

**UNIVERZITET U BIHAĆU**  
**TEHNIČKI FAKULTET**  
**BIHAĆ**

# **RAČUNARSKO VOĐENJE PROCESA**

**Auditorne/Laboratorijske vježbe**

**LADDER programiranje (Vremenske instrukcije)**  
**(Vježba 12)**

**mr. Amel Toroman, dipl.ing.el.**  
**Viši asistent**

**Ak. 2021/2022**

## LADDER (LAD) PROGRAMIRANJE

### SATNE INSTRUKCIJE – *Clock instructions*

#### Čitanje i setovanje sata realnog vremena

Instrukcija čitanja sata realnog vremena (TODR) čita trenutno vrijeme i datum iz hardverskog sata i upisuje ih u vremenski *bafer* dužine 8 bajta, počevši od adrese T. Setovanje (postavljanje) sata realnog vremena (TODW) upisuje trenutno vrijeme i datum u hardverski sat, počevši od osmog bajta vremenskog *bafera* sa adresom T.

Treba voditi računa o ispravnosti datuma, jer neki PLC-ovi ne vrše ovu provjeru, tako da datum, kao npr. 30. Februar može biti prihvaćen.

Ne treba koristiti TODR/TODW instrukcije u isto vrijeme i u glavnom programu i u rutini jer će doći do greške (*non-fatal error 0007*). Također, ako se u korisničkom programu navode godine, treba imati na umu da neki PLC-ovi uzimaju u obzir samo posljednje dvije cifre iz godine.

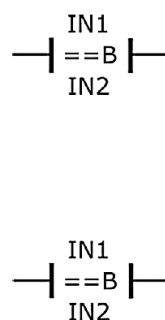
### INSTRUKCIJE POREĐENJA

#### Poređenje brojnih vrijednosti

Instrukcije poređenja koriste se za poređenje dvije vrijednosti, i to sa sljedećim relacijama:

IN1 = IN2	IN1 >= IN2	IN1 <= IN2
IN1 > IN2	IN1 < IN2	IN1 >< IN2

Kada je poređenje istinito, instrukcija poređenja uključuje kontakt.

	==B	==I	==D	==R
	<>B	<>I	<>D	<>R
	>=B	>=I	>=D	>=R
	<=B	<=I	<=D	<=R
	>B	>I	>D	>R
	<B	<I	<D	<R
	==S	<>S		

Slika 1. Simboli instrukcija poređenja

Mora se voditi računa kod pravilnog inicijalizovanja pointera i vrijednosti koje sadrže realne brojeve prije izvršavanja instrukcija poređenja koje koriste ove vrijednosti, jer u slučaju greške pojedini PLC-ovi odmah prekidaju danje izvršavanje korisničkog programa.

Instrukcije poređenja se izvršavaju nezavisno od stanja toka struje.

## Poređenje niza znakova (string)

Ove instrukcije porede dva niza ASCII karaktera i to:

$IN1 = IN2$

$IN1 \lessgtr IN2$

Kada je poređenje istinito, instrukcija poređenja uključuje kontakt. Ovdje se mora voditi računa da string ne bude duži od 254 karaktera, i da početna adresa stringa i njegova dužina mogu da stanu u dodijeljeni memorijski prostor. I ove instrukcije poređenja se izvršavaju bez obzira na stanje toka struje.

## VREMENSKE INSTRUKCIJE (TAJMERI)

Vremenske instrukcije (naredbe) 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 da se istakne da se sam tajmer i način njegovog rada definiše preko naredbe koja se uvrštava u *leder* program. Drugim riječima, kad se u program stavi jedna od moguće tri naredbe i u njoj naznači adresa tajmera u odgovarajućem formatu, onda operativni sistem sam zauzme tri riječi u datoteci koja je navedena u adresi.

Zadana vrijednost (*preset value - PRE*) je vrijednost kojom se definiše željeni broj osnovnog intervala vremena (čime se određuje ukupno vrijeme koje tajmer treba da izmjeri), odnosno ukupni broj događaja koje brojač treba da registruje prije nego što se generiše signal koji označava da su tajmer ili brojač završili rad.

Zadana vrijednost za tajmer može da se kreće u intervalu od 0 do +32767, dok se zadana vrijednost za brojač kreće u opsegu od -32768 do +32767.

Akumulirana vrijednost (*accumulated value - ACC*) predstavlja broj osnovnih vremenskih intervala koje je tajmer izbrojao, odnosno broj događaja koje brojač registrovao u nekom trenutku. Kada akumulirana vrijednost postane veća ili jednaka od zadane vrijednosti tajmer, odnosno brojač, završavaju svoj rad.

Opseg dozvoljenih vrijednosti za akumuliranu vrijednosti isti je kao i za zadanu vrijednost.

## Način rada tajmera

Sve dok tajmer radi u svakom sken ciklusu povećava se akumulirana vrijednost. Pri tome, iznos za koji će se povećati ACC vrijednost zavisi od dužine trajanja sken ciklusa. Naime, kada se prilikom obrade ranga ustanovi da su se stekli uslovi da tajmer počne sa radom onda se istovremeno startuje jedan interni tajmer, koji se ažurira preko prekida (interapta) na svakih 0,01 sec. Broj registrovanih vremenskih intervala se smješta u interni 8-bitni registar (bitovi 0-7 u prvoj riječi). Ukoliko je u pitanju tajmer čija je vremenska baza 0,01 sec, onda se u sljedećem programskom skenu, kada se naiđe na dati rang, vrijednost internog registra, koja zapravo predstavlja interval vremena koji je protekao između dva sukcesivna sken-a, dodaje akumuliranoj vrijednosti. Nakon toga se interni registar resetuje na nulu i počinje ponovo da mjeri vrijeme do sljedećeg skena.

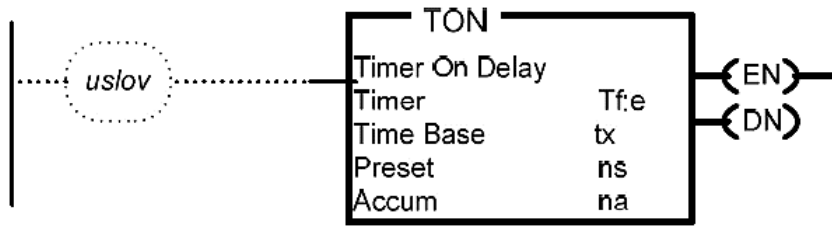
Budući da je maksimalna vrijednost koju može da ima interni registar oko 2,5 sec ( $255 \times 0,01$ ), može se očekivati da će tajmer raditi ispravno samo ako sken ciklus ne traje duže od 2,5 sekundi. Ukoliko se tajmer koristi u programu čiji sken ciklus traje duže, onda je neophodno da se ista naredba za tajmer postavi na više mjesta u programu čime će se obezbijediti da se rangovi koji sadrže taj tajmer obrađuju sa učestanošću koja nije veća od 2,5 sekundi. (Svaki put kada se prođe kroz rang koji sadrži tajmer izvršiće se ažuriranje internog registra i povećati akumulirana vrijednost).

