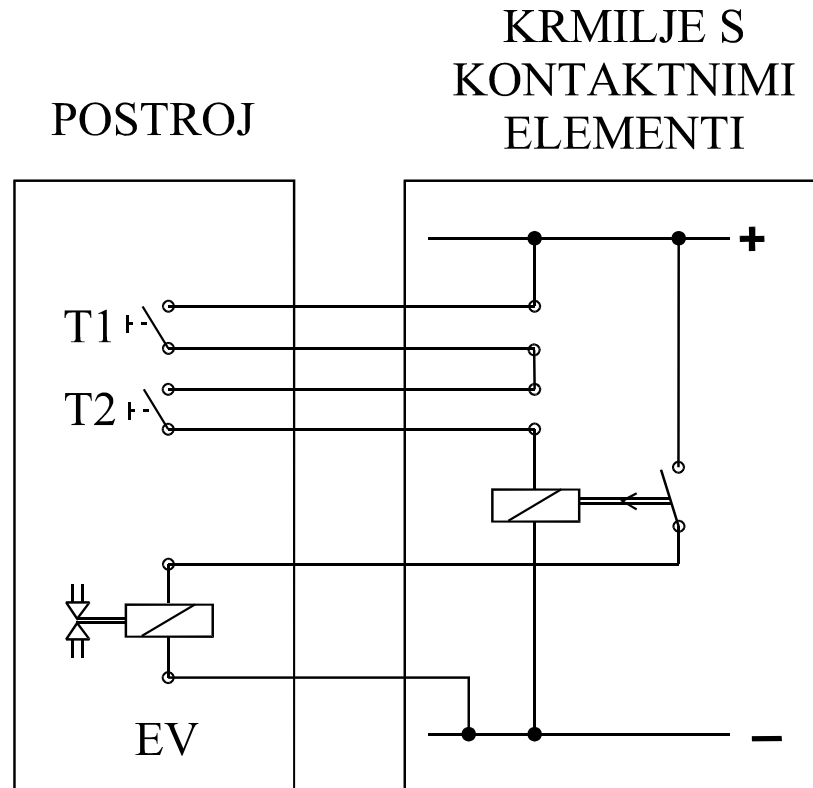


Slika 1.1: Električna shema enostavne krmilne naloge (a) in njena rešitev z releji (b)



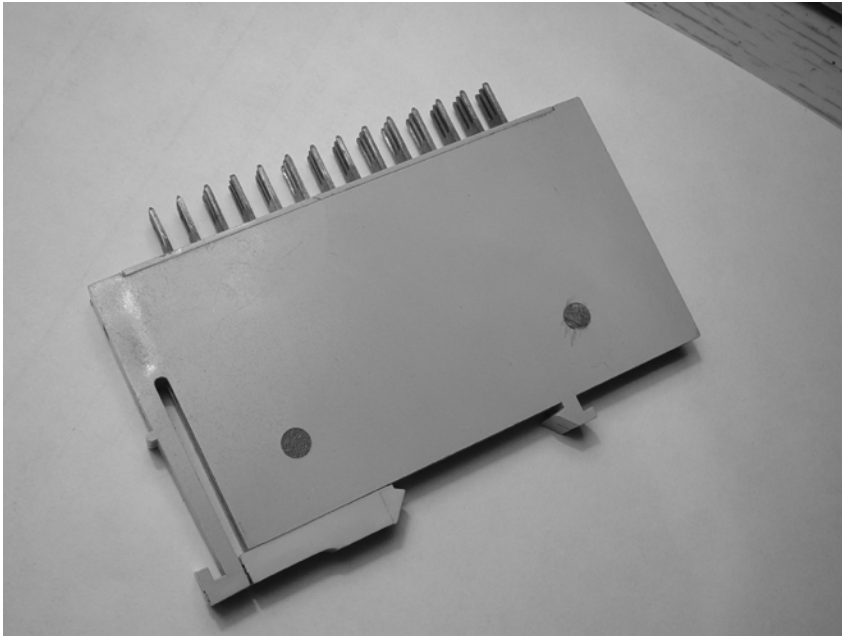
Relejni sistemi



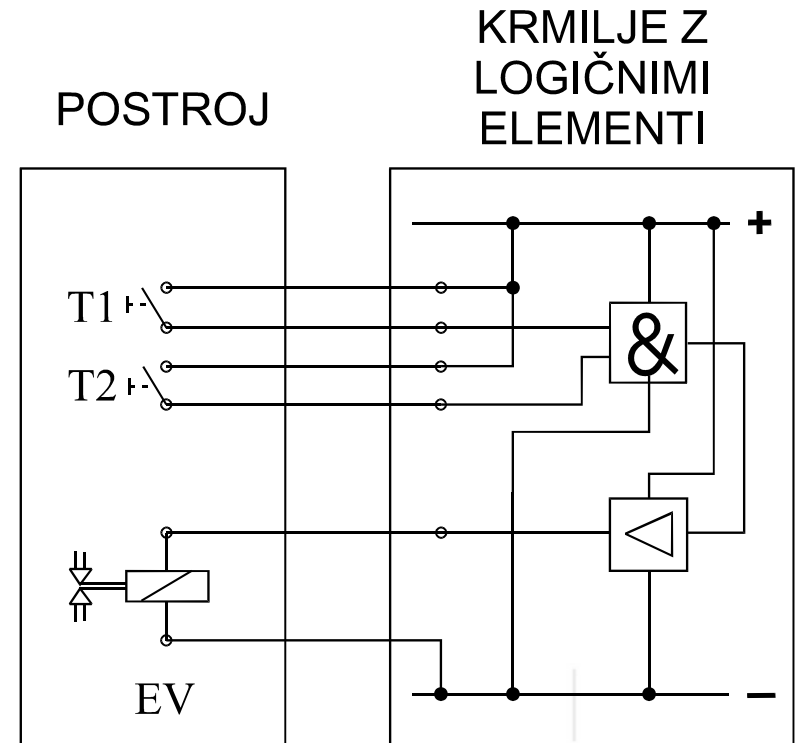
Slika 5.1: Primer kontaktnega krmilja z releji



Ožičeni krmilni sistemi



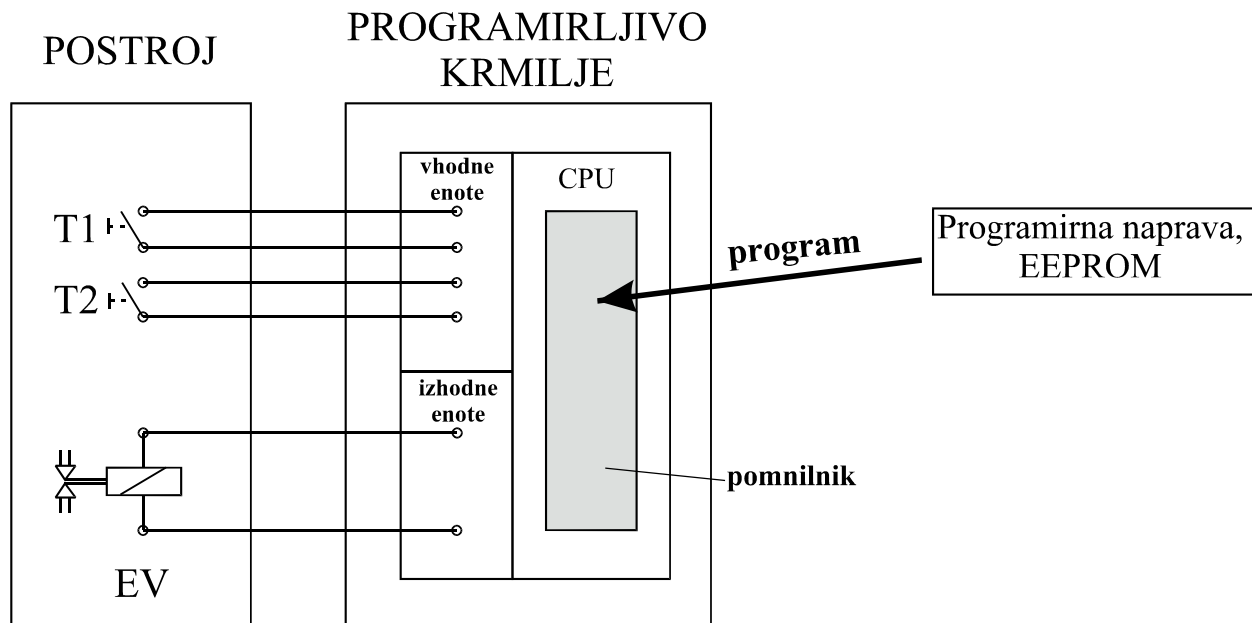
Slika 5.2: Elektronski logični elementi serije SIMATIC C



Slika 5.3: Primer krmilja z elektronskimi logičnimi elementi



Programirljivi krmilni sistemi

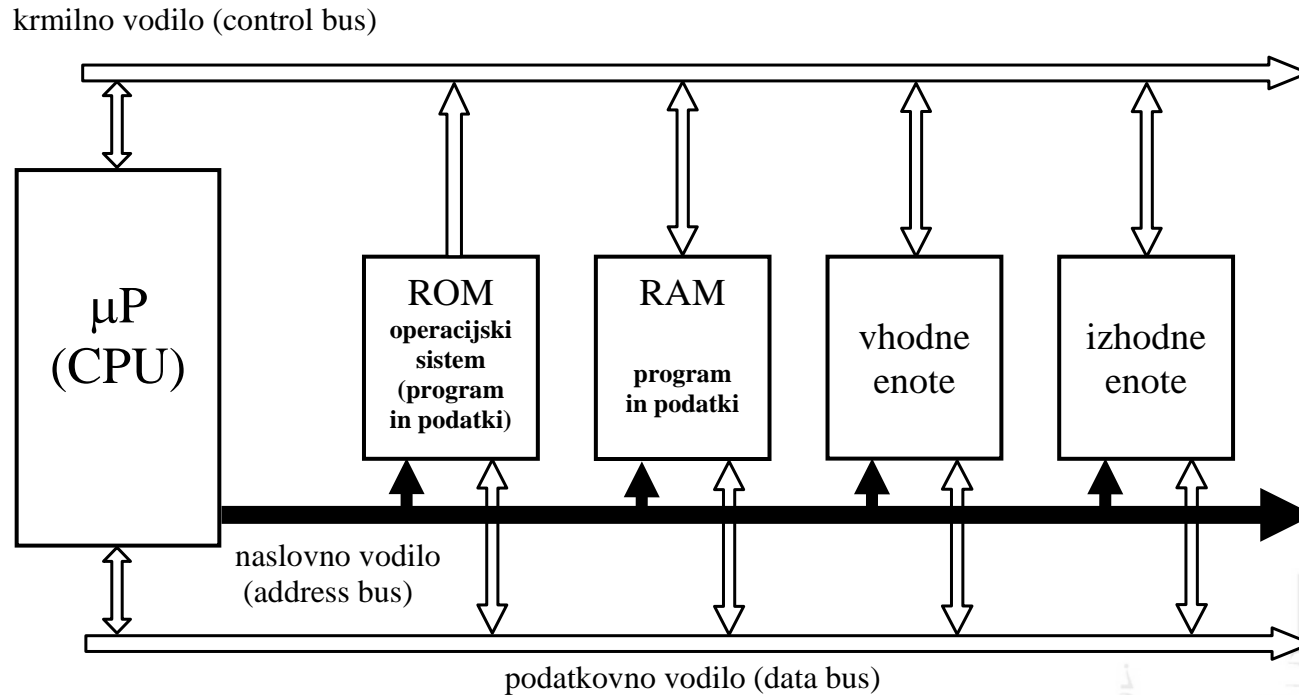


Slika 5.4: Programirljivo krmilje

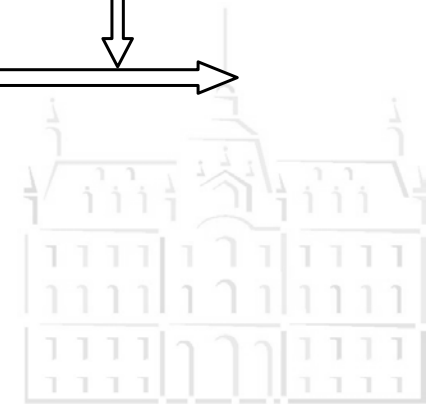




Programirljivi krmilni sistemi



Slika 5.5: Blokovna shema programirljivega krmilnika

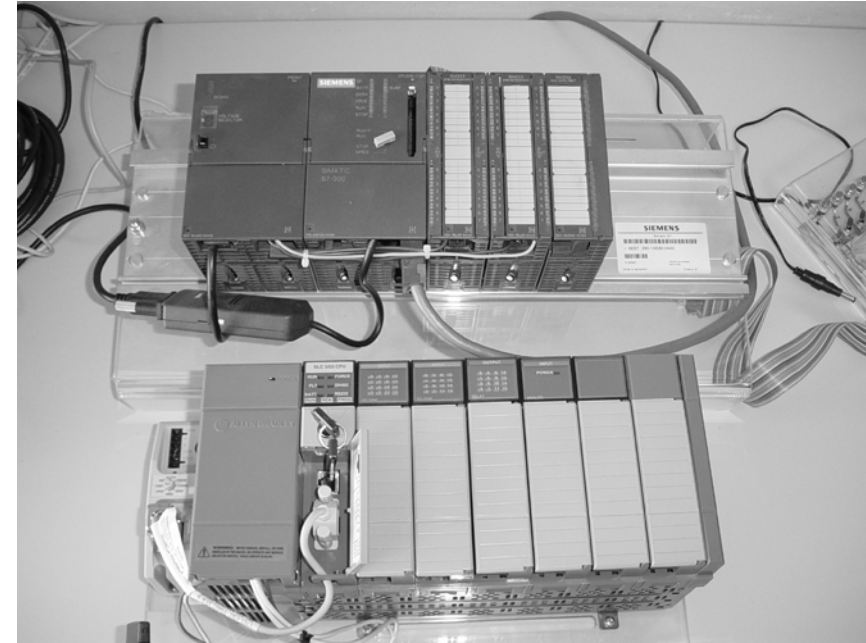




Programirljivi krmilni sistemi



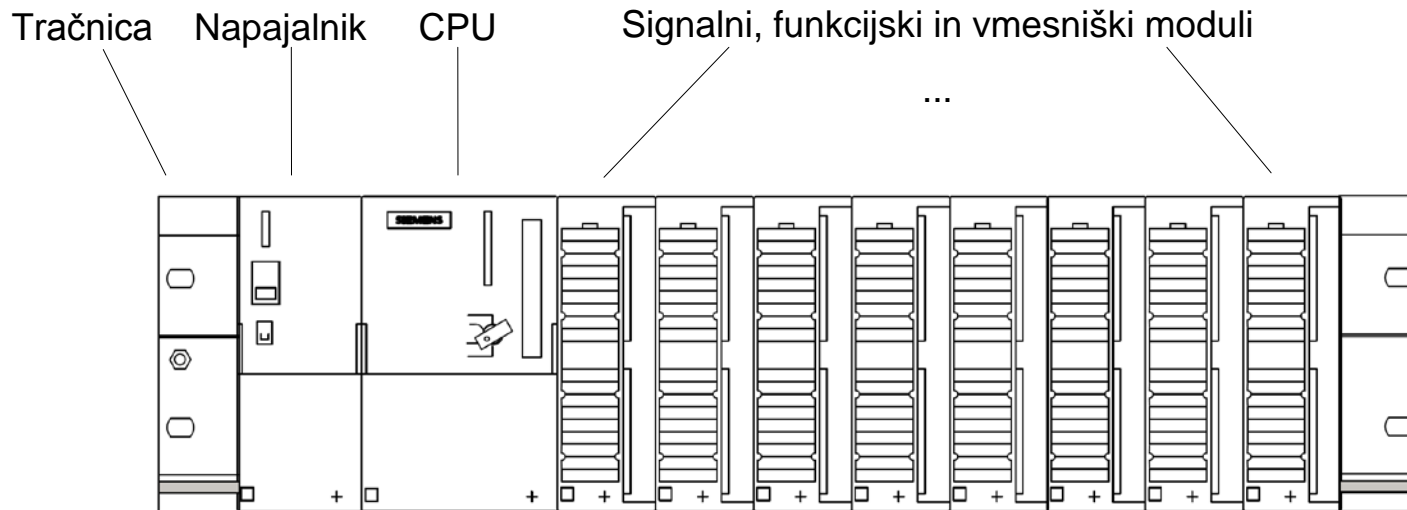
Slika 5.6: Primer
kompaktnega krmilnika
(MicroLogix, Allen Bradley)



