. Sada zamislimo host računaru koji upisuje broj u ovu promenljivu. Ovakav scenario je omogućen host-link komunikacijom. Za host-link komunikaciju tipično se koristi RS232 komunikacija.

Host-link komunikacija se takođe može koristiti za prenos podataka od PLC kontrolera ka računaru. Na ovaj način računaru može pratiti vreme proizvodnje, broj proizvedenih komada i sl.

Peer-to-peer moduli se koriste za komunikaciju između PLC kontrolera istog tipa (Sl. 9-3). U ovakvom jednom sistemu, svakom PLC kontroleru dodeljena je jedinstvena adresa (npr. broj iz opsega 0 do 255). Većina komunikacionih modula ovog tipa nije standardizovana već predstavlja jedinstveno rešenje proizvođača PLC sistema.



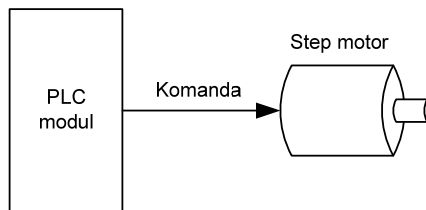
Sl. 9-3 *Peer-to-peer* komunikacija.

9.3 PLC moduli za kontrolu pozicije

Za upravljanje pozicioniranjem na raspolaganju su PLC moduli kako za upravljanje u otvorenoj tako i za upravljanje u zatvorenoj petlji. Sistemi sa zatvorenom petljom poseduju povratnu spregu što obezbeđuje da će komanda biti izvršena.

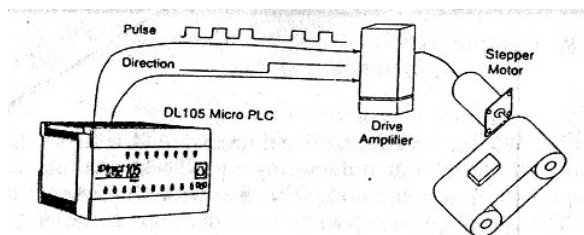
Upravljanje u otvorenoj petlji

Za upravljanje pozicijom u otvorenoj petlji koriste se step-motori u kombinaciji sa odgovarajućim PLC drajverskim modulima (Sl. 9-4). PLC modul izdaje komandu motoru i pri tome podrazumeva da će komanda biti izvršena, tj. povratna sprega ne postoji. Step motori se koriste u primenama koje ne zahtevaju veliku brzinu i veliku snagu. Preciznost pozicioniranja je veoma velika. PLC moduli za pobudu step motora su projektovani tako da u što većoj meri olakšaju integraciju step motora u sistem. Tipično, poseduju ugrađene funkcije za ubrzanje/usporenje, a u nekim varijantama i funkcije učenja. Moduli poseduju i ulaze koji se mogu koristiti za dovođenje motora u referentnu poziciju, kao i za zaštitu od prekoračenja.



Sl. 9-4 Upravljanje step motorom pomocu specijalizovanog PLC modula.

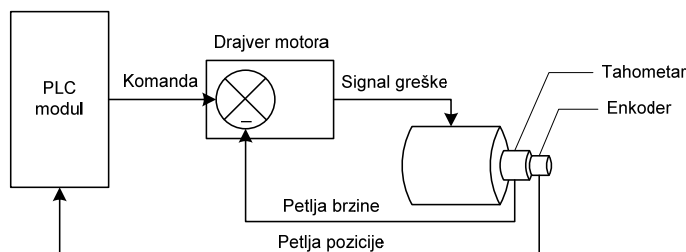
Na Sl. 9-5 je prikazana jedna tipična primena step motora. Uočimo da PLC modul generiše povorku impulsa (*Pulse*), gde svaki impuls predstavlja komandu step motoru da učini jedan korak, i signal koji određuje smer kretanja motora (*Direction*).



Sl. 9-5 Upravljanje step motorom.