Ukoliko tajmer radi sa vremenskom bazom od 1 sekunde obrada tajmera je donekle složenija. Ovde se, naime i dalje koristi interni tajmer koji se ažurira na svakih 0,01 sekundi, ali se pri tome u toku obrade ranga akumulirana vrijednost ažurira samo ako je akumulirana vrijednost veća ili jednaka od 1 sekunde. Pri tome se akumulirana vrijednost uvećava za 1, dok se eventualni ostatak vremena pamti u internom brojaču i na njega se dodaju sljedeći inkrementi od po 0,01 sekunde. Postupak ažuriranja akumulirane vrijednosti je takav da se može očekivati da će tajmer raditi ispravno ako sken ciklus ne traje duže od 1,5 sekundi (maksimalni mogući ostatak poslije očitavanja je 0,99, što uvećano za 1,5 čini 2,5). Naravno, i ovdje se problem ciklusa dužeg trajanja može prevazići stavljanjem naredbe tajmera na više mjesta u programu.

Potrebno je da se naglasi da je pri korištenju tajmera 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 tajmer. Jasno je da se u tom slučaju neće vršiti ažuriranje akumulirane vrijednosti. To nadalje znači da je neophodno da se obezbijedi da u slučaju bilo kakvog programskog skoka, naredba za tajmer ne bude isključena iz obrade u periodu koji je duži od maksimalno dozvoljenog vremena.

Tačnost tajmera (*timer accuracy*) je pojam koji se odnosi na dužinu vremenskog intervala koji protekne od trenutka kada se tajmer uključi do trenutka kada DN bit indicira da je mjerenje vremena završeno. Kao što je već istaknuto, za tajmere koji rade sa vremenskom bazom od 0,01 sekunde tačnost je u granicama od  $\pm 0,01$  sec sve dok sken ciklus ne traje duže od 2,5 sekunde. Tajmeri koji rade sa vremenskom bazom od 1 sekunde zadržavaju svoju tačnost ukoliko je programski sken kraći od 1, 5 sec.

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



*Slika 2. Tajmer TON*

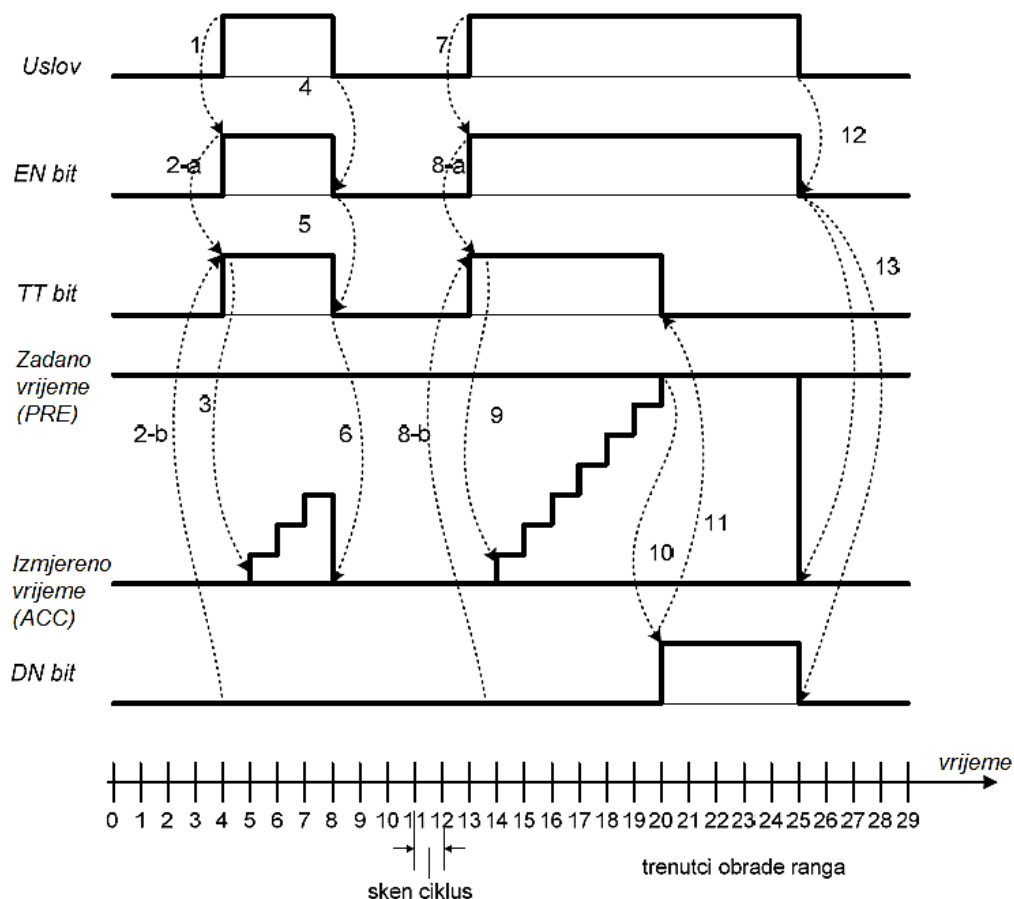
Kao što je već rečeno, stavljanjem ove naredbe u *leder* program automatski se definiše prva vrsta tajmera i zauzimaju tri riječi koje čine elemenat broj *e* u datoteci tajmera broj *f*. Prilikom formiranja naredbe specificiraju se i *vremenska baza (tx)* i *zadana vrijednost (ns)*. *Akumulirana vrijednost (na)* se automatski postavlja na 0.

TON naredba započinje rad tajmera (prebrojavanje osnovnih vremenskih intervala) za vrijeme onog programskog sken ciklusa u kome uslov u rang u kome se naredba nalazi prvi put postaje istinit (prelaz neistinit/istinit – uzlazna ivica). U svakom sljedećem sken ciklusu, sve dok je uslov istinit tajmer vrši ažuriranje akumulirane vrijednosti (ACC) u skladu sa proteklom vremenom između dva ciklusa.