Slika 5.7: Primer modularnega
krmilnika (Siemens in Allen
Bradley)



Zasnova krmilnika

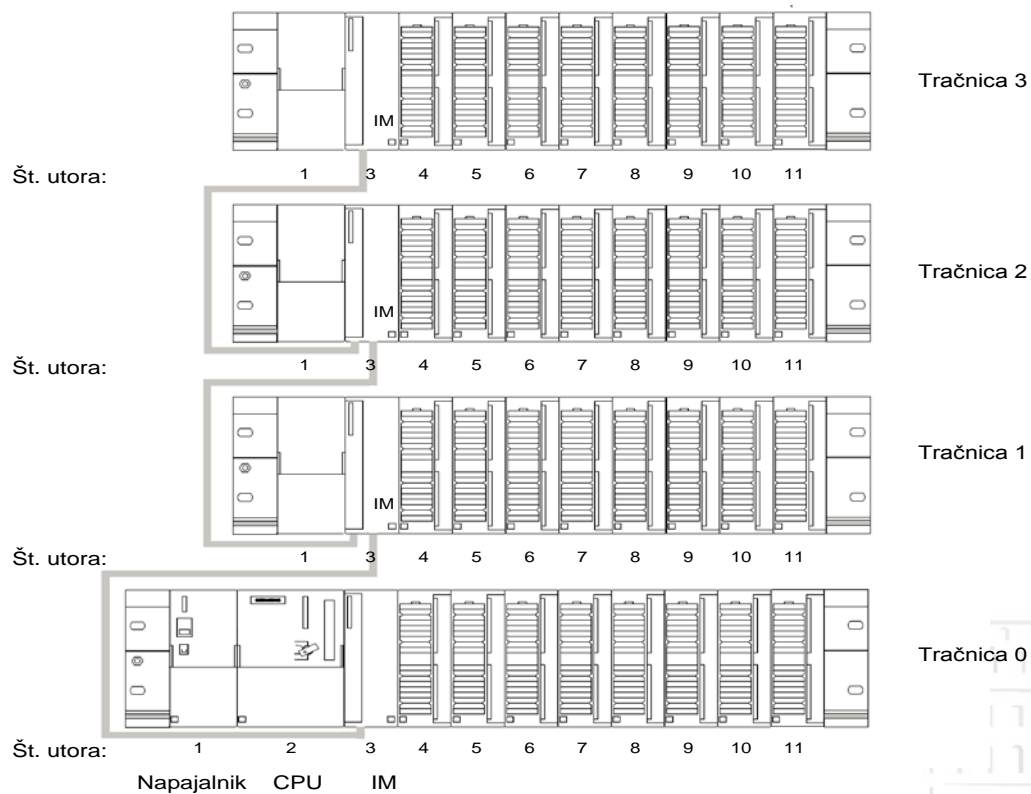


Slika 6.1: Fizična struktura krmilnika na eni tračnici





Zasnova krmilnika



Slika 6.2: Krmilnik z decentraliziranimi signalnimi in funkcijskimi moduli



Krmilnik S7-300

Tračnica	Vrsta modula	Številka utora										
		1	2	3	4	5	6	7	8	9	10	11
		Začetni naslovi modulov										
0	Digitalni	PS	CPU	IM	0	4	8	12	16	20	24	28
	Analogni				256	272	288	304	320	336	352	368
1	Digitalni			IM	32	36	40	44	48	52	56	60
	Analogni				384	400	416	432	448	464	480	496
2	Digitalni			IM	64	68	72	76	80	84	88	92
	Analogni				512	528	544	560	576	592	608	624
3	Digitalni			IM	96	100	104	108	112	116	120	124
	Analogni				640	656	672	688	704	720	736	752

Tabela 6.1: Privzeto naslavljanje modulov pri krmilnikih S7-300 glede na utore



Zasnova krmilnika

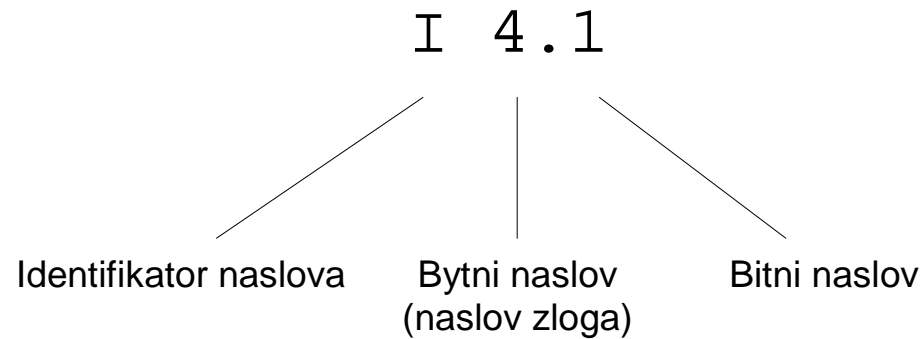
Podatkovni tip (podatkovna širina v bitih)	Pomen	Zgled za zapis konstant
BOOL (1)	Bit	TRUE FALSE
BYTE - zlog (8)	8-bitno šestnajstiško število	B#16#C3
CHAR - znak (8)	Znak (ASCII)	'A' 't'
WORD - beseda (16)	16-bitno šestnajstiško število 16-bitno dvojiško število število dogodkov v števcu (3 dekade BCD) dve 8-bitni nepredznačeni dec. števila	W#16#A30F 2#1010_0011_0000_1111 C#476 B#(163,15)
DWORD dvojna beseda (32)	32-bitno šestnajstiško število 32-bitno dvojiško število štiri 8-bitna nepredznačena dec. števila	DW#16#5F3C_0B20 2#0101_1111_0011_1100_0000_1011_0010_0000 B#(95,60,11,32)
INT (16)	Celo število	od -32768 do +32767
DINT (32)	Celo število	od -2 147 483 648 do +2 147 483 647
REAL (32)	Število s plavajočo vejico	-1.234567E-12
S5TIME (16)	Vrednost časa v SIMATIC formatu (predpona S5T# ali S5TIME#)	od S5T#0ms do S5TIME#2h46m30s
TIME (32)	Vrednost časa v IEC formatu (predpona T# ali TIME#)	od T#-24d20h31m23s647ms do TIME#24d20h31m23s647ms
DATE (16)	Datum (predpona D# ali DATE#)	od D#1990-01-01 do DATE#2168-12-31
TIME_OF_DAY (32)	Dnevni čas (predpona TOD# ali TIME_OF_DAY#)	od TOD#00:00:00 do TIME_OF_DAY#23:59:59.999

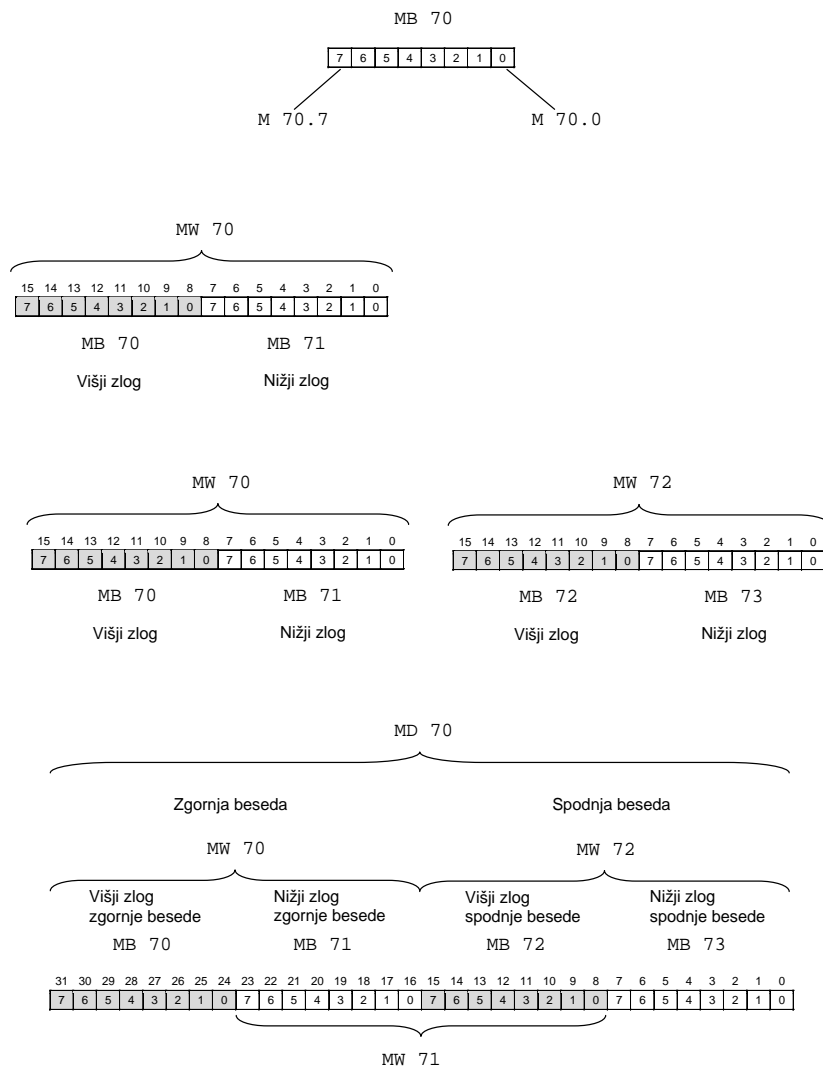
Tabela 6.2: Osnovni podatkovni tipi



Krmilnik S7-300

Naslovi krmilnikovih elementov



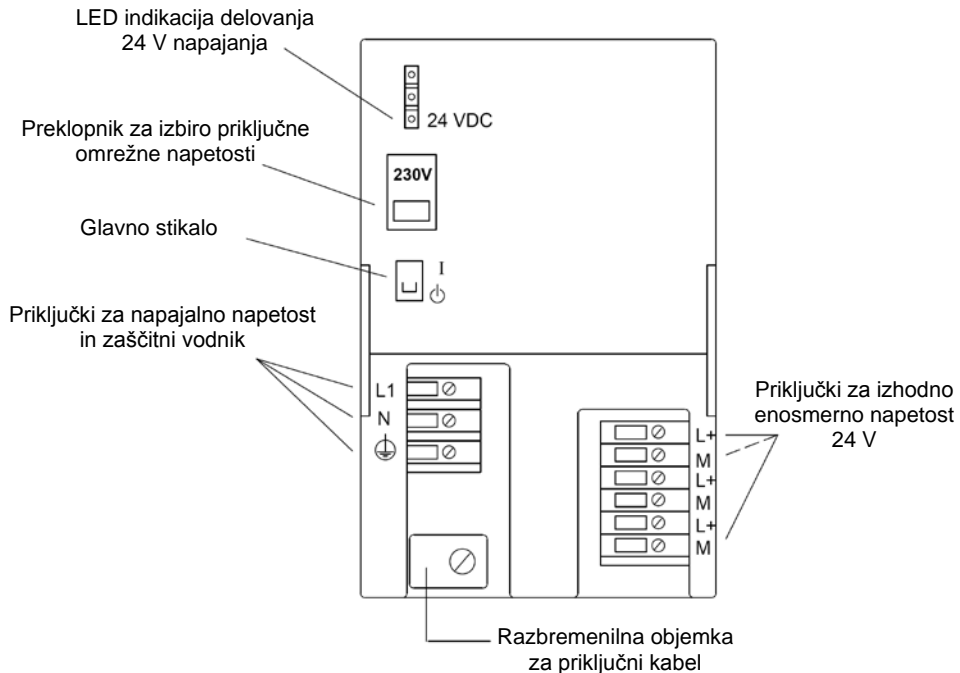


Slika 6.3: Organiziranost bitov v zloge, besede in dvojne besede

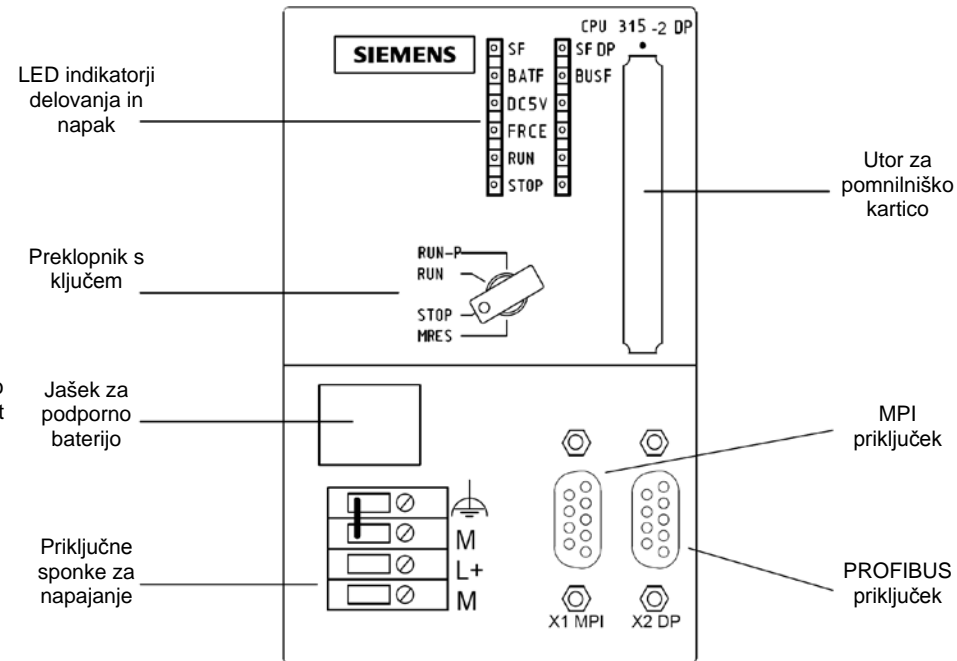




Osnovna konfiguracija v Laboratoriju za krmilne sisteme



Slika 6.4: Napajalnik PS 307 5 A

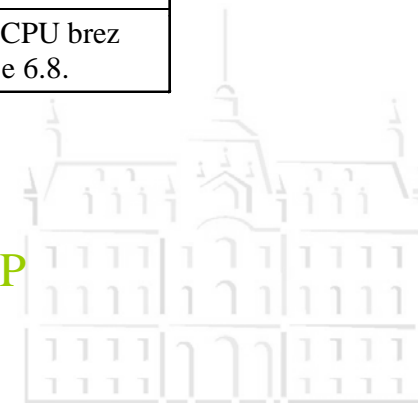


Slika 6.5: Centralna procesna enota CPU 315-2 DP



Položaj	Pomen	Opis
RUN-P	RUN-PROGRAM način	CPU obdeluje uporabniški program. Ključa NI mogoče izvleči iz preklopnika. Na programirni napravi lahko preberemo programe s CPU. Programe lahko naložimo s programirne naprave na CPU.
RUN	RUN način	CPU obdeluje uporabniški program. Ključ je mogoče izvleči iz preklopnika. Na programirni napravi lahko preberemo programe s CPU. Programov NE moremo naložiti s programirne naprave na CPU.
STOP	STOP način	CPU NE obdeluje uporabniškega programa. Ključ je mogoče izvleči iz preklopnika. Na programirni napravi lahko preberemo programe s CPU. Programe lahko naložimo s programirne naprave na CPU.
MRES	resetiranje CPU	Trenutni položaj preklopnika; služi za resetiranje CPU brez programirne naprave. Za podrobnosti glej poglavje 6.8.