Upravljanje pozicijom u zatvorenoj petlji

Tipični primeri aplikacija kod kojih se koristi pozicioniranje u zatvorenoj petlji su roboti i CNC mašine. Kod ovih primena koriste se AC ili DC motori. Na raspolaganju su PLC moduli za praćanje i upravljanje brzinom i pozicijom. U sistemima za pozicioniranje u zatvorenoj petlji postoje dve petlje (Sl. 9-6). Petlja poziciji je zatvorena (tj. nadgleda se) preko enkodera, koji je sastavni deo motora, i upravljačke jedinice (u ovom slučaju PLC). Petlja brzine se obično zatvara preko tahometra, koji je sastavni deo motora i drajvera motora. PLC zadaje brzinu kretanja, komandom u obliku analogog signal sa naponom iz opsega $-10V$ do $+10V$. Spoj za sumiranje, koji je deo drajvera motora, poredi komandu i povratni signal iz tahometra i određuje signal greške koji služi za pobudu motora. Dakle, brzina se kontinualno nadgleda i reguliše od strane drajvera motora, dok poziciju nadgleda PLC.



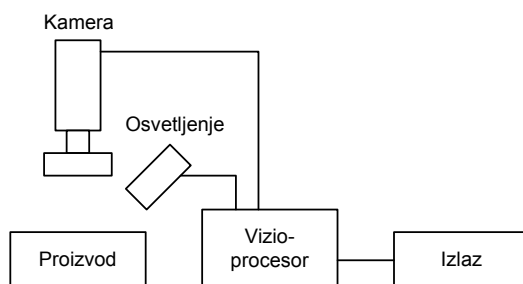
Sl. 9-6 Upravljanje pozicijom u zatvorenoj petlji pomoću specijalizovanog PLC modula.

PLC moduli za upravljanje pozicijom u zatvorenoj petlji tipično poseduju mogućnost sprege sa jednim ili dva motora. Varijante za dva motora omogućavaju upravljanje pozicijom u ravni, po dvema osama. U normalnom režimu rada, ose se nezavisno kontrolišu. U pratećem režimu rada, modul je u stanju da koordinira kretanje po osama kako bi se obavilo neko zadato, složeno ravansko kretanje.

9.4 Moduli za vizuelnu kontrolu

Vizelna inspekcija predstavlja važan zadatak proizvodnog procesa. Obično se obavlja na kraju proizvodnog procesa, u okviru izlazne kontroler kvaliteta, od strane radnika. Napredak u elektronici i računarstvu omogućio je da se pojedini zadaci vizelne kontrole danas mogu obaviti automatski. Automatizacija zadatka vizelne kontrole ima brojne prednosti u odnosu na klasičan pristup. Automatska vizelna kontrola je brža i zato se može sprovesti još u toku samog procesa proizvodnje, a ne samo na kraju kada je proizvod već sklopljen. Na bazi vizelne kontrole, moguće je uvesti korekcije u toku procesa proizvodnje i time sprečiti greške. Na taj način, kvalitete gotovih proizvoda i produktivnost mogu biti značajno poboljšani.

Sistemi za vizelnu kontrolu mogu da obave i do hiljadu inspekcija u minuti. Naročito su pogodni za primene za koje čovek nije dovoljno efikasan – onda kada zadatak treba biti obavljen brzo, kada uključuje pregled velikog broja vizuelnih detalja i kada inspekcija uključuje merenje dimenzija proizvoda i sl.



Sl. 9-7 Blok šema sistema za vizuelnu kontrolu.

Block diagram of a typical system for visual control is shown in Sl. 9-7. The system consists of: one or more cameras, illumination, video monitor and video processor. At certain time intervals, the image from the camera is transferred to the video processor which processes the image so that it can be compared with a stored template, finds characteristic details or performs certain measurements of characteristic dimensions of the product. For application in PLC systems, specialized PLC modules for visual control have been developed, which can perform automatic visual inspection at a rate of up to 1800 objects per minute. An important aspect of the PLC module for visual control is the way the system is trained. Namely, such modules are not designed for a specific intended purpose, but enable the system to learn at the location where it is used how to recognize correct products. This is done so that the image of the product is obtained on the monitor. Besides the image of the product on the monitor, there is also a larger number of icons, each representing one type of defect in visual inspection, such as simply recognizing whether the product is present or not, highlighting individual areas in the image, highlighting characteristic details and so on.