Kada akumulirana vrijednost dostigne zadanu vrijednost, 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 vrijednost se postavlja na 0, bez obzira da li je tajmer prije toga izmjerio zahtjevano vrijeme ili nije.

Bitovi stanja tajmera mijenjaju se u toku programskog sken ciklusa na sljedeći način:

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



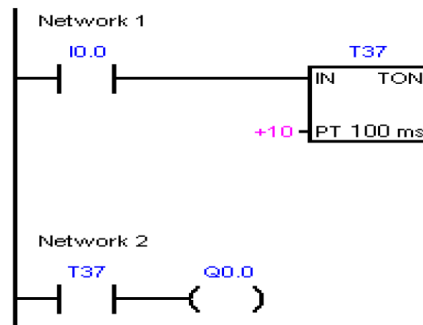
**Slika 3.** Vremenski dijagram izvršavanja TON naredbe

U vezi sa radom tajmera potrebno je da se zapazi nekoliko činjenica. Prije svega *tajmer radi samo dok je uslov istinit* (signal na ulazu u tajmer je u stanju “on”).

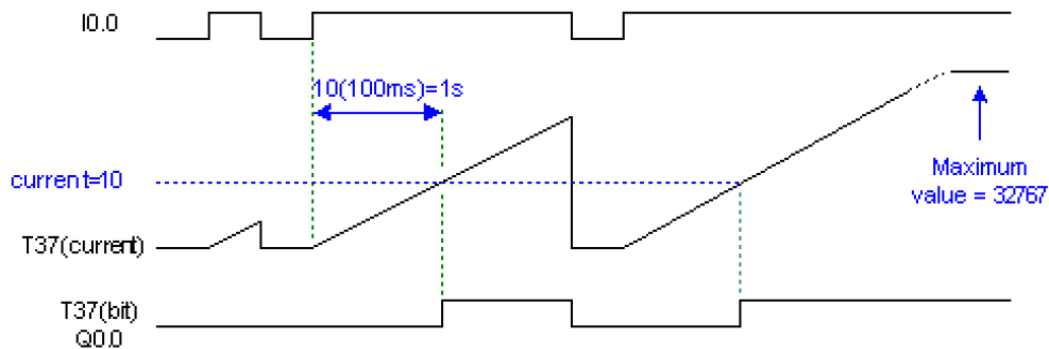
Istonsnu vrijednost uslova pokazuje EN bit. Drugim riječima, ovaj bit ima vrijednost 1 onda kada je *uslov istinit* i to označava da je rad tajmera *omogućen* (enable). Kada je *uslov neistinit*, EN bit ima vrijednost 0, što znači da je rad tajmera *onemogućen*. Međutim, činjenica da EN bit ima vrijednost 1 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 *istinit*. Rad tajmera indicira TT bit. Naime, taj bit je postavljen na 1 za cijelo vrijeme za koje *tajmer aktivno mjeri vreme* (timer timing), i postavlja se na 0 kada tajmer ne radi.

Konačno, kada je vrijednost DN bita 1, onda to znači da je tajmer *završio* (done) svoj posao, tj. izmjerio zadano vrijeme. Pri tome, DN bit ne govori o tome *kada* je tajmer završio sa poslom, jer će on ostati na vrijednosti 1 sve dok *uslov* ne postane *neistinit*.

Vremenski dijagram rada tajmera prikazan je na *Slici 3*. Stanje tajmera se može resetovati posebnom *RES naredbom*.



*Slika 4. Primjena TON tajmera*



*Slika 5. Vremenski dijagram TON tajmera za primjer sa Slike 4*

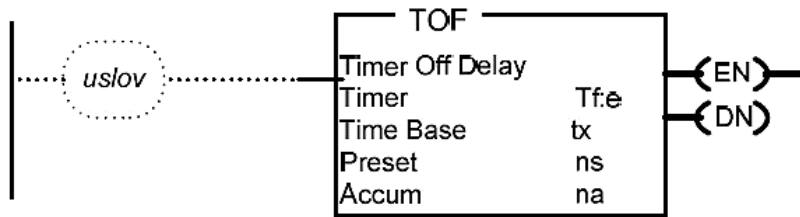
**TON** počinje da broji vrijeme kada je „IN“ aktivan. Bit koji zauzima tajmer (Txx) postaje aktivan onda kada je tajmer izbrojao veću ili jednaku vrijednost od zadate PT (*preset time*). Kada je ulaz „IN“ tajmera neaktivan nema ni brojanja, a u memorijsku lokaciju koja pripada tom tajmeru (Txx) upisana je logička „0“.

*TON nastavlja da broji pošto je postignuta vrijednost PT i može da broji do maks. vrijednosti 32767.*

- 100 ms predstavlja rezoluciju tajmera tajmer ima tri različite rezolucije: 1 ms, 10 ms i 100 ms.
- Izlaz Q0.0 će postati aktivan tek 1 sekundu nakon što da ulaz I0.0 postane aktivan



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



Slika 6. Tajmer TOF

Ovom naredbom se definiše druga vrsta tajmera i zauzimaju tri riječi koje čine element broj *e* u datoteci tajmera broj *f*. Prilikom formiranja naredbe specificiraju se i *vremenska baza* (*tx*) i *zadana vrijednost* (*ns*). *Akumulirana vrijednost* (*na*) se automatski postavlja na 0.

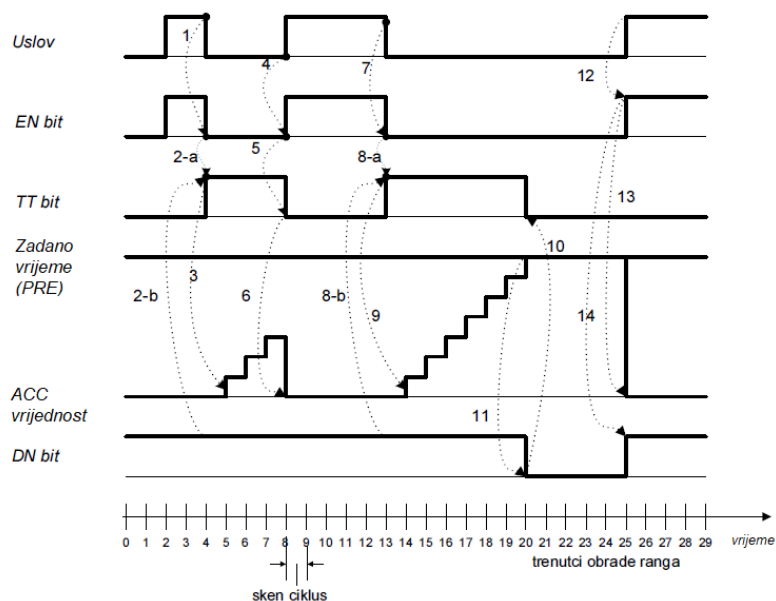
TOF naredba započinje rad tajmera za vrijeme onog programskog sken ciklusa u kome uslov u rangu u kome se naredba nalazi prvi put postaje neistinit (prelaz istinit/neistinit – silazna ivica). U svakom sljedećem sken ciklusu, sve dok je uslov neistinit tajmer vrši ažuriranje akumulirane vrijednosti (ACC) u skladu sa proteklom vremenom između dva ciklusa.