Tabela 6.5: Vloga preklopnika na CPU 315-2 DP





LED	Barva	Pomen	Opis
SF	rdeča	System error/fault (sistemska napaka)	Sveti , če se pojavi: <ul style="list-style-type: none">– hardverska napaka– napaka operacijskega sistema (firmware)– programska napaka– napaka pri parametriranju– aritmetična napaka– napaka pri časovnikih– okvara pomnilniške kartice– okvara podporne baterije– vhodno-izhodna napaka Napako lahko natančno razberemo s pomočjo diagnostičnih orodij na programirni napravi.
BATF	rdeča	Battery fault (okvara podporne baterije)	Sveti , če je podpora baterija: <ul style="list-style-type: none">– pokvarjena– prazna– ni vstavljena
DC5V	zelena	enosmerna napetost 5 V za CPU in S7-300 vodilo	Sveti , če je notranje napajanje 5 V brezhibno.
FRCE	rumena	FORCE	Sveti , kadar smo s programirne naprave sprožili prisilno postavljanje elementov v zelena stanja mimo zahtev krmilnega programa.
RUN	zelena	RUN način delovanja	Sveti , če se uporabniški program obdeluje. Utripa (2 Hz) med zagonom CPU (restart).
STOP	rumena	STOP način delovanja	Sveti , če se uporabniški program NE obdeluje. Utripa med resetiranjem CPU.

Tabela 6.6: Pomen LED indikatorjev na CPU 315-2 DP



Krmilnik S7-300

Pomen	Opis
Delovni pomnilnik (vgrajeni)	48 kB
Nalagalni pomnilnik (vgrajeni) Nalagalni pomnilnik (razširljivi)	80 kB RAM do 512 kB FEPRAM (različne pomnilniške kartice)
Hitrost	približno 0,3 ms za 1000 binarnih ukazov
Pomožni pomnilniki (M-področje) nastavljiva remanentnost: privzeta remanentnost:	2048 bitov MB 0 do MB 255 MB 0 do MB 15 (16 zlogov)
Števci (C) nastavljiva remanentnost: privzeta remanentnost:	64 C 0 do C 63 C 0 do C 7 (8 števecov)
Časovniki (T) nastavljiva remanentnost: privzeta remanentnost:	128 T 0 do T 127 ni remanentnih časovnikov
Remanentno področje podatkovnih blokov	8 podatkovnih blokov (DB), skupaj največ 4096 remanentnih podatkovnih zlogov
Skupno največje število remanentnih zlogov	4736
Globina gnezdenja	8 nivojev po prednostnem razredu, 4 dodatni nivoji znotraj OB za sinhronsko napako
Največje število digitalnih vhodov	1024
Največje število digitalnih izhodov	1024
Največje število analognih vhodov	128
Največje število analognih izhodov	128

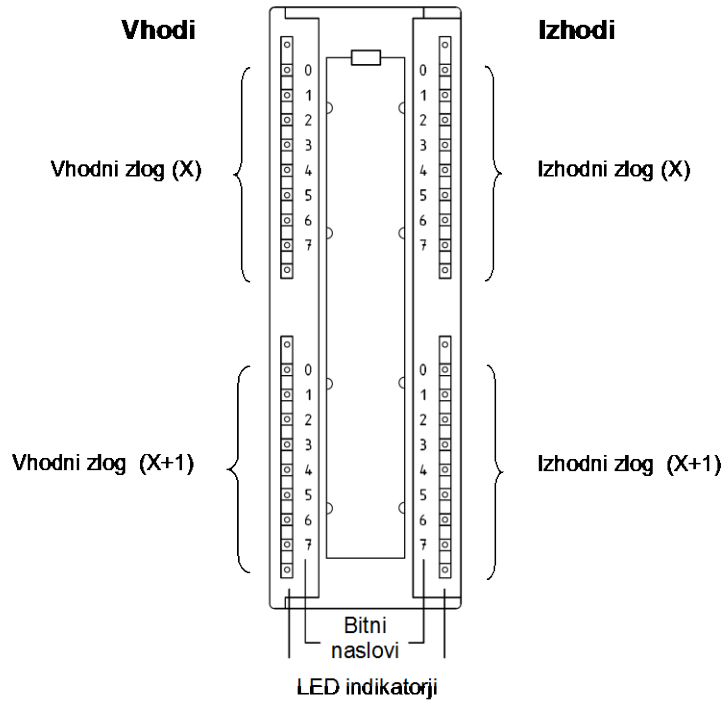
Tabela 6.7: Najpomembnejše specifikacije CPU 315-2 DP (1)



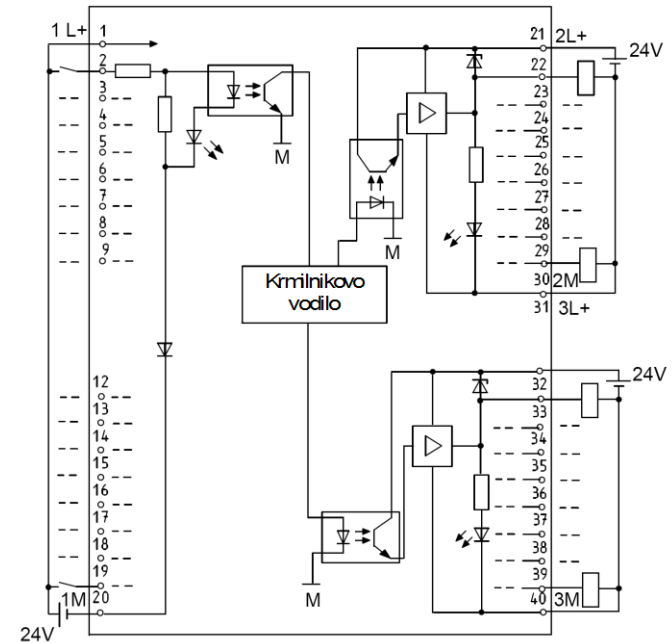
Krmilnik S7-300

Pomen	Opis
Procesne preslikave vhodov izhodov	0 do 127 I 0.0 do I 127.7 Q 0.0 do Q 127.7
DP naslovno področje	2 kB s funkcijo SFC 14 ("DPRD_DAT") ali SFC 15 ("DPWR_DAT"), od tega 512 bytov za ukaze nalaganja in prenosa
Število blokov OB (organizacijski bloki) FB (funkcijski bloki) FC (funkcije) DB (podatkovni bloki) SFC (sistemske funkcije) SFB (sistemski funkcijski bloki)	14 128 128 127 53 7
Časovnik realnega časa	hardversko izveden
Števec delovnih ur obseg selektivnost remanentnost	1 od 0 do 32767 ur 1 ura da
MPI (MultiPoint Interface) povezave zagotovljene povezave s programirno napravo (PG) zagotovljene povezave z operacijskim panelom (OP) nedoločene povezave za PG/OP/prog. komunikacijo zagotovljene povezave za programsko komunikacijo največje število vozlišč (nodes) hitrost komunikacije razdalja brez obnavljalnikov (repeaterjev) razdalja z dvema obnavljalnikoma razdalja z 10 zaporednimi obnavljalniki	1 1 2 8 32 187,5 Kbaud 50 m 1100 m 9100 m

Tabela 6.7: Najpomembnejše specifikacije CPU 315-2 DP (2)



Slika 6.6: Digitalni vhodno-izhodni modul SM 323 DI16/DO16 24 VDC/0.5 A



Slika 6.7: Priključitvena shema DIO modula SM 323 DI16/DO16 24 VDC/0.5 A



Krmilnik S7-300

Dimenzije in masa	
Dimenzije (širina x višina x globina)	40 x 125 x 120 mm
Masa	260 g
Specifični podatki o modulu	
Število vhodov	16
Število izhodov	16
Dolžina kabla:	
• neoklopljeni	največ 600 m
• oklopljeni	največ 1000 m
Napetosti, toki, potenciali	
Nazivna bremenska napetost L+	24 V enosmerna
• zaščita pred naspr. polariteto	DA
Število sočasno nadzorovanih vhodov	
• vodoravna namestitvev	
do 40 °C	16
do 60 °C	8
• navpična namestitvev	
do 40 °C	16

Stanja, prekinitve, diagnostika	
Prikaz stanj	zelena LED za vsak kanal
Prekinitve	NE
Diagnostične funkcije	NE
Podatki za izbiro senzorjev	
Vhodna napetost	
• nazivna vrednost	24 V enosmerna
• logična "1"	od 13 do 30 V
• logična "0"	od -3 do 5 V
Vhodni tok	
• pri logični "1"	tipično 7 mA
Zakasnilni čas vhoda	
• z logične "0" na "1"	od 1,2 do 4,8 ms
• z logične "1" na "0"	od 1,2 do 4,8 ms
Vhodna karakteristika:	po IEC 1131, tip 1
Priključitev 2-žičnega BERO	možna
• dopustna predpolarizacija največ	1,5 mA
Podatki za izbiro aktuatorjev	

Tabela 6.8: Najpomembnejše specifikacije DIO modula SM 323 DI16/DO16 24 VDC/0.5 A (1)



Krmilnik S7-300

<ul style="list-style-type: none">navpična namestitvev do 40 °C	16
Skupni tok izhodov (po skupini)	
<ul style="list-style-type: none">vodoravna namestitvev do 20 °C	največ 4 A
do 40 °C	največ 3 A
do 60 °C	največ 2 A
<ul style="list-style-type: none">navpična namestitvev do 40 °C	največ 2 A
Galvanska ločitev	
<ul style="list-style-type: none">med kanali in krmilnikovim vodilom	DA
<ul style="list-style-type: none">med kanali	DA
vhodi v skupinah po 16	
izhodi v skupinah po 8	
Dopustne potencialne razlike	
<ul style="list-style-type: none">med različnimi tokokrogi	75 V enosmerna 60 V izmenična
Izolacija preizkušena pri:	600 V enosmerna
Tokovna obremenitev	
<ul style="list-style-type: none">iz krmilnikovega vodila	največ 55 mA
<ul style="list-style-type: none">iz bremenske napetosti L+ (brez bremena)	največ 100 mA
Izgube na modulu	tipično 6,5 W

Podatki za izbiro aktuatorjev	
Izhodna napetost	
<ul style="list-style-type: none">pri logični "1"	najmanj L+ (-0,5 V)
Izhodni tok	
<ul style="list-style-type: none">pri logični "1"	nazivna vrednost 0,5 A dopusten obseg od 5 mA do 0,6 A
<ul style="list-style-type: none">pri logični "0"	preostali tok največ 0,5 mA
Bremenska impedanca	od 48 Ω do 4 kΩ
Obremenitev s svetilko	največ 5 W
Paralelna povezava dveh izhodov	
<ul style="list-style-type: none">za redundantno proženje bremena	možna (samo za izhode iz iste skupine)
<ul style="list-style-type: none">za povečanje moči	ni možna
Proženje digitalnega vhoda	možno
Preklopna frekvenca	
<ul style="list-style-type: none">ohmsko breme	največ 100 Hz
<ul style="list-style-type: none">induktivno breme po IEC 947-5-1, DC 13	največ 0,5 Hz
<ul style="list-style-type: none">obremenitev s svetilko	največ 10 Hz

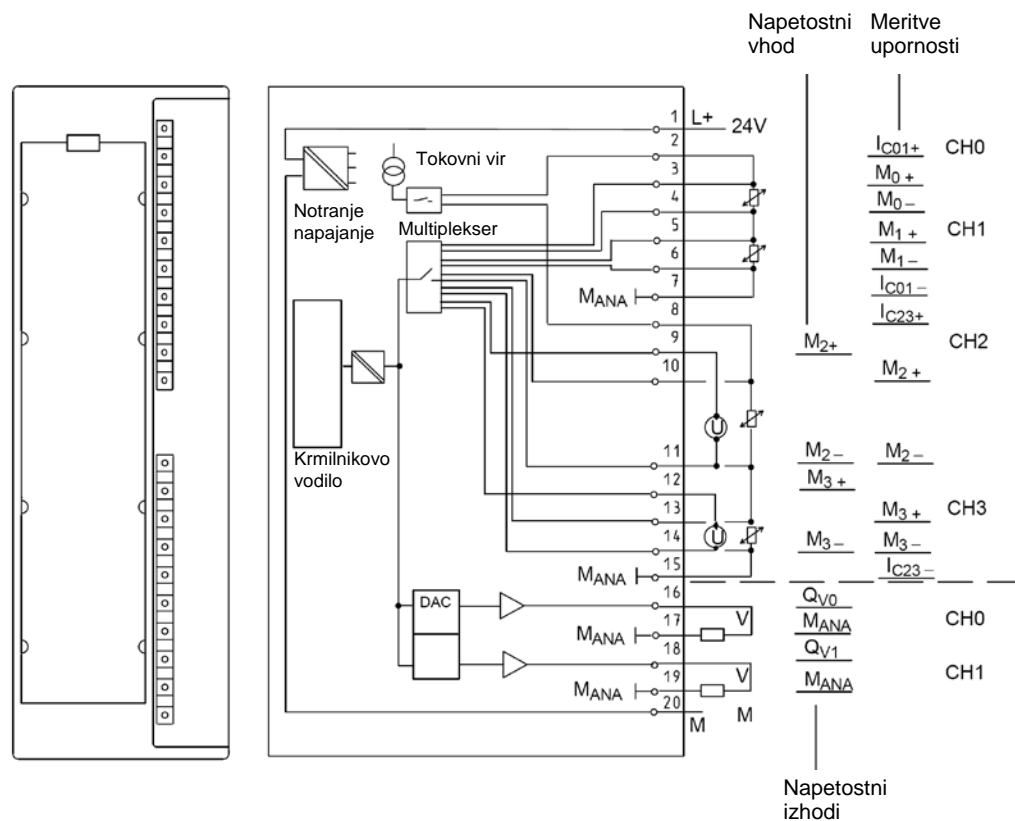
Tabela 6.8: Najpomembnejše specifikacije DIO modula SM 323 DI16/DO16 24 VDC/0.5 A (2)