9.5 Bar-kod moduli

Automatic identification of products is gaining wider application in industry. There are several types of automatic identification, such as: vision, bar-code and radio-frequency identification (RFID). Bar-coding (Sl. 9-8) is standardly used in trade, but in recent times it is also used in industrial processes. For this reason, specialized PLC modules, so-called bar-code modules, which enable easy integration of PLC systems with bar-code readers. A typical application can be the following: bar-code module reads bar-code from a box moving on a conveyor belt. On the basis of the read bar-code, the PLC controller directs the box to the appropriate production line.



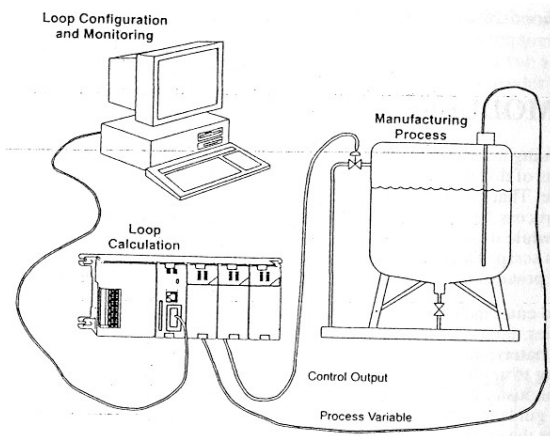
Sl. 9-8 Bar-kod zapis.

9.6 Moduli za PID upravljanje

PID is an abbreviation that refers to the technique of automatic control known under the name of proportional-integral-derivative control. Producers of PLC systems approach the problem of PID control in two different ways: some producers offer specialized modules for PID control, while others use standard analog/digital U/I modules with specialized software for PID control. PID can be used for controlling physical quantities such as: temperature, pressure, concentration, humidity. PID is widely used in industry so that it can be achieved precise control under the most unfavorable conditions and disturbances that can occur during the production process. In essence, PID is an equation that the control unit uses to calculate the output, the control quantity on the basis of the error signal. The controlled quantity (e.g. temperature) is measured, and the measured value is fed back to the control unit in the form of a return signal. The control unit compares the measured value of the controlled quantity with the setpoint value and generates

signal greške. Greška se ispituje na tri različita načina: proporcionalni, integracioni i diferencijalni. Svaki od ova tri faktora se može tretirati kao pojačanje koje na neki specifičan način utiče na izračunavanje odziva na datu grešku. Upravljačka jedinica koristi ova pojačanja (proporcionalno pojačanje, integraciono pojačanje i diferencijalno pojačanje) za izračunavanje vrednosti komande (izlaznog signala) za korekciju izmerene greške.

Tipičan sistem sa PID upravljanjem prikazan je na Sl. 9-9. Sistem je rezervoar u kome se obavlja sagorevanje tečnog goriva. Za upravljanje sistemom koristi se PLC. Logička jedinica PLC kontolera dobija vrednost izmerene temperature od ulaznog analognog modula, obavlja PID izračunavanje, generiše signal greške koga šalje izlaznom modulu (digitalnom ili analognom) koji upravlja ventilom preko koga se u rezervoar dovodi gorivo. Podešavanje sistema i nagledanje rada sisteme obavlja se uz pomoć izdvojenog računara koji je sa PLC kontrolerom u vezi preko host-link komunikacije.