Kada akumulirana vrijednost dostigne zadanu vrijednost, tajmer prekida svoj rad. Pri tome, ako u nekom sken ciklusu uslov postane istinit, tajmer prekida svoj rad i akumulirana vrijednost se postavlja na 0, bez obzira da li je tajmer prije toga izmjerio zahtjevano vrijeme ili nije.

Bitovi stanja tajmera mijenjaju se u toku programskog sken ciklusa na sljedeći način:

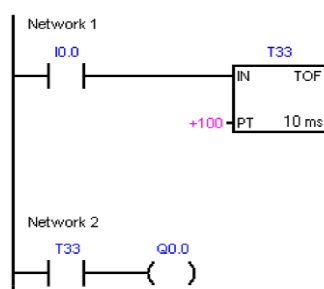
- DN - *Timer done* bit se postavlja na 1 kada je uslov istinit. On se resetuje na 0 kada je uslov neistinit i pri tome je  $ACC \geq PRE$ .
- EN - *Timer enable* bit se postavlja na 1 kada je uslov istinit, i resetuje na 0 kada je uslov neistinit.
- TT - *Timer timing* bit se postavlja na 1 kada je 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 tajmera potrebno je da se zapazi nekoliko činjenica. Prije svega tajmer radi samo dok je uslov neistinit (signal na ulazu u tajmer je u stanju “off”). Istonsnu vrijednost uslova pokazuje EN, ali za razliku od TON naredbe, ovdje on onemogućava rad tajmera. Drugim riječima, ovaj bit ima vrijednost 1 onda kada je uslov istinit i to označava da je rad tajmera onemogućen. Kada je uslov neistinit, EN bit ima vrijednost 0, što znači da je rad tajmera omogućen. Međutim, činjenica da EN bit ima vrijednost 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 cijelo vrijeme za koje tajmer aktivno mjeri vrijeme (*timer timing*), i postavlja se na 0 kada tajmer ne radi. Konačno, kada je vrijednost DN bita 0, onda to znači da je tajmer završio (*done*) svoj posao, tj. izmjerio zadano vrijeme. Pri tome, DN bit ne govori o tome kada je tajmer završio sa mjerenjem vremena, jer će on ostati na vrijednosti 0 sve dok uslov ne postane istinit.

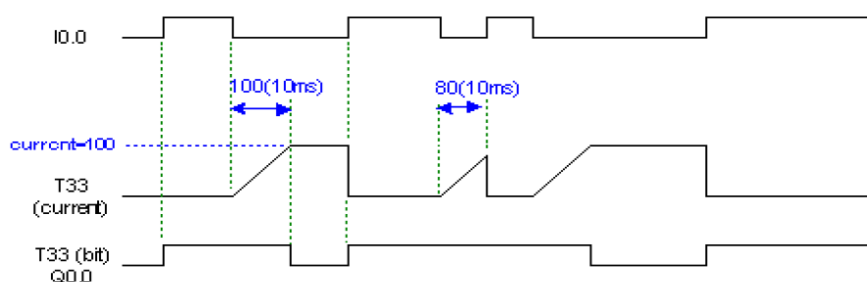


Slika 7. Vremenski dijagram izvršavanja TOF naredbe

Vremenski dijagram rada tajmera prikazan je na Slici 7.



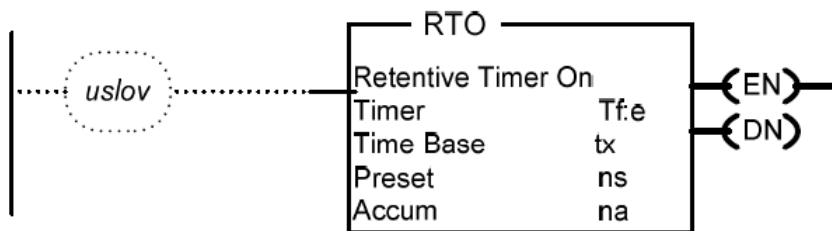
Slika 8. Primjena TOF tajmera



Slika 9. Vremenski dijagram TOF tajmera za primjer sa Slike 8

- TOF služi da produži aktivno stanje tj. bit Txx za vremenski period PT od trenutka kada je neaktivan tajmer (IN postao „0“)
- Bit koji odgovara TOF (Txx) postaje aktivan čim se aktivira „IN“.
- Ovo stanje se zadržava još za vremenski period PT pošto je *enable* bit (IN) postao neaktivan.
- Bez obzira šta se dešava sa *enable* bitom (IN) stanje na TOF ostaje aktivno još za PT od trenutka kad se „IN“ postavi na „0“.