Vhodna veličina	Obseg	Opis
U: vhodna napetost	0 - 10 V	10 V = 27648 (DEC) = 6C00 (HEX)
R-4L: upornost (4-žična priključitev)	0-10 kΩ	10 kΩ = 27648 (DEC) = 6C00 (HEX)
RTD-4L: temperatura (linearna upornost, 4-žična priključitev)	Pt 100	130°C = 13000 (DEC) = 32C8 (HEX)
Izhodna veličina		
U: izhodna napetost	0 - 10 V	10 V = 27648 (DEC) = 6C00 (HEX)

Tabela 6.9: Vhodni in izhodni obseg AIO modula SM 334 pri različnih nastavitvah





Slika 6.8: Priključitvena shema AIO modula SM 334; AI 4/AO 2X12 bits



Dimenzije in masa	
Dimenzije (širina x višina x globina)	40 x 125 x 117 mm
Masa	pribl. 200 g
Specifični podatki o modulu	
Število vhodov	4
• za uporovni tip senzorja	4
Število izhodov	2
Dolžina oklopljenega kabla:	največ 100 m
Napetosti, toki, potenciali	
Nazivna bremenska napetost L+	24 V enosmerna
• zaščita pred naspr. polariteto	DA
Nazivna napetost za napajanje elektronike	24 V enosmerna
Napajanje izhodov	
• zaščita pred kratkim stikom	DA

Stanja, prekinitve, diagnostika	
Prekinitve	NE
Diagnostične funkcije	NE
Podatki za izbiro senzorjev	
Vhodni obseg (nazivne vrednosti) / vhodna upornost	
• napetost	0 – 10 V 100 kΩ
• upornost	10 kΩ 10 mΩ
• temperatura	PT 100 10 mΩ
Največja dopustna napetost na vhodih (meja uničenja)	največ 20 V trajno 75 V največ 1 s, cikel 1:20
Priklučitev senzorjev	
• za meritev napetosti	možna
• za meritev upornosti:	
dvožična vezava	možna
trižična vezava	možna
štirižična vezava	možna

Tabela 6.10: Najpomembnejše specifikacije AIO modula SM 334; AI 4/AO 2X12 bits (1)



Krmilnik S7-300

Zaščita pred kratkim stikom	
Konstantni merilni tok za uporovni tip senzorja:	
• za PT 100	tipično 490 μ A
• pri 10 k Ω	pri 105 μ A
Galvanska ločitev	
• med kanali in krmilnikovim vodilom	DA
• med kanali in napajanjem elektronike	DA
• med posameznimi kanali	NE
Dopustne potencialne razlike	
• med vhodi in M_{ANA} (U_{CM})	1 V
• med vhodi (ECM)	1 V
• med M_{ANA} in $M_{internal-}$ (U_{ISO})	75 V enosmerna 60 V izmenična
Izolacija preizkušena pri:	500 V enosmerna
Tokovna obremenitev	
• iz krmilnikovega vodila	največ 60 mA
• iz napajalnika in iz bremenske napetosti L+ (brez bremena)	največ 80 mA
Izgube na modulu	tipično 2 W

Linearizacija karakteristike	s parametriranjem
• za RTD	PT 100
Uporabniški podatki (ENG):	v stopinjah Celzija
Podatki za izbiro aktuatorjev	
Izhodni obseg (nazivna vrednost)	
• napetost	0 – 10 V
Impedanca (v nazivnem izhodnem obsegu)	
• za napetostne izhode	najmanj 2,5 k Ω
– kapacitivno breme	največ 1,0 μ F
Napetostni izhodi	
• zaščita pred kratkim stikom	DA
• kratkostični tok	največ 10 mA
Meja uničenja za zunanje napetosti ali toke	
• napetost na izhodu proti M_{ANA}	največ 15 V trajno
Priključitev aktuatorjev	
• na napetostni izhod	
dvožična vezava	možna
štirižična vezava (za merilno vezje)	NI možna

Tabela 6.10: Najpomembnejše specifikacije AIO modula SM 334; AI 4/AO 2X12 bits (21)



Zajemanje analognih vrednosti z vhodov

Merilni princip	integracijski	
Čas integriranja/pretvorbe (po kanalu)		
• možnost parametriranja	DA	
• čas integriranja	16 ² / ₃ ms	20 ms
• čas pretvorbe (vključno s časom integriranja)	72 ms	85 ms
• dodatni čas pretvorbe pri meritvi upornosti	72 ms	85 ms
• ločljivost (resolucija) (vključno s presegom območja)	12 bitov	12 bitov
• preprečevanje motenj motilne napetosti s frekvenco	60 Hz	50 Hz
Glajenje merjenih vrednosti	programirljivo, v dveh stopnjah	
Časovna konstanta vhodnega filtra	0,9 ms	
Osnovni odzivni čas modula (če so omogočeni vsi kanali)	350 ms	

Preprečevanje motenj in meje pogreška vhodov

Dušenje šuma za $f = n \times (f_1 \pm 1 \%)$ ($f_1 =$ motilna frekvenca)		
• sofazne motnje ($U_{pp} < 1 \text{ V}$)		> 38 dB
• protifazne motnje (temenska vrednost motnje je manjša od nazivne vrednosti vhodnega obsega)		> 36 dB
Presluh med vhodi		> 88 dB
Pogrešek (v celotnem temperaturnem območju, glede na vhodni obseg)		
• meritev napetosti	od 0 do 10 V	$\pm 0,7 \%$
• meritev upornosti	10 k Ω	$\pm 3,5 \%$
• meritev temperature	PT 100	$\pm 1 \%$
Osnovni pogrešek (pri 25 °C, glede na vhodni obseg)		
• meritev napetosti	od 0 do 10 V	$\pm 0,5 \%$
• meritev upornosti	10 k Ω	$\pm 2,8 \%$
• meritev temperature	PT 100	$\pm 0,8 \%$

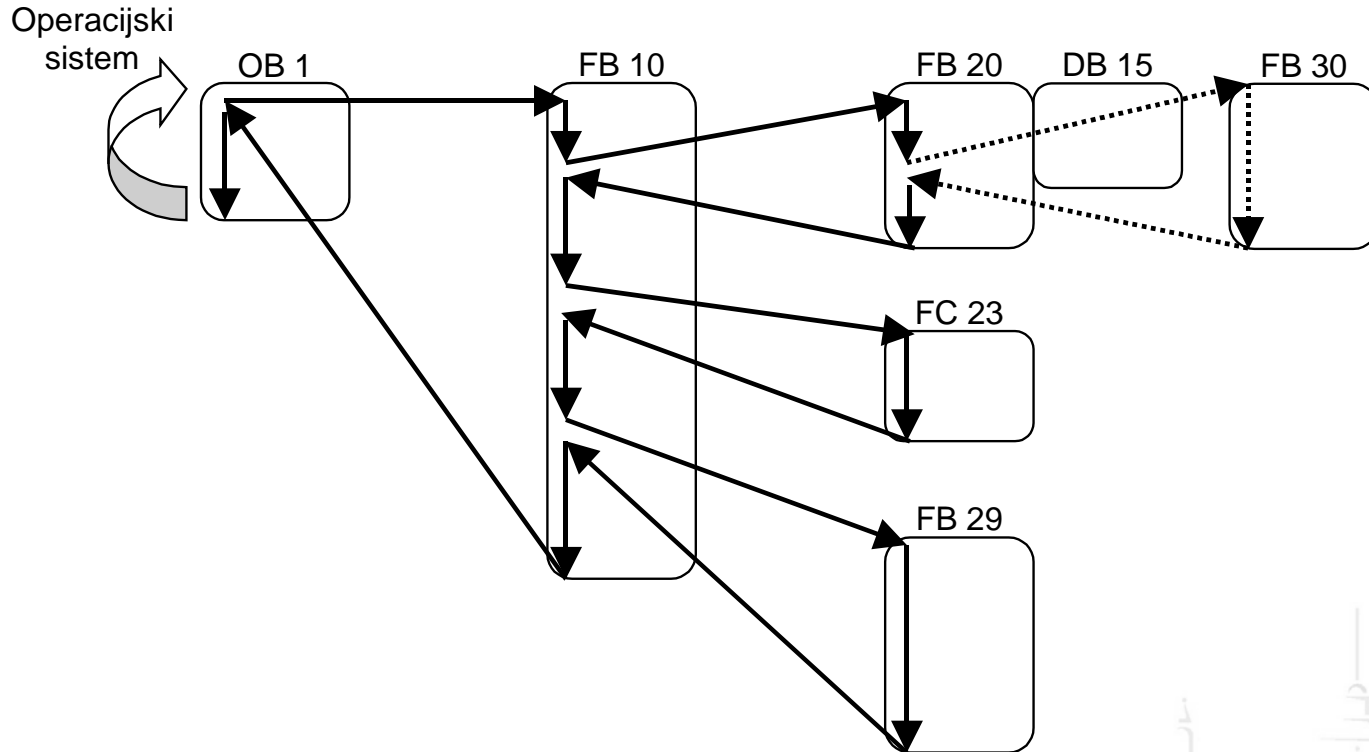
Tabela 6.10: Najpomembnejše specifikacije AIO modula SM 334; AI 4/AO 2X12 bits (3)



Generiranje analognih vrednosti na izhodih	
Ločljivost (rezolucija) (vključno s presegom območja)	12 bitov
Čas pretvorbe (po kanalu)	500 μ s
Čas ustalitve	
• za ohmska bremena	največ 0,8 ms
• za kapacitivna bremena	največ 0,8 ms

Temperaturni pogrešek (glede na vhodni obseg)	$\pm 0,01$ % / K
Nelinearnost (glede na vhodni obseg)	$\pm 0,05$ %
Ponovljivost meritev (v ustaljenem stanju pri 25 °C, glede na vhodni obseg)	$\pm 0,05$ %
Preprečevanje motenj in meje pogreška izhodov	
Presluh med izhodi	> 88 dB
Pogrešek (v celotnem temperaturnem območju, glede na izhodni obseg)	
• napetostni izhodi	$\pm 1,0$ %
Osnovni pogrešek (pri 25 °C, glede na izhodni obseg)	
• napetostni izhodi	$\pm 0,85$ %
Temperaturni pogrešek (glede na izhodni obseg)	$\pm 0,01$ % / K
Nelinearnost (glede na izhodni obseg)	$\pm 0,01$ %
Ponovljivost meritev (v ustaljenem stanju pri 25 °C, glede na izhodni obseg)	$\pm 0,01$ %
Izhodna valovitost; pasovna širina od 0 do 50 kHz (glede na vhodni obseg)	$\pm 0,1$ %

Tabela 6.10: Najpomembnejše specifikacije AIO modula SM 334; AI 4/AO 2X12 bits (4)



Slika 6.9: Način izvajanja strukturiranega uporabniškega programa



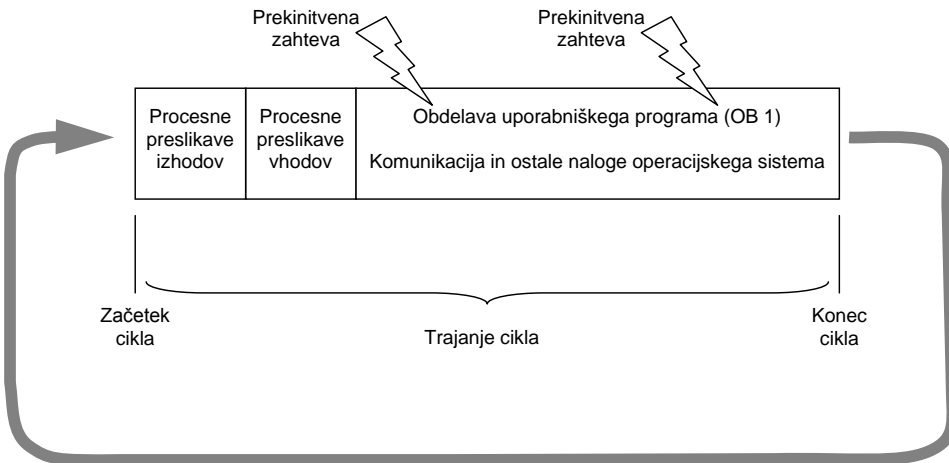
```
M003:      L      IW 2      // naloži v AKU vhodno besedo
```

Diagram illustrating the components of the instruction line:

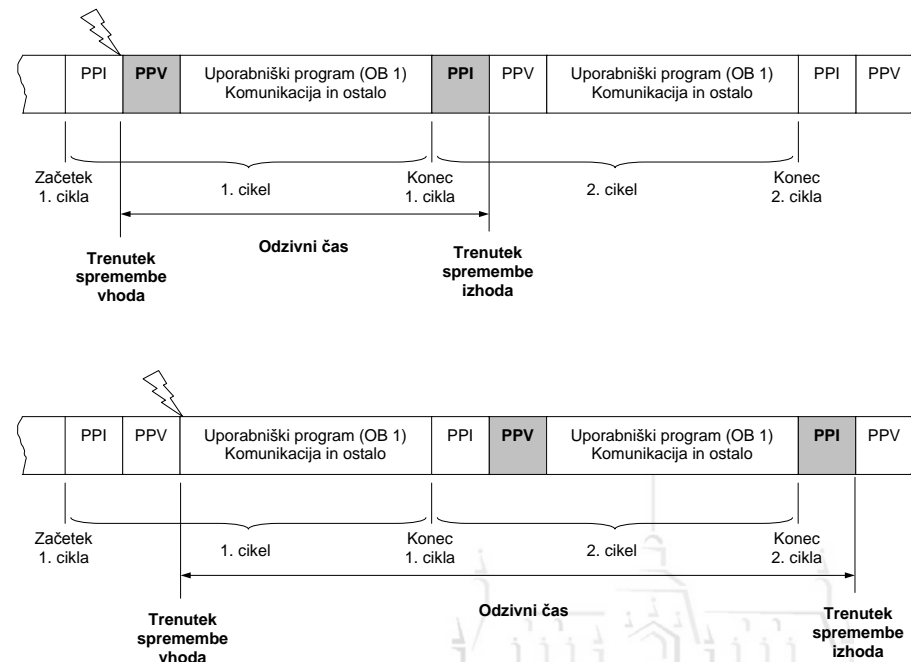
- Labela**: M003
- Operacija**: L
- Operand**: IW 2
- Komentar**: // naloži v AKU vhodno besedo