Sl. 9-9 Blok dijagram procesa kojim se upravlja PID kontrolerom. (Loop Configuration and Monitoring – konfiguracija i nadgledanje petlje; Loop Calculation – izračunavanje parametara petlje; Manufacturing Process – proizvodni proces; Process Variable – promenljiva procesa; Control Output – upravljački izlaz)

Proporcionalno upravljanje uzima u obzir samo magintudu signala greške i obično ima najveći uticaj na rad sistema. Proporcionalno upravljanje reaguje srazmerno trenutom iznosu greške. Što je greška veća, veći je i odziv. Proporcionalno upravljanje obično nije dovoljno za uspešno upravljanje sistemom. Proporcionalno upravljanje nije u stanju da koriguje male greške, tzv. *offset-e* ili greške stabilnog-stanja. To su greške, čiji je iznos mali, ali se njihov efekat akumulira tokom vremena.

Integraciono upravljanje ima zadatak da koriguje upravo *offset-e*, koji ne mogu biti korigovani proporcionalnim upravljanjem. Integraciono upravljanje predstavlja reakciju sistema za integral greške (proizvod greške i vremena). Male greške koje kratko traju nisu od značaja. Međutim, čak iako je greška mala ali je zato stalno prisutna, vremenom njen značaj raste i značajnije utiče na odziv. Integracioni faktor (zove se i brzina resetovanja – *reset rate*) se može podešavati. Brzina resetovanja ima dimenziju vremena i što je manja to je i korekcija akumulirane greške brža. Međutim, isuviše male vrednosti mogu uzrokovati nestabilnosti u radu sistema.

Diferencijalno upravljanje reaguje na brzinu promene greške. Naime, u slučajevima kada je deluju nagli i jaki proremećaji, proporcionalno upravljanje nije dovoljno da koriguje grešku. Proporcionalno upravljanje reaguje samo na trenutnu vrednost greške, pa se zato može desiti da i pored oziva koji teži da smanji grešku, greška nastavlja da raste. Drugim rečima, proporcionalno upravljanje nije u mogućnosti da predviti šta će se desiti u budućnosti kako bi pojačanim odzivom predupredilo očekivanu grešku. Diferencijalno upavljanje usloviće da odziv sistema bude veći kada je brzina promene greške velika, nego kada je brzina promene greške mala. Drugim rečima, diferencijalno upravljanje nadgleda tendenciju promene greške i utiče na odziv sistema tako da se u izračunavanje odziva uključi i pretpostavljeno ponašanje sistema u bliskoj budućnosti. Diferencijalno upravljanje je značajno kod brzih sistema koji moraju brzo korigovati greške. Kod takvih sistema proporcionalno pojačanje mora biti veliko. Međutim, veliko proporcionalno pojačanje može usloviti premašenje. Svrha diferencijalnog upravljanja je upravo eliminacija premašenja.

PID jednačina je sledećeg oblika:

$$C_o = K \left(E + \frac{1}{T_i} \int_0^t E dt + KD(E - E(n-1)) / dt \right) + const, \text{ gde je}$$