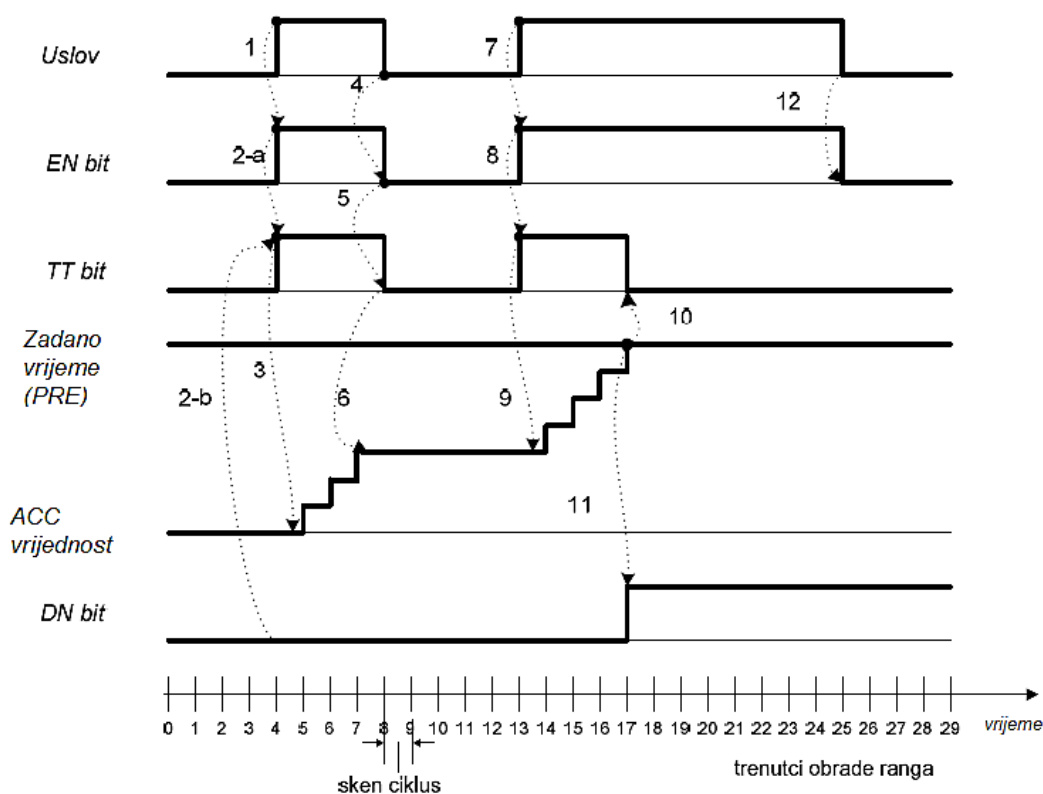
- **Retentive Timer (RTO) ili The Retentive On-Delay Timer (TONR)**



Slika 10. Tajmer RTO

Ovom naredbom se definiše treća vrsta tajmera i zauzimaju tri riječi koje čine elemenat broj *e* u datoteci tajmera broj *f*. Prilikom formiranja naredbe specificiraju se i *vremenska baza* (*tx*) i *zadana vrijednost* (*ns*). *Akumulirana vrijednost* (*na*) se automatski postavlja na 0.

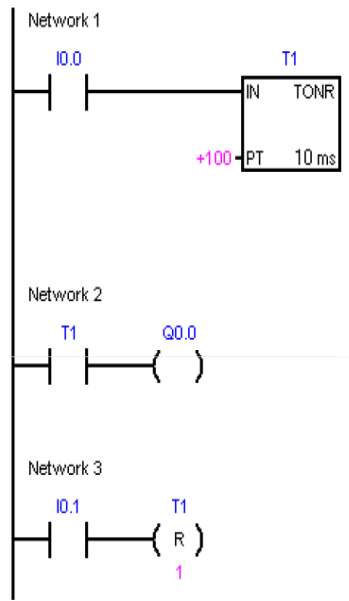
RTO naredba razlikuje se od TON naredbe samo po tome što se akumulirana vrijednost ne resetuje, već zadržava i onda kada *uslov* postane *neistinit* (Slika 11). Drugim riječima, ovaj tajmer počinje da radi kada *uslov* postane *istinit*, i nastavlja sa radom povećavajući akumuliranu vrijednost sve dok je *uslov* *istinit*. Kada *uslov* postane *neistinit*, tajmer *prekida rad*, ali se akumulirana vrijednost pri tome ne mijenja. To znači da će kada *uslov* ponovo postane *istinit*, tajmer nastaviti sa radom i prethodno izmjerenom vremenu (ACC) dodavati nove vrijednosti. Na taj način ovaj tajmer omogućuje da se kumulativno mjere intervali vremena u kojima je *uslov* bio *istinit* (Slika 11).



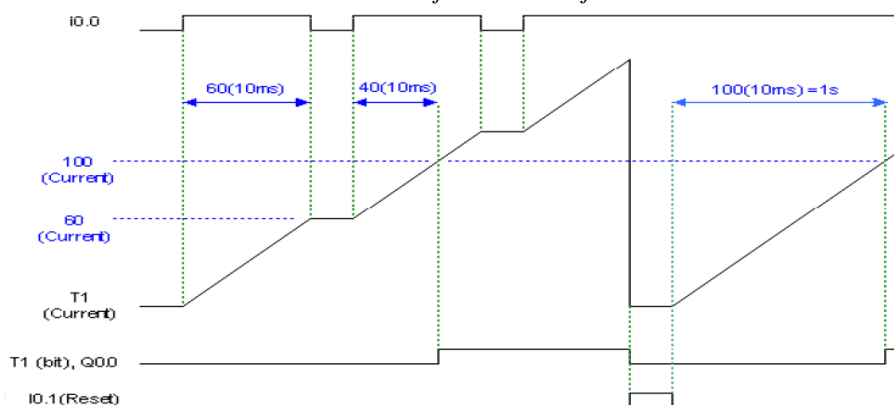
Slika 11. Vremenski dijagram izvršavanja RTO naredbe

Bitovi stanja tajmera mijenjaju se u toku programskog sken ciklusa na sljedeći način:

- DN - *Timer done* bit se postavlja na 1 kada je  $ACC \geq PRE$  (tajmer je izmjerio zadano vrijeme). On se resetuje na 0 pomoću posebne RES naredbe.
- EN – *Timer enable* bit se postavlja na 1 kada je uslov u rangu istinit (rad tajmera je omogućen) i resetuje na 0 kada uslov postane neistinit (rad tajmera je onemogućen).
- TT - *Timer timing* bit se postavlja na 1 kada je uslov istinit i ako je  $ACC \leq PRE$  (tajmer radi). On se resetuje na 0 kada uslov postane neistinit ili kada se DN bit postavi na 1 (tajmer prestaje sa radom).



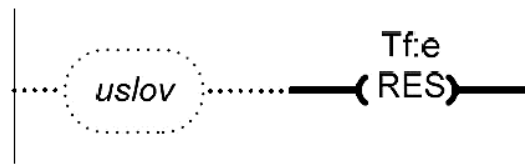
**Slika 11.** Primjena TONR tajmera



**Slika 12.** Vremenski dijagram TONR tajmera za primjer sa Slike 11

- TON sa zadržkom (TONR) postaje aktivan (broji) kada je „IN“ na log. „1“.
- Kada vrijednost brojanja postane veća ili jednaka od PT (*preset time*) bit koji odgovara tom tajmeru (T1) postavlja se na log. „1“.
- Kod ovog tipa tajmera ta vrijednost se zadržava čak i kada „IN“ postane neaktivan.
- TONR ima mogućnost da akumulira (broji) vrijeme bez obzira na promjene ulaza „IN“
- Reset funkcija služi da ga vrati na nulto stanje sa kog je i počelo prvobitno brojanje.