Slika 6.10: Zgled ukazne vrstice v STEP 7





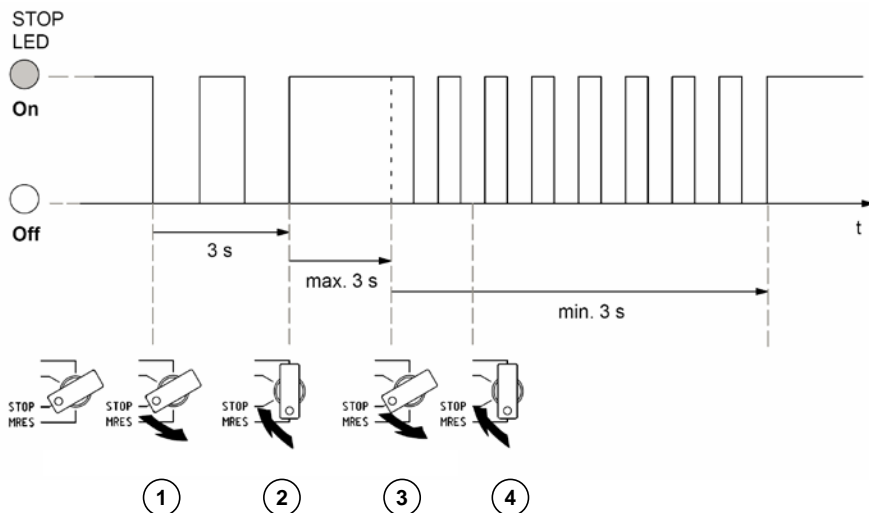
Slika 6.11: Opravila pri cikličnem obdelovanju programa



Slika 6.12: Najkrajši in najdaljši odzivni čas krmilja

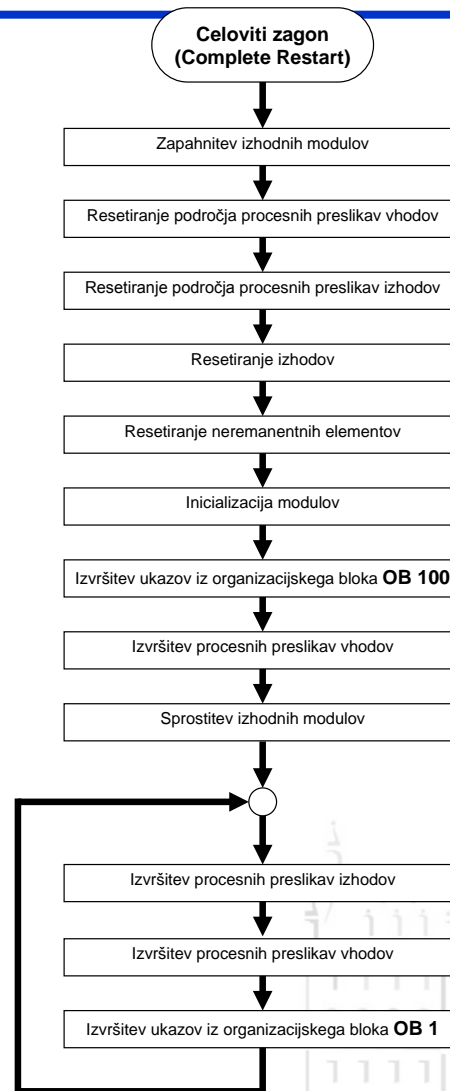


Krmilnik S7-300 - Resetiranje



Slika 6.13: Postopek resetiranja krmilnika s ključem v preklopniku

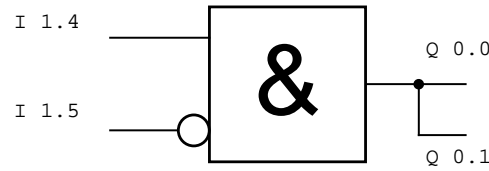
Slika 6.14: Postopki pri celovitem zagonu (Complete Restart) krmilnika





STEP 7 – Logične funkcije

Kombinacijske funkcije



Osnovni programski blok

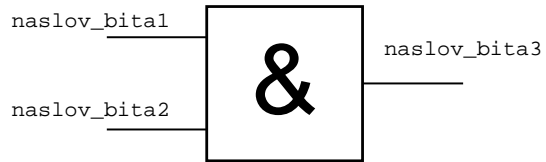
	STA	RLO	/FC	
...				
= M 3.1	X	X	0	Pogojna operacija
A I 1.4	1	1	1	Prvo povpraševanje
AN I 1.5	1	1	1	
= Q 0.0	1	1	0	Pogojna operacija
= Q 0.1	1	1	0	Pogojna operacija
A M 5.6	X	X	1	Prvo povpraševanje
...				

Slika 7.1: Preprost funkcijski načrt in pripadajoči program v STEP 7; prikazane vrednosti bitov v statusni besedi veljajo za I 1.4 na "1" in I 1.5 na "0"

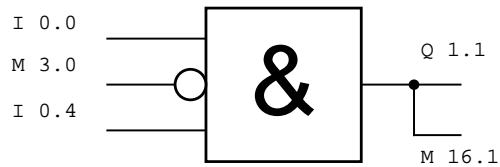


STEP 7 – Logične funkcije

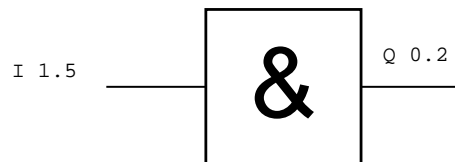
Logična funkcija IN (AND)



A naslov_bita1
A naslov_bita2
= naslov_bita3



A I 0.0
AN M 3.0
A I 0.4
= Q 1.1
= M 16.1



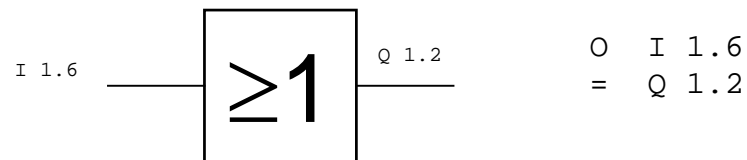
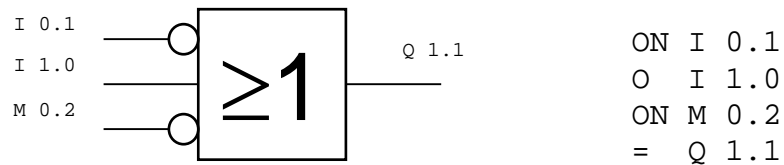
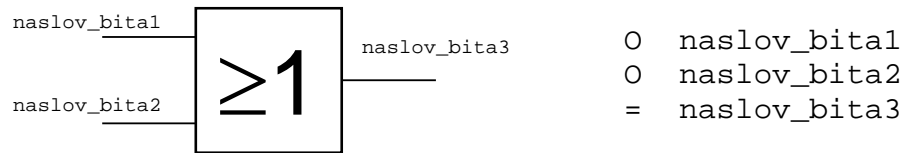
A I 1.5
= Q 0.2





STEP 7 – Logične funkcije

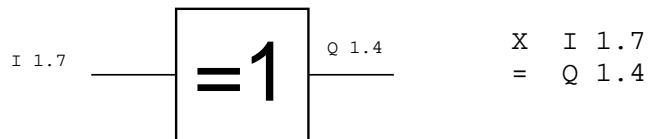
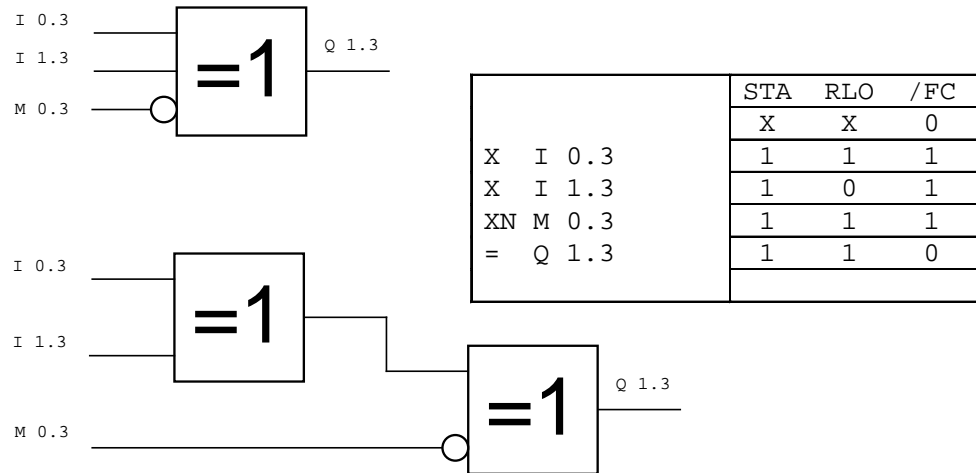
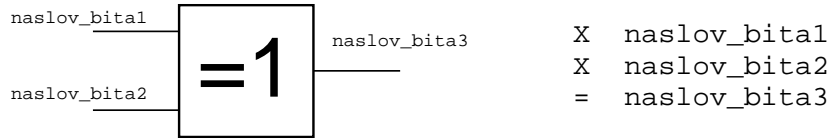
Logična funkcija ALI (OR)





STEP 7 – Logične funkcije

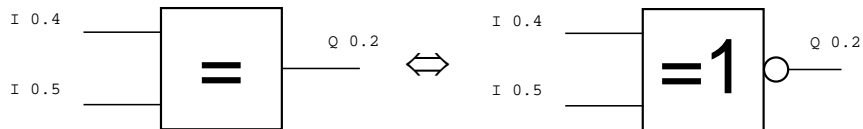
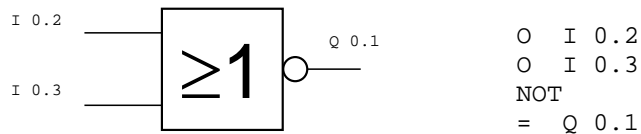
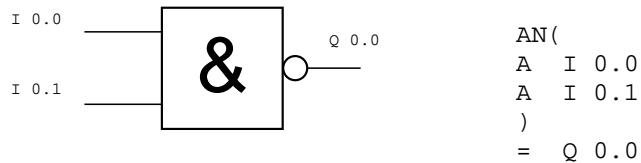
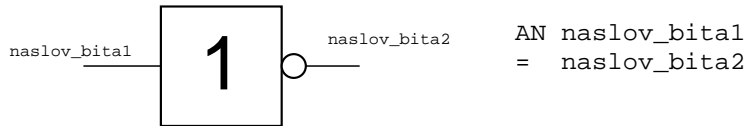
Logična funkcija EKSKLUZIVNI ALI (XOR)





STEP 7 – Logične funkcije

Negacija RLO





STEP 7 – Logične funkcije

Setiranje RLO

SET

= Q 0.0
= Q 0.1
= M 4.3

Brisanje RLO

CLR

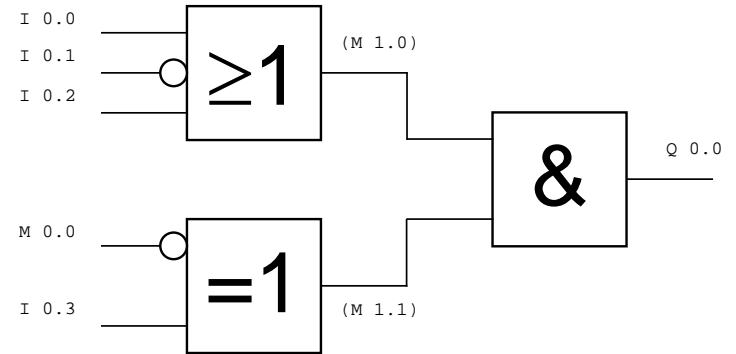
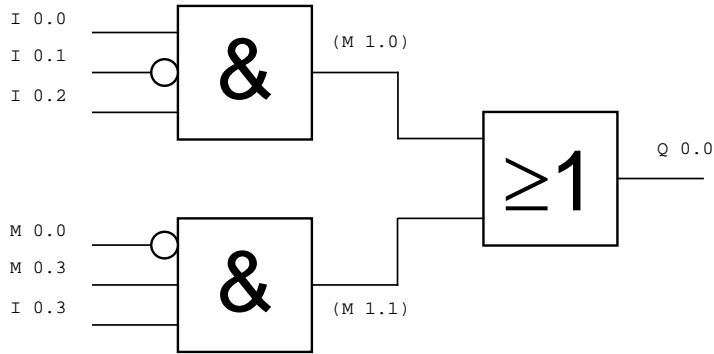
= Q 0.2
= Q 0.3
= M 4.4





STEP 7 – Logične funkcije

Sestavljene logične funkcije



```

A I 0.0
AN I 0.1
A I 0.2
= M 1.0
AN M 0.0
A M 0.3
A I 0.3
= M 1.1
O M 1.0
O M 1.1
= Q 0.0

```

```

A I 0.0
AN I 0.1
A I 0.2
O
AN M 0.0
A M 0.3
A I 0.3
= Q 0.0

```

```

O I 0.0
ON I 0.1
O I 0.2
= M 1.0
XN M 0.0
X I 0.3
= M 1.1
A M 1.0
A M 1.1
= Q 0.0

```

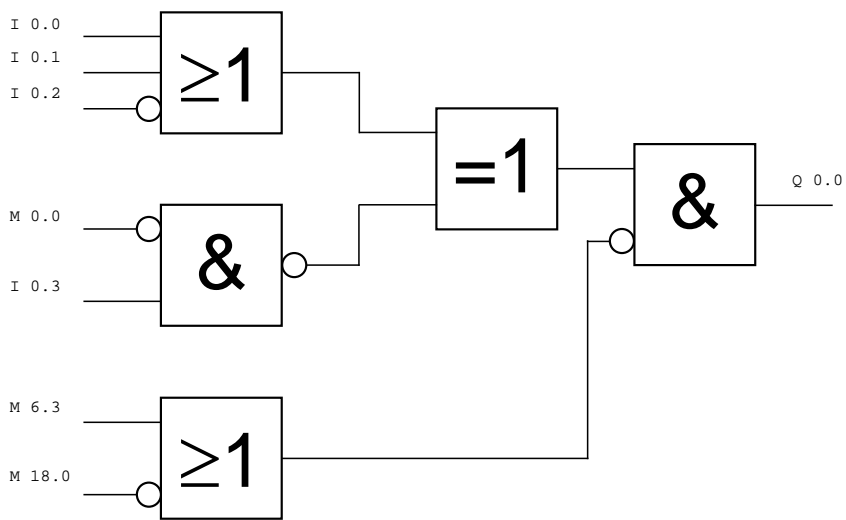
```

A(
O I 0.0
ON I 0.1
O I 0.2
)
A(
XN M 0.0
X I 0.3
)
= Q 0.0

