

Leder programiranje

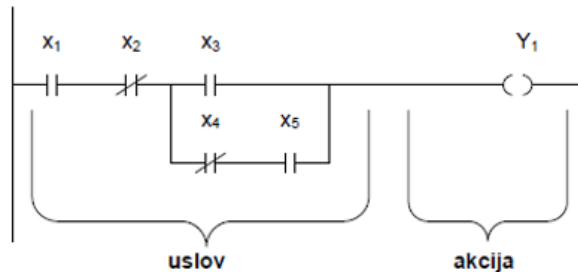
Ako se PLC posmatra kao mikroprocesorski sistem, što on i jeste, onda bi se moglo očekivati da se za njegovo programiranje koriste standardni programski jezici. Međutim, ako se pođe od činjenice da je PLC projektovan kao namenski mikroprocesorski sistem za upravljanje i nadzor rada nekog procesa, i da u skladu s 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.

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 poređani 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. 4-16).

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 slici ulaza (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 “ILP” operacija. To znači da će iskaz predstavljen nizom paralelnih grana biti istini, 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.



Sl. 4-16 Leder rang

Akcija – Grafički simboli na desnoj strani ranga odnose se ili na fizički izlaz (promenljive smeštene u slici izlaza (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 levoj 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.

Leder program se izvršava u toku programskog dela sken ciklusa i to tako što se obrađuje rang po rang po redosledu kako su oni poređani u leder dijagramu. U svakom rangu ispituje se istinitost uslova i ukoliko je uslov istinit izvršavaju se odgovarajuće naredbe u desnom delu ranga. To znači da promenljive u 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 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.

Bit naredbe

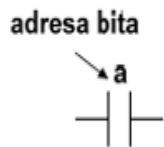
Bit naredbe su, kao što samo ime kaže naredbe čiji su operandi bitovi. S gledišta lokacije operanada, to znači da se oni najčešće nalaze u datoteci 3 (bit file - B), digitalnim ulaznim ili izlaznim datotekama (input image file 1 ili output image file 0) ili u korisničkim datotekama bit tipa. Pored toga, 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).

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)

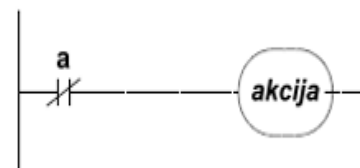
Grafički simbol



Tablica istinitosti

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

Položaj u rangu

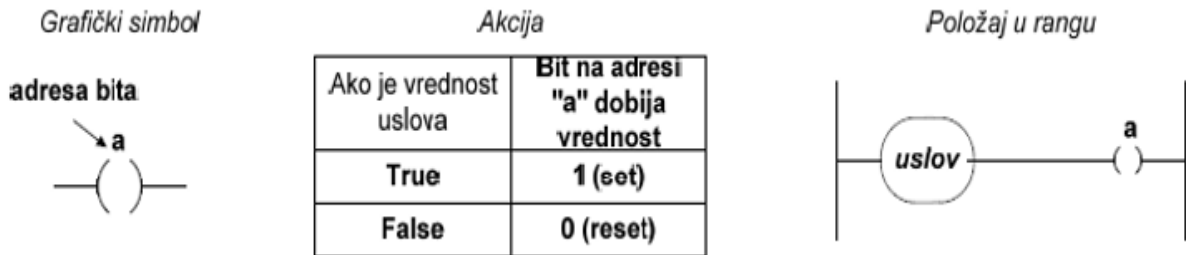


Nazivi ove dve naredbe potiču od ispitivanja binarnih signala koji dolaze sa prekidačkih kola. U tom smislu, XIC naredba se odnosi na normalno otvoren prekidač, dok se XIO naredba odnosi na normalno zatvoren prekidač.

Bit naredbe za postavljanje vrednosti izlaza

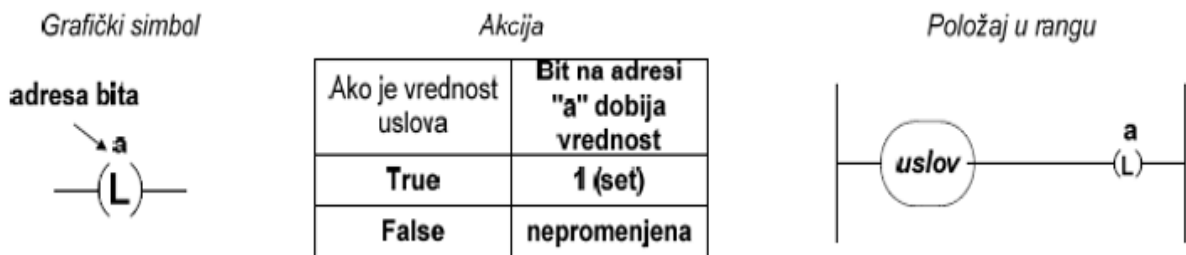
Ovim naredbama se bitu čija je adresa navedena u naredbi dodeljuje vrednost 1 ili 0. Podsetimo se da se ove naredbe nalaze na desnoj strani ranga, što znači da će se one izvršiti samo ako je iskaz (uslov) na levoj strani ranga istinit.

OPE - Output energize (pobuđivanje izlaza)



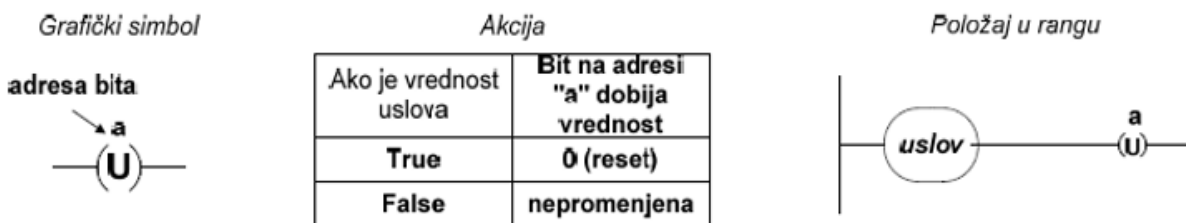
Potrebno je da se zapazi da se ovom naredbom vrednost bita čija je adresa "a" može promeniti samo jedanput za vreme sken ciklusa. Ova vrednost ostaće neizmenjena sve do sledećeg sken ciklusa, kada će se pri skeniranju odgovarajućeg ranga ponovo ispitati uslov i izvesti odgovarajuća akcija.

OTL - Output latch (pamćenje izlaza)



OTL naredbom se adresirani bit može isključivo postaviti na 1. Naime, za razliku od OPE 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 adresirani 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.

Bit trigger naredba

OSR - One-shot rising (uzlazna ivica)



OSR naredba omogućava da se obezbedi izvođenje neke akcije samo jedanput. Ova naredba je specifična po tome što 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.