- **Reset naredba (RES)**



Slika 13. RES naredba

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

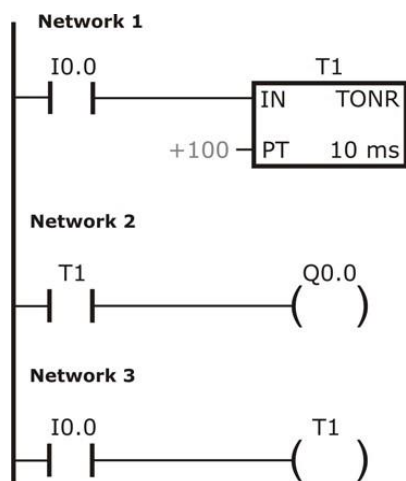
## Rezolucije tajmera

Tabela 1. Rezolucije tajmera

Tip tajmera	Rezolucija	Maksimalna vrijednost	Broj tajmera
TONR (sa pamćenjem)	1 ms	32.767 s (0.546 min)	T0, T64
	10 ms	327.67 s (5.46 min)	T1 do T4, T65 do T68
	100 ms	3276.7 s (54.6 min)	T5 do T31, T69 do T98
TON, TOF (bez pamćenja)	1 ms	32.767 s (0.546 min)	T32, T96
	10 ms	327.67 s (5.46 min)	T33 do T36, T9 do T100
	100 ms	3276.7 s (54.6 min)	T37 do T63, T101 do T255

## PRIMJERI KORIŠTENJA TAJMERA

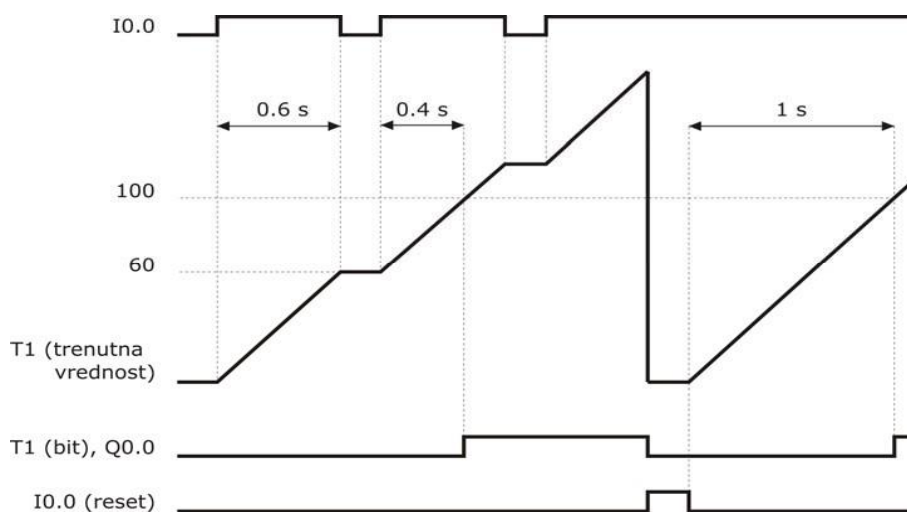
### PRIMJER 1



**Network 1**    // tajmer T1 sa rezolucijom  
                 // od 10 ms uključuje se  
                 // kada je  
                 //  $PT = 100 \times 10 \text{ ms} = 1 \text{ s}$

**Network 2**    // tajmer T1 kontroliše T1 bit  
                 // Q0.0 se uključuje nakon što  
                 // tajmer akumulira jednu  
                 // sekundu

**Network 3**    // kada je I0.0 uključen tajmer  
                 // T1 se resetuje



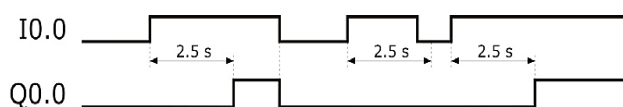
## PRIMJER 2

Napisati PLC program kojim će se obezbijediti kašnjenje ulaznog signala od 2.5 sekunde.

Za rješavanje ovog problema koristit će se jedna ulazna i jedna izlazna promjenljiva, i tajmer. U Tabeli 2, dati su nazivi i adrese korištenih promjenljivih.

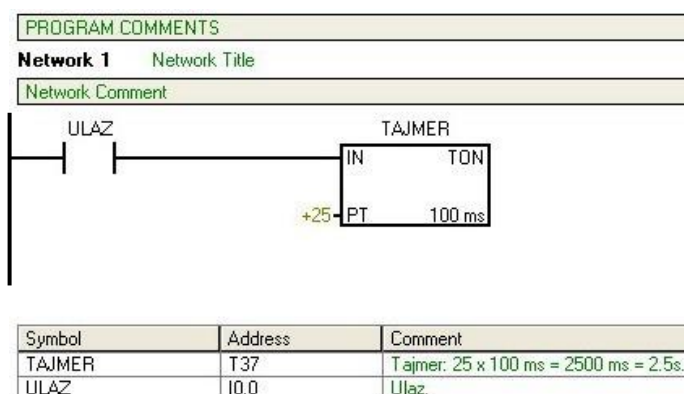
Tabela 2. Parametri za Primjer 2

PROMJENLJIVA	Naziv	Adresa	Komentar
ULAZNE	ULAZ	I0.0	Ulazni element (prekidač, taster, senzor...).
IZLAZNE	IZLAZ	Q0.0	Izlazni element (sijalica, motor, sirena...).
TAJMER	TAJMER	T37	Tajmer rezolucije 100ms (25 x 100ms = 2.5s).



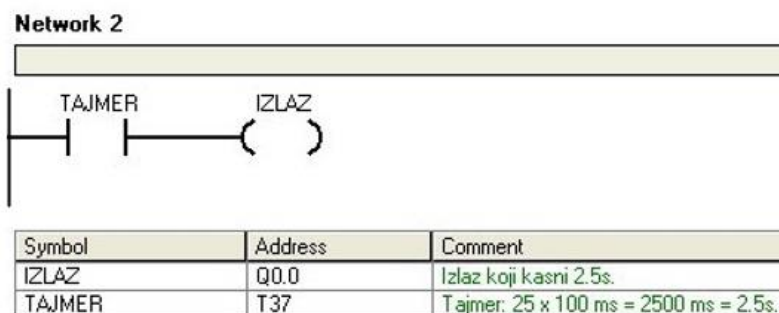
Slika 14. Vremenski dijagram za Primjer 2

Nakon uključivanja ulaza (*Network 1*) tajmer počinje da odbrojava inkremente vremena od 100ms. Kada tajmer dostigne vrijednost  $25 \times 100 \text{ ms} = 2.5 \text{ s}$ , uključuje se njegov izlaz TON.