```



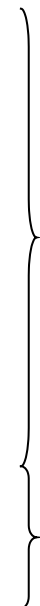
STEP 7 – Logične funkcije



```

A(
X(
O I 0.0
O I 0.1
ON I 0.2
)
)
X(
AN M 0.0
A I 0.3
NOT
)
)
AN(
O M 6.3
ON M 18.0
)
)
= Q 0.0

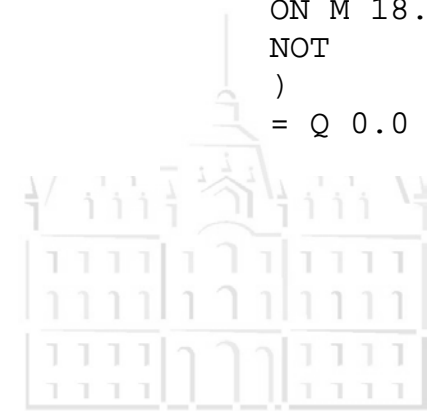
```



```

A(
X(
O I 0.0
O I 0.1
ON I 0.2
)
)
XN(
AN M 0.0
A I 0.3
)
)
A(
O M 6.3
ON M 18.0
NOT
)
)
= Q 0.0

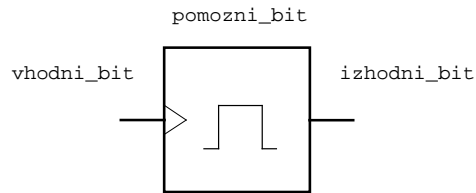
```



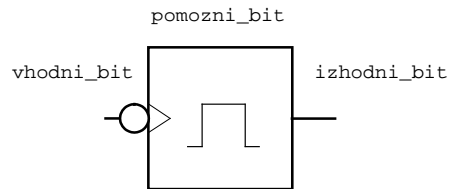
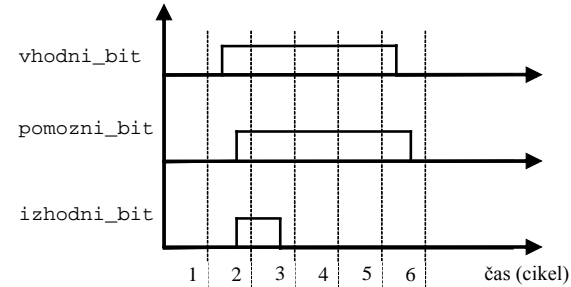


STEP 7 – Logične funkcije

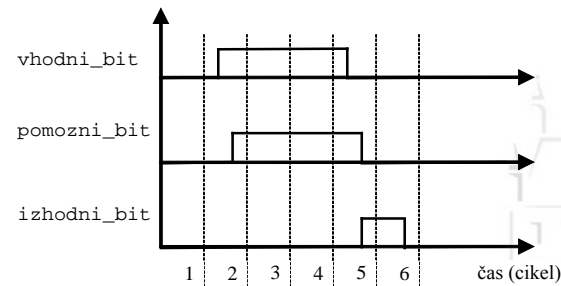
Dinamična logična funkcija



A vhodni_bit
FP pomozni_bit
= izhodni bit



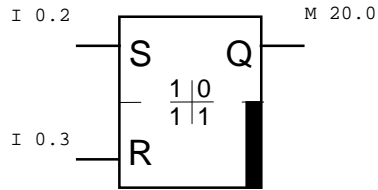
A vhodni_bit
FN pomozni_bit
= izhodni bit



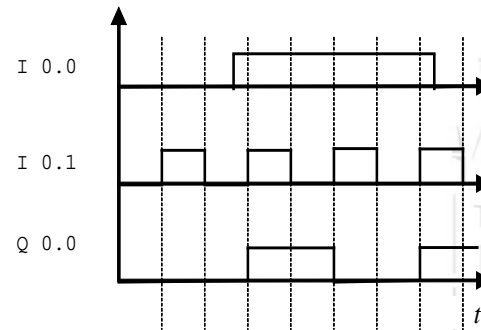
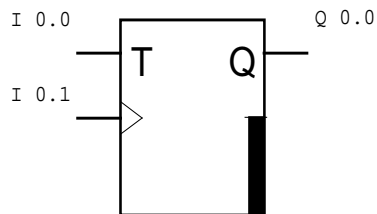
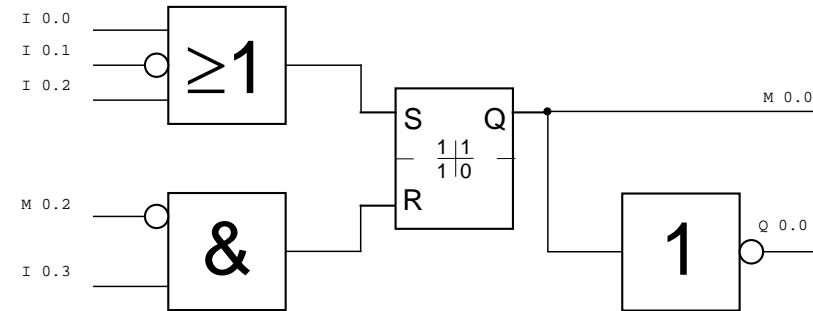


STEP 7 – Pomnilne funkcije

Pomnilne funkcije

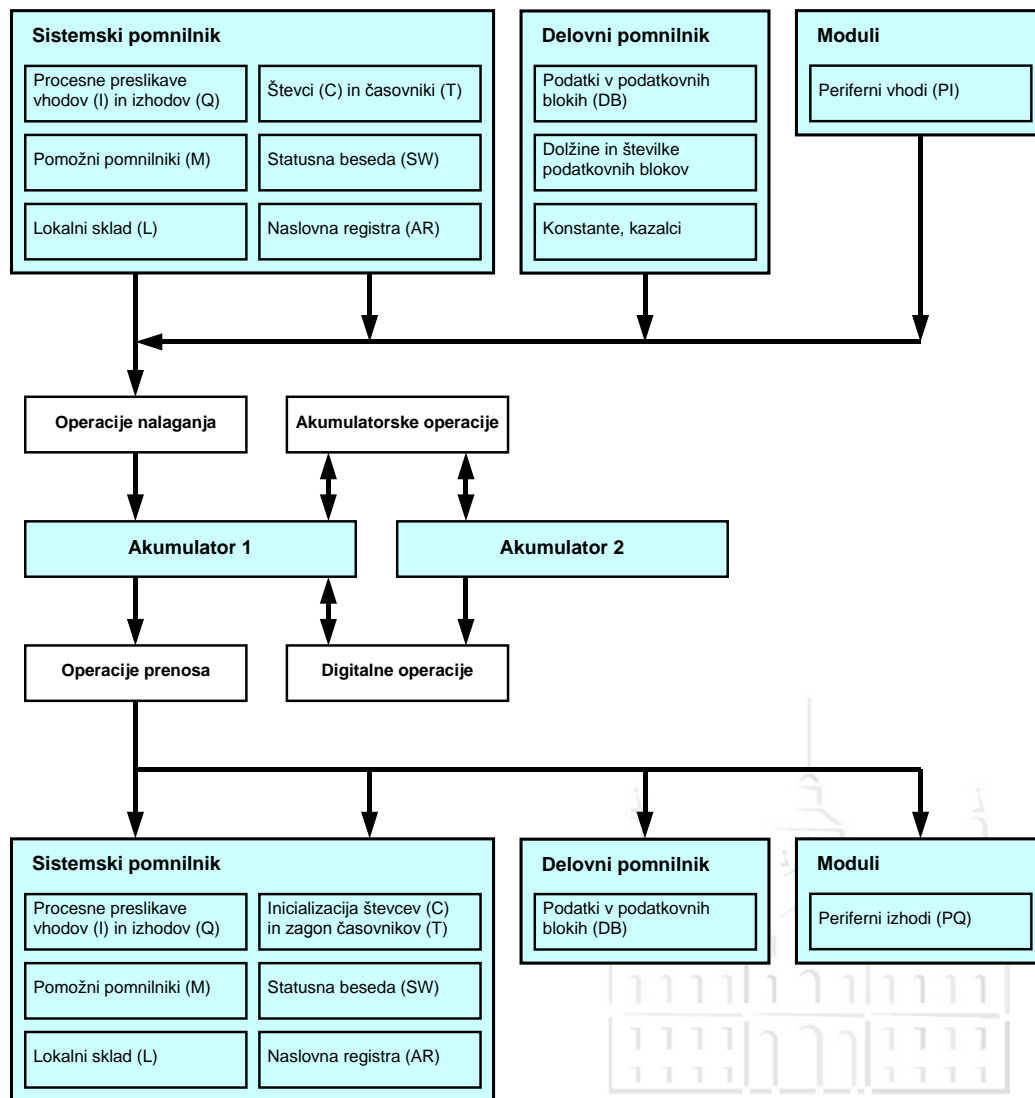


A I 0.2
S M 20.0
A I 0.3
R M 20.0





Akumulator



Slika 7.3: Različne možnosti pri delu z akumulatorjema



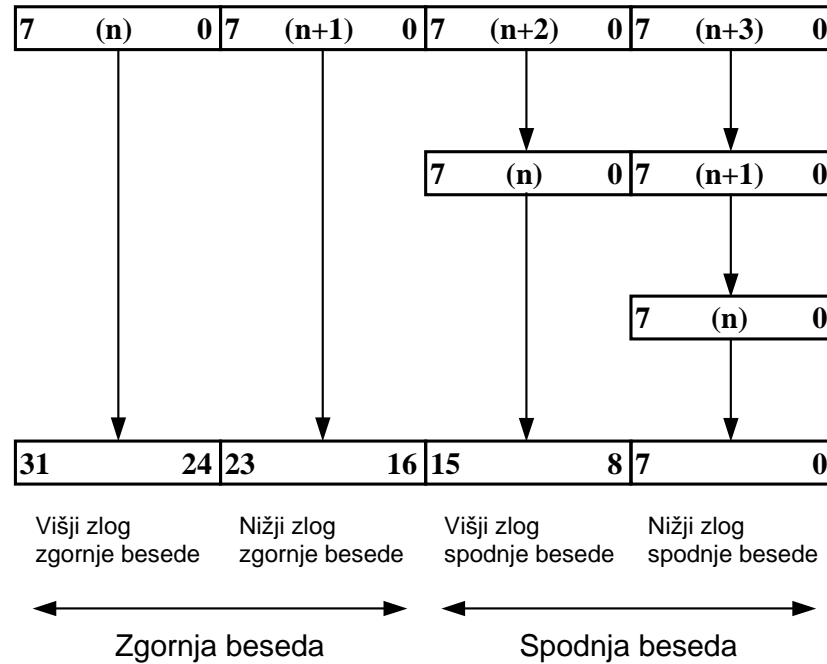
Nalaganje v akumulator (L - load)

Primer	Opis	Vsebina AKU1 po operaciji
L +5	Naloži 16-bitno celoštevilčno konstanto v AKU1.	16#0000_0005
L B#(1,10)	Naloži konstanto, sestavljeno iz dveh zlogov, v AKU1. (10 se naloži v najnižji zlog AKU1, 1 pa v sosednji, torej višji zlog spodnje besede AKU1).	16#0000_010A
L B#(1,10,5,50)	Naloži konstanto, sestavljeno iz štirih zlogov, v AKU1. (1 se naloži v zlog z najvišjo težo (višji zlog zgornje besede), 10 v nižji zlog zgornje besede, 5 v višji zlog spodnje besede in 50 v najnižji zlog AKU1).	16#010A_0532
L L#+5	Naloži 32-bitno celoštevilčno konstanto v AKU1.	16#0000_0005
L B#16#EF	Naloži 8-bitno šestnajstiško konstanto v AKU1.	16#0000_00EF
L W#16#FAFB	Naloži 16-bitno šestnajstiško konstanto v AKU1.	16#0000F_AFB
L DW#16#1FFE_1ABC	Naloži 32-bitno šestnajstiško konstanto v AKU1.	16#1FFE_1ABC
L 2#1111_0000_1110_0001	Naloži 16-bitno dvojiško konstanto v AKU1.	16#0000_F0E1
L 2#1001_0110_1111_0000_1110_0001_0111_1000	Naloži 32-bitno dvojiško konstanto v AKU1.	16#96F0_E178
L 'X6'	Naloži dva znaka v AKU1.	16#0000_5836
L '264U'	Naloži štiri znake v AKU1.	16#3236_3455
L C#105	Naloži 16-bitno števsko konstanto v AKU1.	16#0000_0105
L S5T#2S250MS	Naloži 16-bitno časovniško konstanto v AKU1.	16#0000_0225
L 1.3E+5	Naloži 32-bitno konstanto v AKU1 (IEEE format s plavajočo vejico).	16#47FD_E800
L P#I1.4	Naloži 32-bitni znotrajpodročni kazalec v AKU1.	16#8100_000C
L P#30.0	Naloži 32-bitni prekopolodročni kazalec v AKU1.	16#0000_00F0
L D#1994-03-15	Naloži 16-bitni datum v AKU1.	16#0000_05FE
L T#2d_14h_48m_14s_350ms	Naloži 32-bitno vrednost časa v AKU1.	16#0D79_ED0E
L TOD#01:10:3.300	Naloži 32-bitno vrednost dnevnega časa v AKU1.	16#0040_2324

Tabela 7.1: Zgledi nalaganja v akumulator za vsebine, podane v različnih formatih



Nalaganje v akumulator (L - load)



Dvojna beseda n (D)

Beseda n (W)

Zlog n (B)

Akumulator 1

Slika 7.4: Nalaganje v akumulator za operande, ki imajo različno podatkovno širino





Nalaganje v akumulator (L - load)

Za navedene primere predpostavimo, da je vsebina akumulatorja AKU1 pred izvršitvijo posameznega ukaza L poljubna, stanje na krmilniku pa naslednje:

```
ID 0:      2#0110_0101_1001_1111_0010_0111_1100_0011 = 16#659F_27C3
QD 4:      2#1111_1101_0001_0101_1100_1001_1001_1010 = 16#FD15_C99A
PID 124:   2#0000_1000_0101_1101_1001_1011_0001_1101 = 16#085D_9B1D
MD 16:     2#0111_1100_0001_0011_1101_1000_1111_1000 = 16#7C13_D8F8
```