C_o – izlazna veličina

E – sistemska greška

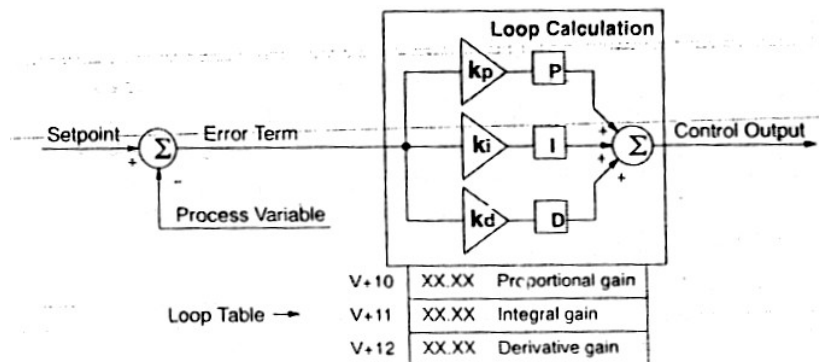
$E(n-1)$ - greška poslednjeg odmerka

K – ukupno (proporcionalno) pojačanje

$1/T_i$ – integracioni član

KD – diferencijalni član

dt – vreme između dva odmeravanja signala greške

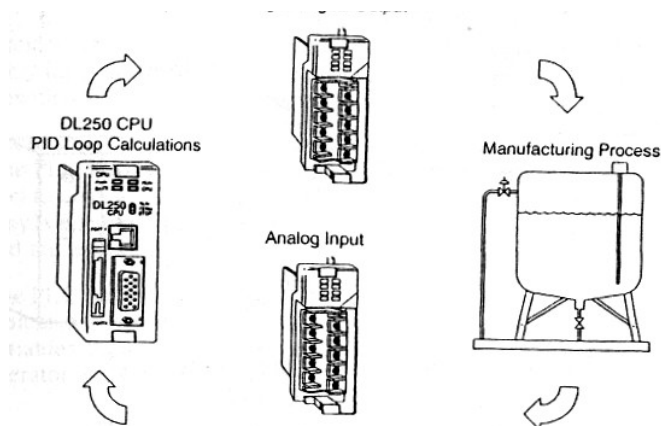


Sl. 9-10 Blok dijagram PID izračunavanja.

Na Sl. 9-10 je prikazano kao se u PLC kontroleru realizuje PID algoritam. Uočimo tabelu (*loop table*) koja sadrže vrednosti za svako od tri pojačanja. PID algoritam uvek radi na isti način, a jedine promenljive ovog algoritma su vrednosti pojačanja i učestanost odmeravanja signala greške. Podešavanje PID kontrolera se sastoji u izboru vrednosti pojačanja koje će obezbediti optimalne performanse sistema u konkretnim dinamičkim uslovima. Podešavanje pojedinačnih pojačanja se obavlja uz pomoć potenciometra ili direktnim unosom, preko tastature, numeričkih vrednosti pojačanja. U opštem slučaju procedura podešavanja se odvija na sledeći način:

1. Postaviti sva pojačanja na nulu.
2. Povećavati proporcionalno pojačanje sve dok sistem ne počne da osciluje
3. Smanjiti proporcionalno pojačanje tako da sistem prestane da osciluje, a zatim dodatno smanjiti proporcionalno pojačanje za 20%.
4. Povećati diferencijalni faktor kako bi se poboljšala stabilnost sistema
5. Povećavati integracioni faktor sve dok sistem ne dostigne tačku nestabilnosti, zatim neznatno smanjiti integracioni faktor.

Većina PLC sistema nudi mogućnost PID upravljanja u obliku softvera i analognih U/I modula (Sl. 9-11). Manji broj proizvođača PLC kontrolera nudi specijalizovane PID module.

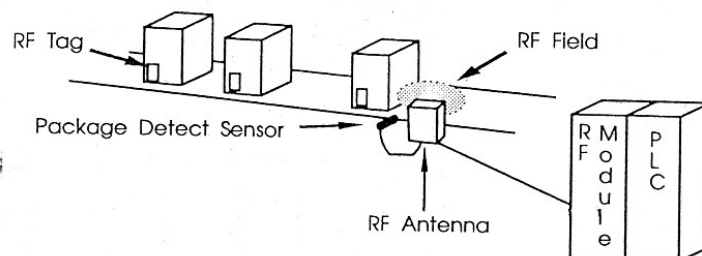


Sl. 9-11 Komponente PLC sistema neophodne za softversku realizaciju PID upravljanja.

9.7 Radio-frekvencijski moduli

Radio-frekvencijski (RF) moduli se koriste za identifikaciju objekata, praćenje proizvoda, kontrolu proizvodnje i drugo. Oni eliminišu neke od problema koji su karakteristični za bar-kodove. Bar kodovi moraju biti čisti, a očitavanje bar koda zahteva dobru optičku vidljivost. RF moduli nisu osetljivi na prašinu, ulje i druge oblike zaprljanja. Takođe, nisu osetljivi na elektromagnetski šum, koji se generiše u fabričkim postrojenjima, a potiče od motora, sklopki, transformatora.