Slika 15. Network 1 za Primjer 2

Izlaz tajmera u sljedećem koraku (*Network 2*) pali izlaz. Nakon gašenja ulaza, izlaz se odmah gasi.



Slika 16. Network 2 za Primjer 2

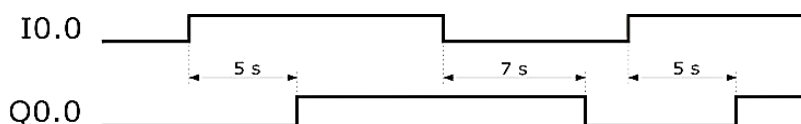
### PRIMJER 3

*Napisati LAD program kojim izlazni signal nastaje 5 sekundi posijle početka ulaznog signala i koji će djelovati još 7 sekundi po prestanku ulaznog signala.*

Za rješavanje ovog zadatka koristit će se jedan ulaz, jedan izlaz i dva tajmera. Jedan tajmer definisat će vrijeme kašnjenja izlaznog signala za ulaznim signalom, a drugi vrijeme zadržke izlaznog signala (vrijeme za koje će, nakon prestanka djelovanja ulaznog signala, izlazni signal postojati). U *Tabeli 3*, dati su nazivi i adrese korištenih promjenljivih.

*Tabela 3. Parametri za Primjer 3*

PROMJENLJIVA	Naziv	Adresa	Komentar
ULAZNE	ULAZ	I0.0	Ulazni signal (prekidač, taster, senzor...).
IZLAZNE	IZLAZ	Q0.0	Izlazni signal (sijalica, motor, sirena...).
TAJMER	T_START	T37	Startovanje kasni: $50 \times 100\text{ms} = 5\text{s}$ .
	T_STOP	T38	Zaustavljanje kasni: $70 \times 100\text{ms} = 7\text{s}$ .

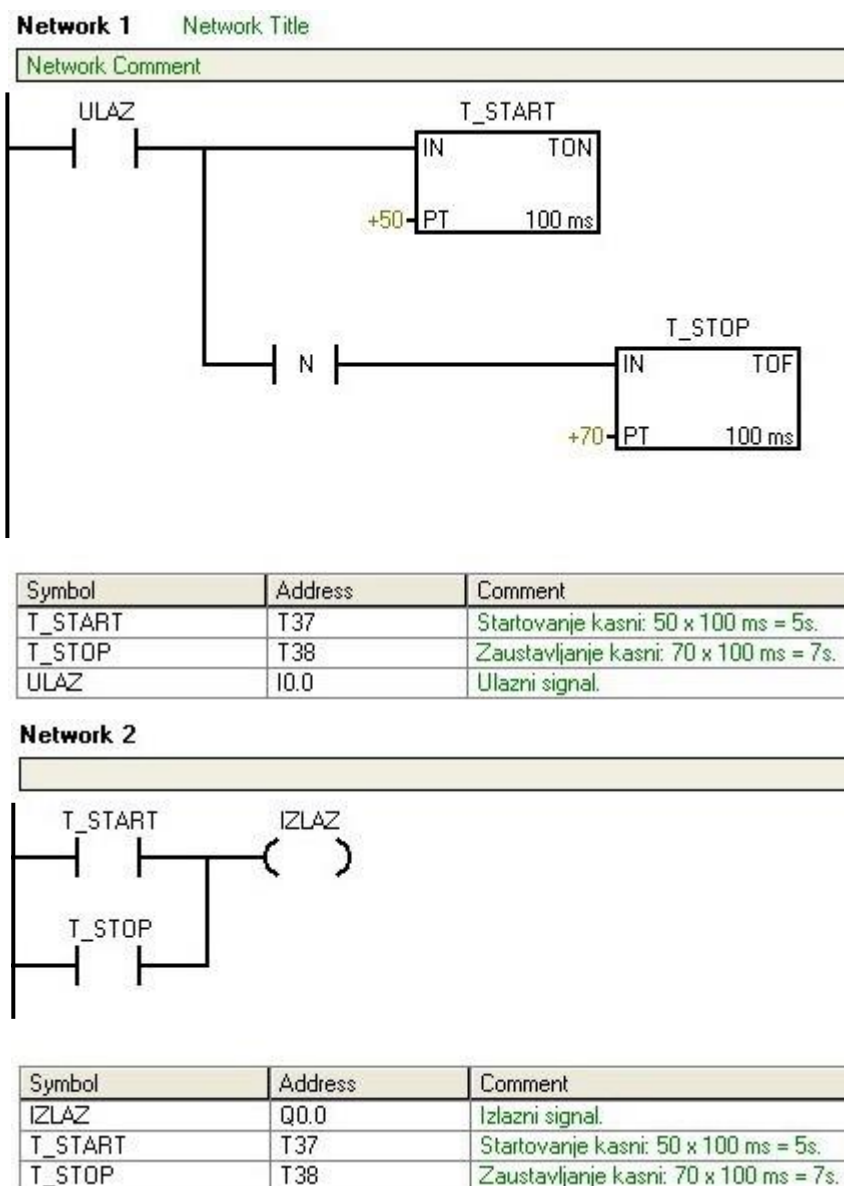


*Slika 17. Vremenski dijagram za Primjer 3*

Na pojavu ulaznog signala (*Network 1*), startuje se tajmer T\_START, tajmer sa kašnjenjem paljenja (TON), koji definiše vrijeme kašnjenja izlaznog signala za ulaznim. Nakon isteka predefinisano vremena ( $50 \times 100\text{ms} = 5\text{s}$ ), T\_START će na svom izlazu dati signal, koji će upaliti izlaz Q0.0 (*Network 2*).

U paralelnoj grani, nakon prestanka djelovanja ulaznog signala, na negativnu ivicu ulaznog signala (prelaz ON-OFF) startuje se tajmer sa kašnjenjem gašenja (TOF) T\_STOP, koji definiše vrijeme koje prođe od gašenja ulaznog signala do gašenja izlaznog signala. Nakon što istekne predefinisano vrijeme na ovom tajmeru ( $70 \times 100\text{ms} = 7\text{s}$ ), njegov izlaz će se ugasi. Kako T\_START nema signal (zbog gašenja ulaza, T\_START gasi svoj izlaz), i izlaz T\_STOP tajmera se ugasio, izlaz se gasi.





**Slika 18.** Network 1 i Network 2 za Primjer 3

Ukoliko bi ulazni signal nestao za kraće vrijeme od 5 sekundi nakon svog pojavljivanja, izlazni signal se ne bi pojavio.