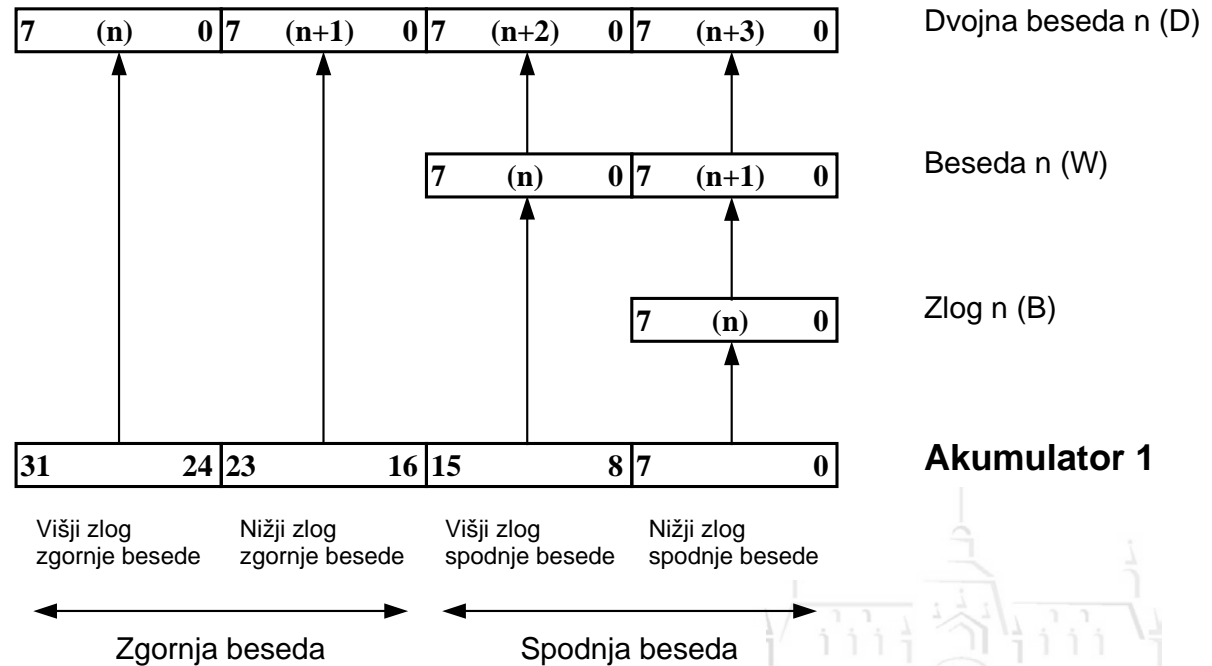
Primer	Opis	Vsebina AKU1 po operaciji
L IB 0	Naloži vsebino vhodnega zloga 0 v najnižji zlog AKU1	16#0000_0065
L IW 0	Naloži vsebino vhodne besede 0 na spodnjo besedo AKU1	16#0000_659F
L ID 0	Naloži vsebino vhodne dvojne besede 0 v AKU1	16#659F_27C3
L QB 6	Naloži vsebino izhodnega zloga 6 v najnižji zlog AKU1	16#0000_00C9
L QW 5	Naloži vsebino izhodne besede 5 na spodnjo besedo AKU1	16#0000_15C9
L QD 4	Naloži vsebino izhodne dvojne besede 4 v AKU1	16#FD15_C99A
L PIB 127	Naloži vsebino perifernega vhodnega zloga 127 v najnižji zlog AKU1	16#0000_001D
L PIW 126	Naloži vsebino periferne vhodne besede 126 na spodnjo besedo AKU1	16#0000_9B1D
L PID 124	Naloži vsebino periferne vhodne dvojne besede 124 v AKU1	16#085D_9B1D
L MB 17	Naloži vsebino pomožnega pomnilniškega zloga 17 v najnižji zlog AKU1	16#0000_0013
L MW 16	Naloži vsebino pomožne pomnilniške besede 16 na spodnjo besedo AKU1	16#0000_7C13
L MD 16	Naloži vsebino pomožne pomnilniške dvojne besede 16 v AKU1	16#7C13_D8F8

Tabela 7.2: Zgledi nalaganja v akumulator za operande, ki imajo različno podatkovno širino



Prenos iz akumulatorja (T - transfer)

Slika 7.5: Prenos vsebine
akumulatorja na operande, ki
imajo različno podatkovno
širino





STEP 7 - Akumulatorske operacije

Prenos iz akumulatorja (T - transfer)

Za navedene primere predpostavimo, da je vsebina akumulatorja AKU1 pred izvršitvijo posameznega ukaza T:

AKU1: 2#1000_1001_1010_1011_1100_1101_1110_0000 = 16#89AB_CDE0

stanje na krmilniku pa naslednje:

QD 0: 2#1111_1111_1111_1111_1111_1111_1111_1111 = 16#FFFF_FFFF

MD 16: 2#1111_1111_1111_1111_1111_1111_1111_1111 = 16#FFFF_FFFF

Primer	Opis	Vsebina naslova po operaciji
T QB 3	Prenese vsebino najnižjega zloga AKU1 na QB 3	QD 0: 16#FFFF_FFE0
T QW 2	Prenese vsebino spodnje besede AKU1 na QW 2	QD 0: 16#FFFF_CDE0
T QD 0	Prenese vsebino AKU1 na QD 0	QD 0: 16#89AB_CDE0
T MB 17	Prenese vsebino najnižjega zloga AKU1 na MB 17	MD 16: 16#FFE0_FFFF
T MW 16	Prenese vsebino spodnje besede AKU1 na MW 16	MD 16: 16#CDE0_FFFF
T MD 16	Prenese vsebino AKU1 na MD 16	MD 16: 16#89AB_CDE0

Tabela 7.3: Zgledi prenosa vsebine akumulatorja na operande z različno podatkovno širino



Digitalne logične operacije - AW

```
... // prej          AKU1: 16#89AB_CDEF = 2#1000_1001_1010_1011_1100_1101_1110_1111
                        AKU2: 16#0123_6543 = 2#0000_0001_0010_0011_0110_0101_0100_0011
```

AW

```
... // potem        AKU1: 16#89AB_4543 = 2#1000_1001_1010_1011_0100_0101_0100_0011
                        AKU2: 16#0123_6543 = 2#0000_0001_0010_0011_0110_0101_0100_0011
```

```
... // prej          AKU1: 16#89AB_CDEF = 2#1000_1001_1010_1011_1100_1101_1110_1111
```

```
AW W#16#678F //          16#678F = 2#0110_0111_1000_1111
```

```
... // potem        AKU1: 16#89AB_458F = 2#1000_1001_1010_1011_0100_0101_1000_1111
```

```
L MD 40 // MD 40 v AKU1
```

```
L IW 0 // IW 0 v AKU1, MD 40 gre v AKU2!
```

```
AW // pozor: kaj je spodnjih 16 bitov!?
```

```
T QW 0 // rezultat na QW 0
```





Digitalne logične operacije - AW

```
L   IW 0           // IW 0 v AKU1
AW  W#16#F00F     // maska: prepušča najvišji in najnižji nibble
T   QW 0           // rezultat na QW 0
```

```
L   IW 0           // AKU1: 16#0000_XXXX = 2#0000_0000_0000_0000_XXXX_XXXX_XXXX_XXXX
AW  W#16#F00F     //           16#F00F =                2#1111_0000_0000_1111
T   QW 0           // AKU1: 16#0000_X00X = 2#0000_0000_0000_0000_XXXX_0000_0000_XXXX
```





Digitalne logične operacije - AW

```
... // prej AKU1: 16#89AB_CDEF = 2#1000_1001_1010_1011_1100_1101_1110_1111
AKU2: 16#0123_6543 = 2#0000_0001_0010_0011_0110_0101_0100_0011
```

OW

```
... // potem AKU1: 16#89AB_EDEF = 2#1000_1001_1010_1011_1110_1101_1110_1111
AKU2: 16#0123_6543 = 2#0000_0001_0010_0011_0110_0101_0100_0011
```

```
... // prej AKU1: 16#89AB_CDEF = 2#1000_1001_1010_1011_1100_1101_1110_1111
```

```
OW W#16#678F // 16#678F = 2#0110_0111_1000_1111
```

```
... // potem AKU1: 16#89AB_EFEF = 2#1000_1001_1010_1011_1110_1111_1110_1111
```

```
L IB 0 // IB 0 v AKU1
L QW 0 // QW 0 v AKU1, IB 0 gre v AKU2!
OW //
T MW 16 // rezultat na MW 16
```

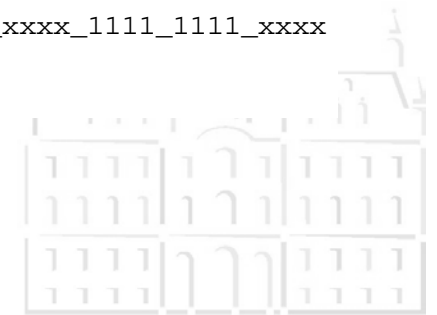




Digitalne logične operacije - AW

```
L   IW 0           // IW 0 v AKU1
OW  W#16#0FF0     // maska: prepušča najvišji in najnižji nibble
T   QW 0           // rezultat na QW 0
```

```
L   IW 0           //   AKU1: 16#0000_XXXX = 2#0000_0000_0000_0000_xxxx_xxxx_xxxx_xxxx
OW  W#16#0FF0     //           16#0FF0 =                2#0000_1111_1111_0000
T   QW 0           //   AKU1: 16#0000_XFFX = 2#0000_0000_0000_0000_xxxx_1111_1111_xxxx
```





Digitalne logične operacije - XOW

```
... // prej AKU1: 16#89AB_CDEF = 2#1000_1001_1010_1011_1100_1101_1110_1111
AKU2: 16#0123_6543 = 2#0000_0001_0010_0011_0110_0101_0100_0011
```

XOW

```
... // potem AKU1: 16#89AB_A8AC = 2#1000_1001_1010_1011_1010_1000_1010_1100
AKU2: 16#0123_6543 = 2#0000_0001_0010_0011_0110_0101_0100_0011
```

```
... // prej AKU1: 16#89AB_CDEF = 2#1000_1001_1010_1011_1100_1101_1110_1111
```

```
XOW W#16#678F // 16#678F = 2#0110_0111_1000_1111
```

```
... // potem AKU1: 16#89AB_AA60 = 2#1000_1001_1010_1011_1010_1010_0110_0000
```

```
L MW 0 // MW 0 v AKU1
```

```
L IW 0 // IW 0 v AKU1, MW 0 gre v AKU2!
```

```
XOW //
```

```
T QW 124 // rezultat na QW 124
```





Digitalne logične operacije - XOW

```
L   IW 0           // IW 0 v AKU1
XOW W#16#0FF0     // maska: prepušča najvišji in najnižji nibble
T   QW 0          // rezultat na QW 0
```

```
L   IW 0           // AKU1: 16#0000_XXXX = 2#0000_0000_0000_0000_xxxx_xxxx_xxxx_xxxx
XOW W#16#0FF0     //           16#0FF0 =                2#0000_1111_1111_0000
T   QW 0          // AKU1: 16#0000_XYYX = 2#0000_0000_0000_0000_xxxx_yyyy_yyyy_xxxx
```





Aritmetični operaciji INC in DEC

```
L +2208 // AKU1: 16#0000_08A0 = B#( 0, 0, 8,160) = 2208
INC 92 // 16#5C = B#( 0, 0, 0, 92) = 92
T MW 20 // AKU1: 16#0000_08FC = B#( 0, 0, 8,252) = 2300
```

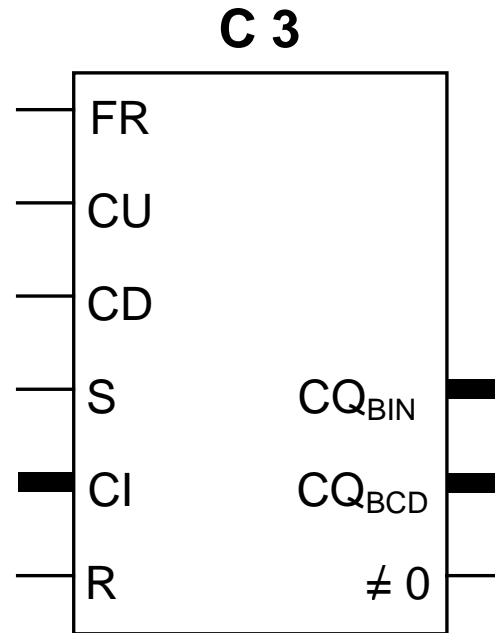
```
L +2208 // AKU1: 16#0000_08A0 = B#( 0, 0, 8,160) = 2208
INC 114 // 16#72 = B#( 0, 0, 0,114) = 114
T MW 20 // AKU1: 16#0000_0812 = B#( 0, 0, 8, 18) = 2066
```

```
L +555 // AKU1: 16#0000_022B = B#( 0, 0, 2, 43) = 555
DEC 55 // 16#72 = B#( 0, 0, 0, 55) = 55
T QW 0 // AKU1: 16#0000_02F4 = B#( 0, 0, 2,244) = 756
```





STEP 7 - Števec



Slika 7.6: Grafični simbol za števec





STEP 7 - Števec

Vpisovanje začetnega stanja števca

```
L C#142 // naloži v AKU začetno vrednost 142 dogodkov  
S C 3 // setira števec C3
```

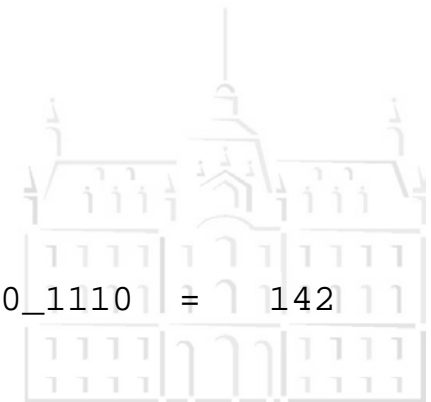
AKU1: 16#0000_0142 = 2#0000_0000_0000_0000_0000_0001_0100_0010 = C#142

Spodnja beseda AKU1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2#	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0
16#	0			1			4			2						

```
L +142 // naloži v AKU začetno vrednost 142  
S C 3 // setira števec C3
```

AKU1: 16#0000_008E = 2#0000_0000_0000_0000_0000_0000_1000_1110 = 142





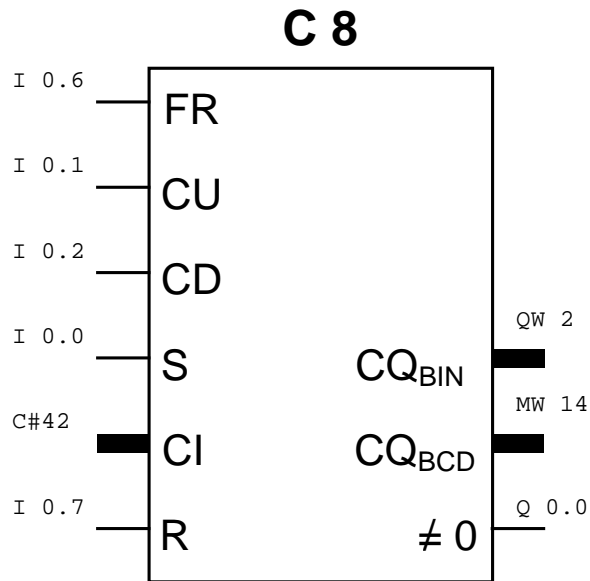
Branje tekočega stanja števca

```
...  
L   C 4      // AKU1: 16#0000_01F2 = 2#0000_0000_0000_0000_0000_0001_1111_0010 = 498  
T   MW 20  
LC  C 4      // AKU1: 16#0000_0498 = 2#0000_0000_0000_0000_0000_0100_1001_1000 = 1176  
T   MW 22  
...
```