Sistem za RF identifikaciju se sastoji od *RF tagova* (priveska) koji se mogu lako prikačiti za objekte ili proizvode. RF tagovi se realizuju kao poluprovodnički čipovi koji se mogu identifikovati RF modulom kada se nađu u njegovoj blizini. Između taga i RF modula uspostavlja se radio komunikacija, putem koje je moguće pročitati informacije zapisane u tagu ili čak upisati neku informaciju u tag. Tipična aplikacija prikazana je na slici. Duž pokretne trake kreću se proizvodi koji jedan za drugim ulaze u RF polje. Dodatni senzor (optički, induktivni,..) detektuje prisustvo proizvoda i aktivira antenu preko koje se obavlja čitanje/upis taga.



Sl. 9-12 Tipična primena RF modula. (*Package Detect Senzor - senzor za detekciju paketa*)

Zamislimo proces proizvodnje automobila i RF tag na automobilu koji se kreće duž proizvodne linije. Tag može da sadrži, osim identifikacionog broja automobila i neke karakteristične informacije koje se tiču specifikacije, uključujući na primer opcije koje je kupac naručio. Kako se automobil kreće od jedne do druge proizvodne ćelije, tako svaka ćelija može da očita sadržaj taga i shodno tome ugradi ili ne odgovarajuću opciju. Kada automobil dođe do kraja proizvodne linije, u tag se može upisati celokupna istorija njegove proizvodnje (koji sastavni delovi su ugrađeni, i sl.). Tipično, tagovi koji se koriste u industriji poseduju memoriju od 100 do 2000 bajtova.

Postoje dva tipa RF tagova: *aktivni* i *pasivni*. Aktivni tag sadrži bateriju, dok je pasivni tag ne sadrži. Za napajanje pasivnog taga se koristi elektromagnetno polje koje emituje RF modul. Drugim rečima, pasivni tag je operativan samo dok se nalazi u neposrednoj blizini RF modula. Zbog toga je domet RF taga relativno mali. Aktivni tagovi su skuplji, ali je njihov domet veći.

9.8 Operatorski terminali

Operatorski terminali nalaze se u proizvodnom programu mnogih proizvođača PLC sistema. Najjednostavniji tipovi operatorskih terminala su u mogućnosti da prikazuju samo kratke tekstualne poruke. Složeniji moduli imaju mogućnost prikaza grafike i teksta uz dodatnu mogućnost prihvatanja operatorskog ulaza sa tastature, touch-screen ekrana, bar-kod čitača i td. Kod sistema koji se upravljaju PLC kontrolerima, mnoge bitne informacije o procesu se nalaze memoriji PLC kontrolera. Operatorski terminali predstavljaju neku vrstu prozora u memoriju PLC kontrolera.

Za formiranje prikaza koriste se specijalizovani razvojni softveri koji se izvršavaju na PC računarima. Ovi softveri omogućavaju korisniku da nacрта grafički prikaz i odluči koje će promenljive iz PLC kontrolera biti prikazane. Korisnik, takođe, definiše kakav ulaz je neophodan od strane operatera. Nakon što je prikaz dizajniran, on se može preneti u permanentnu memoriju operatorskog terminala. Ovakvi tipovi terminala mogu u svojoj memoriji da zapamte na stotine stranica. PLC, slanjem odgovarajućih poruka daje instrukcije terminalu koju od zapamćenih stranica i koje informacije da prikaže na ekranu. Ekрани obično sadrže grafičke prikaze pojedinih delova procesa, dok promenljive tipično predstavljaju neka karakteristična vremena, broj proizvedenih komada ili bilo koju drugu informaciju koja može biti od pomoći operateru u praćenju procesa ili pranalazanju eventualnih otkaza.