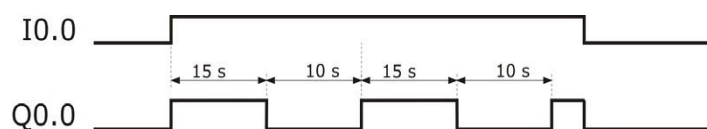
## PRIMJER 4

Napisati LAD program koji ponavlja izlazne signale u određenim vremenskim intervalima, kada se na ulaz dovodi signal.

Za rješavanje ovog zadatka koristit će se jedan ulaz, jedan izlaz i dva tajmera (Tabela 4). Jedan tajmer definisat će vrijeme za koje je izlazni signal uključen, a drugi, vrijeme za koje je izlaz isključen.

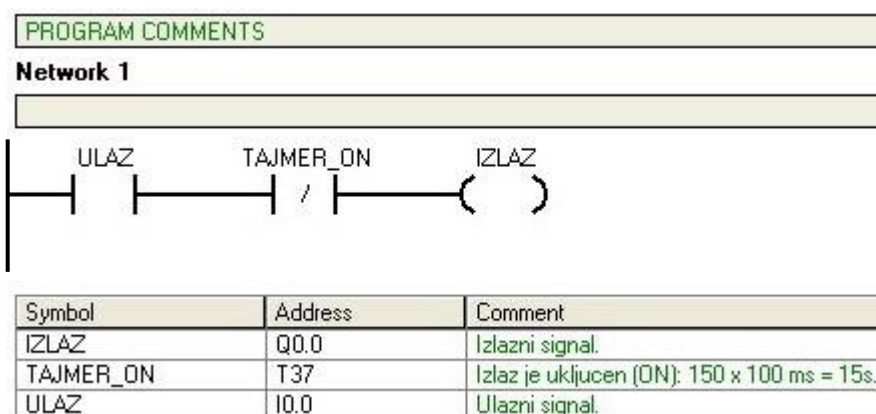
Tabela 4. Parametri za Primjer 4

PROMJENLJIVA	Naziv	Adresa	Komentar
ULAZNE	ULAZ	I0.0	Ulazni signal (prekidač, taster, senzor...).
IZLAZNE	IZLAZ	Q0.0	Izlazni signal (sijalica, motor, sirena...).
TAJMER	TAJMER_ON	T37	Izlaz je uključen (ON): 15 x 100ms = 15s.
	TAJMER_OFF	T38	Izlaz je isključen (OFF): 10 x 100ms = 10s.



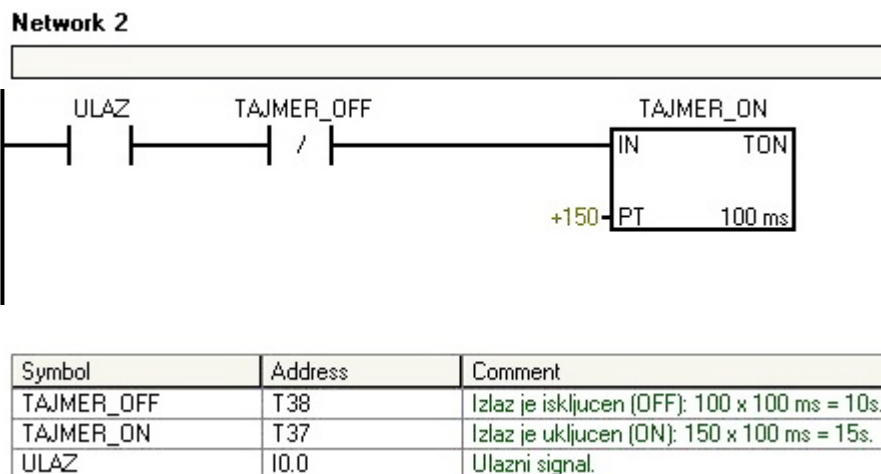
Slika 19. Vremenski dijagram za Primjer 4

Nakon uključenja ulaznog signala, stanje TAJMER\_ON tajmera je OFF, izlaz se pali (Network 1), stanje tajmera TAJMER\_OFF je također OFF, pa se startuje TAJMER\_ON (Network 2).



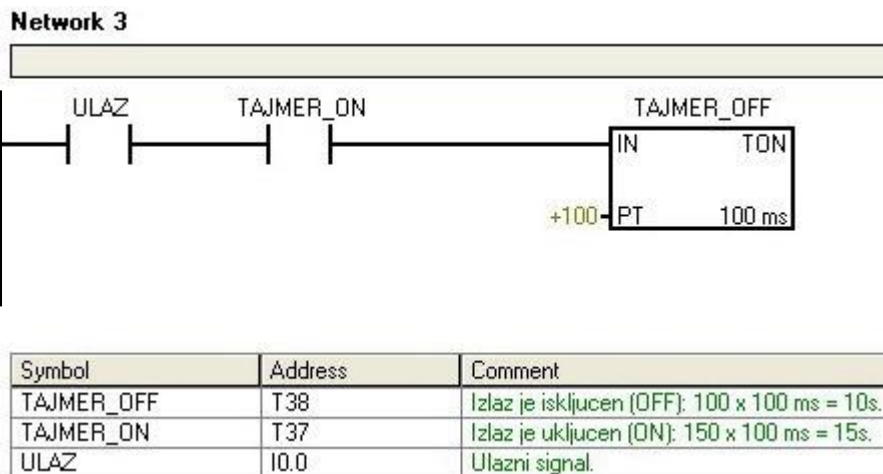
Slika 20. Network 1 za Primjer 4

Kada TAJMER\_ON, nakon 15 sekundi, uključi svoj izlaz, pali se i tajmer TAJMER\_OFF (*Network 3*), i odbrojava 10 sekundi. Izlaz se gasi jer je ulaz uključen, a TAJMER\_ON ima stanje logičke jedinice, pa normalno zatvoren kontakt prekida tok struje (*Network 1*).



*Slika 21. Network 2 za Primjer 4*

Kada tajmer TAJMER\_OFF, nakon deset sekundi, uključi svoj izlaz, TAJMER\_ON se resetuje, prelazi u stanje logičke nule (OFF) i počinje sa odbrojavanjem novih 15 sekundi (*Network 2*). Isključeni tajmer TAJMER\_ON ponovo pali izlaz (*Network 1*).



*Slika 22. Network 3 za Primjer 4*

Signal na izlazu gasi se onog trenutka kada nestane ulazni signal, bez obzira u kom stanju su tajmeri, ili izlaz.