STEP 7 - Števec



```
A I 0.6
FR C 8 // sprostittev

A I 0.1
CU C 8 // štetje navzgor

A I 0.2
CD C 8 // štetje navzdol

A I 0.0
L C#42
S C 8 // začetna vr.

A I 0.7
R C 8 // resetiranje

A C 8 // povpraševanje
= Q 0.0 // števec<>0

L C 8 // binarno
T QW 2

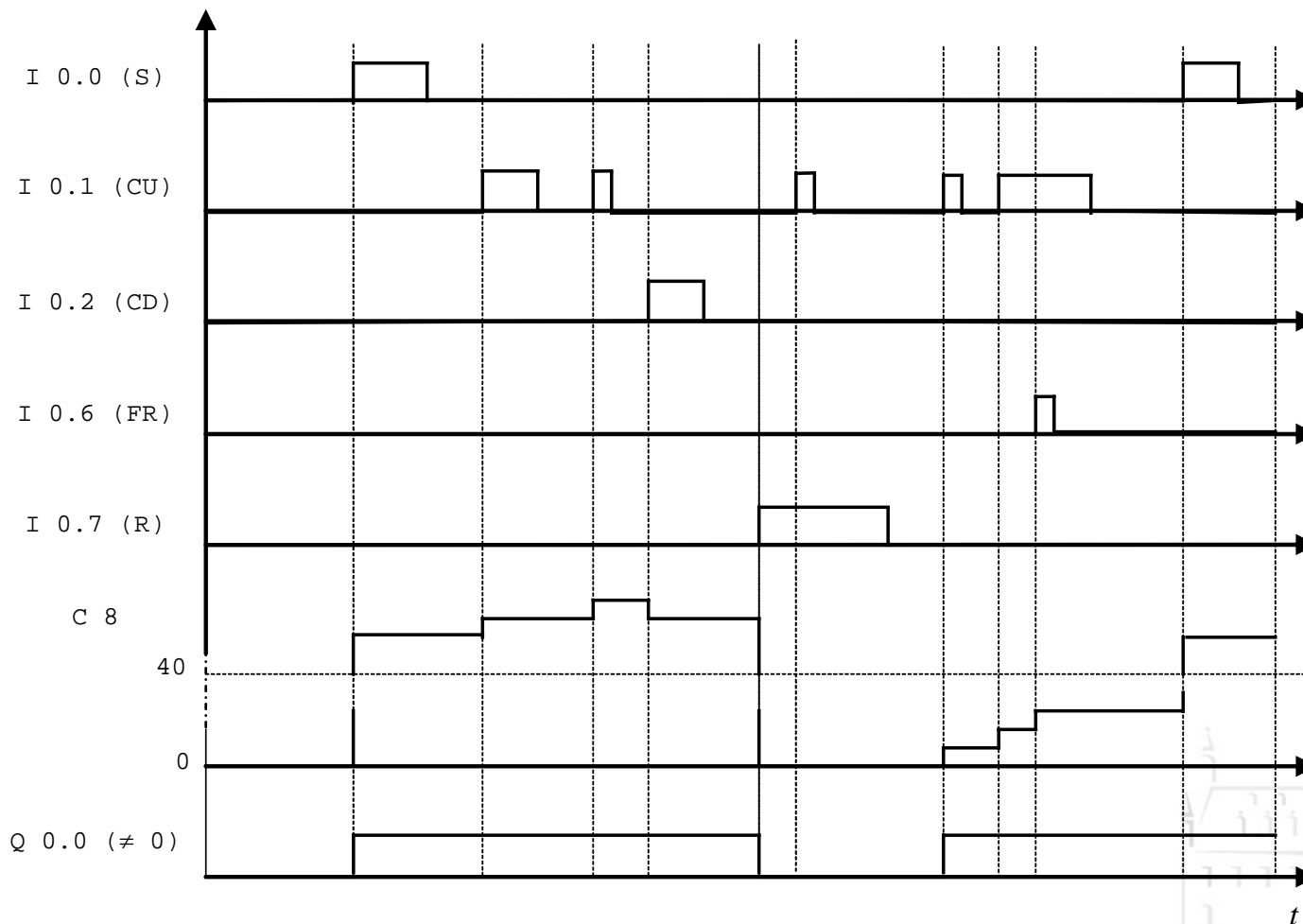
LC C 8 // BCD
T MW 14
```

Slika 7.7: Zgled za uporabo števca





STEP 7 - Števec

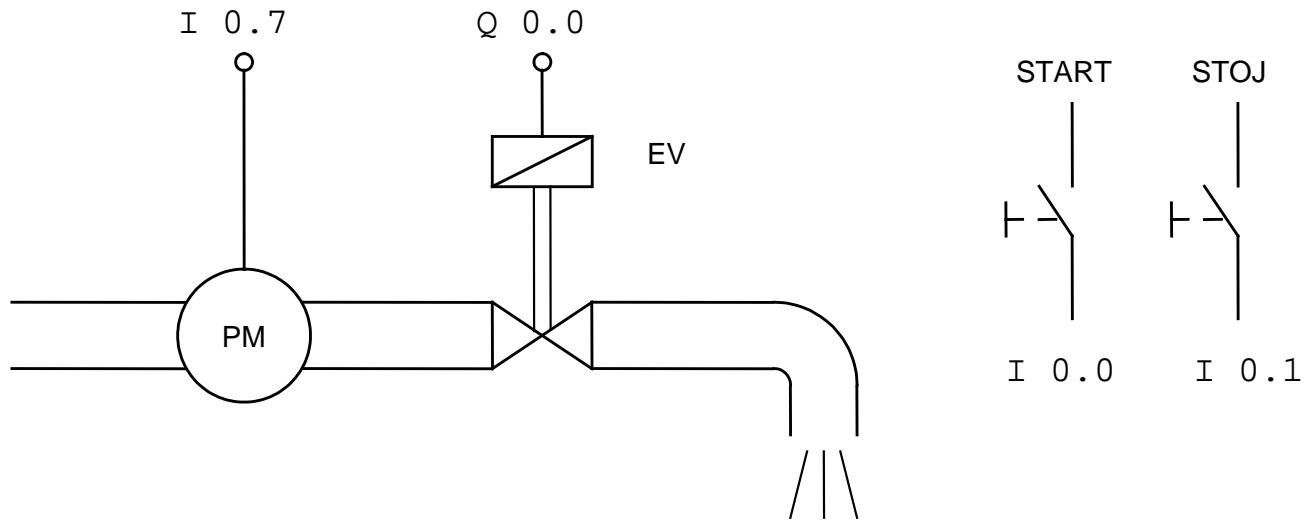


Slika 7.8: Časovni potek signalov pri števcu



STEP 7 - Števec

Zgled

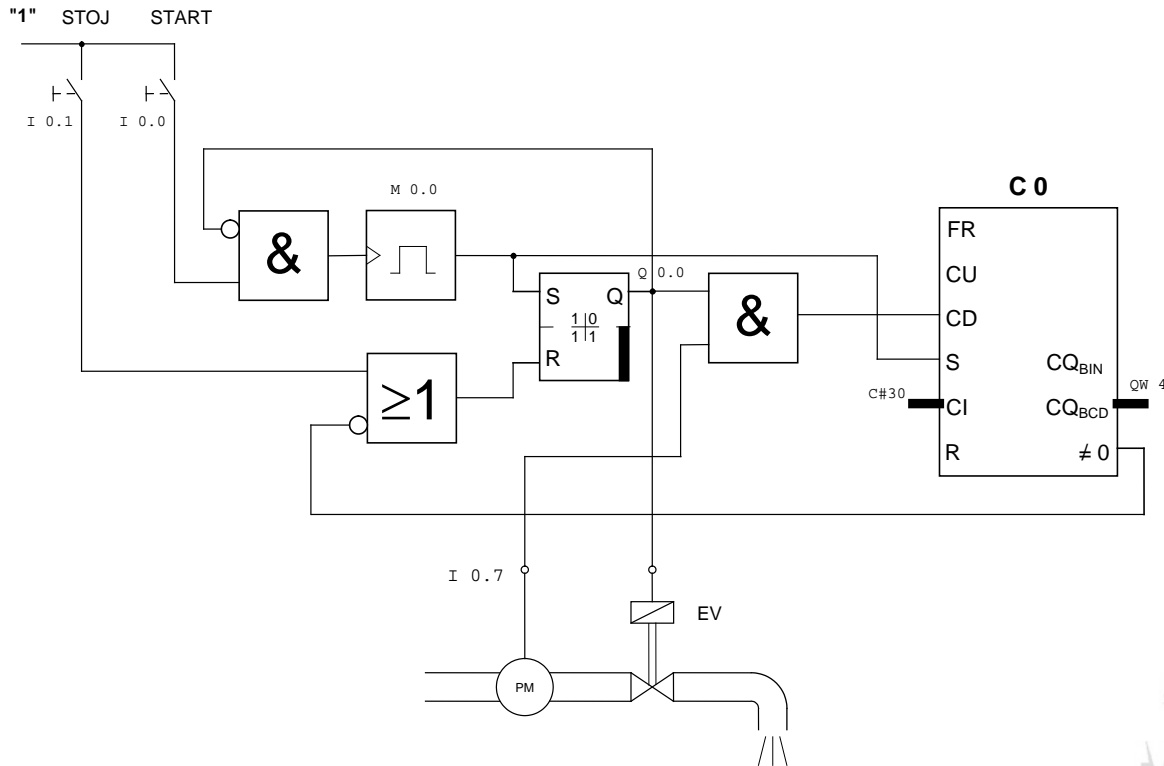


Slika 7.9: Tehnološka shema procesa doziranja tekočine





Zgled



Slika 7.10: Funkcijski načrt za krmiljenje procesa doziranja tekočine



STEP 7 - Števec

Zgled

Št.	Naslov	Enota	Tehno- oznaka	Opis	Opomba
1	I 0.0	vhod	START	tipkalo	
2	I 0.1	vhod	STOJ	tipkalo	
3	I 0.7	vhod	PM	pretočni merilnik	1 impulz za pretočeni liter
4	Q 0.0	izhod	EV	elektromagnetni ventil	
5	QW 4	izhod	–	prikaz stanja števca BCD	
6	C 0	števec	–	števec pretočenih litrov	odšteva pretočene litre
7	M 0.0	pomn.	–	pomožni pomnilnik za DLF	pri pogoju za odprtje EV

Tabela 7.4: Prireditvena tabela za krmilje pri procesu doziranja tekočine



STEP 7 - Števec

Zgled

```
A   Q 0.0    // če je ventil EV odprt
A   I 0.7    // in če je impulz z merilnika PM
CD  C 0      // odštej dogodek v števcu C 0

AN  Q 0.0    // če je ventil EV zaprt
A(          // in
A   I 0.0    // če je tipka START
FP  M 0.0    // dinamični člen
)

S   Q 0.0    // odpri ventil EV
L   C#30     // naloži v AKU1 30 dogodkov
S   C 0      // setira števec C 0 ob spremembi RLO z 0 na 1

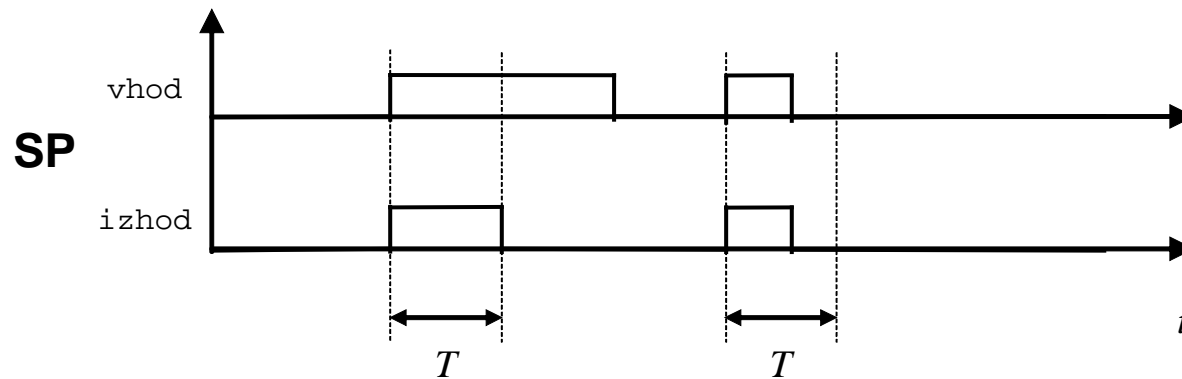
O   I 0.1    // če je tipka STOJ
ON  C 0      // ali če je število dogodkov v C 0 enako 0
R   Q 0.0    // zapri ventil EV
LC  C 0      // naloži v AKU1 vsebino C 0 v BCD obliki
T   QW 4     // na QW 4 bomo po nibbljih opazovali BCD številke
```





STEP 7 – Časovnik

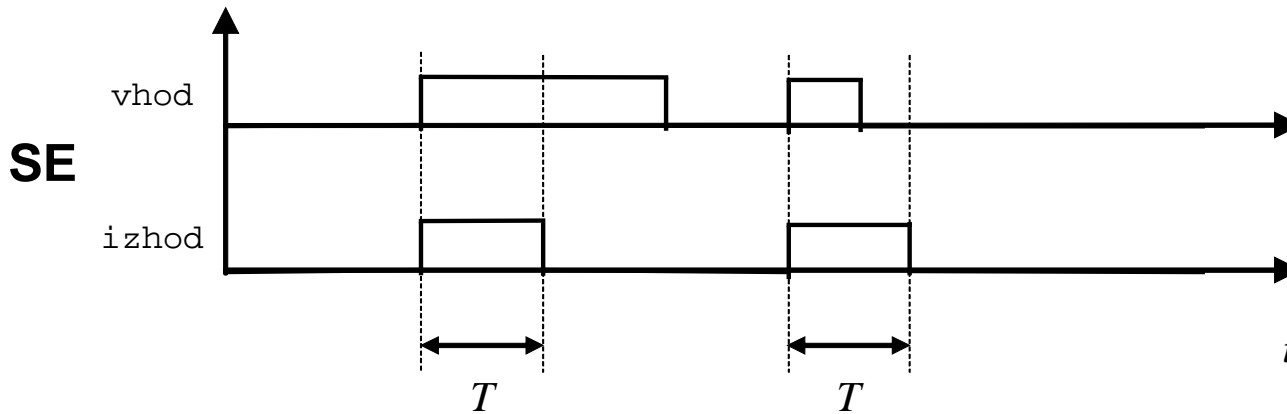
Skrajšanje pulza (SP)





STEP 7 – Časovnik

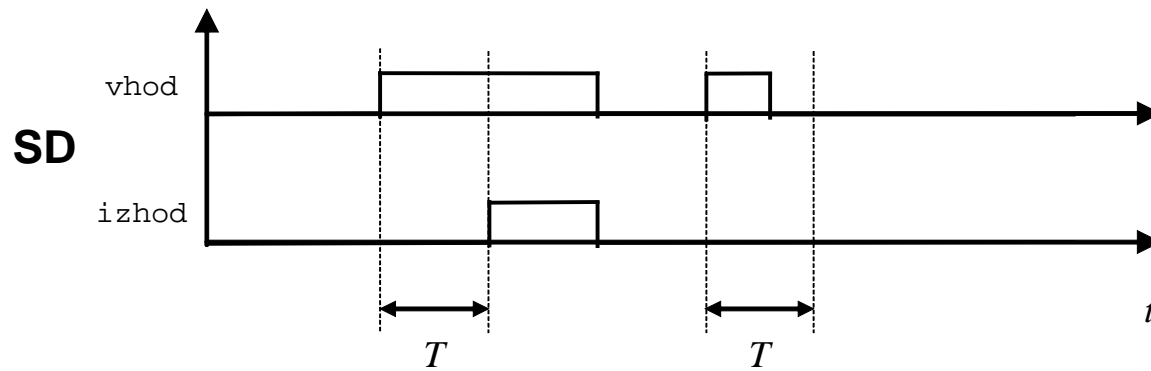
Podaljšanje pulza (SE)





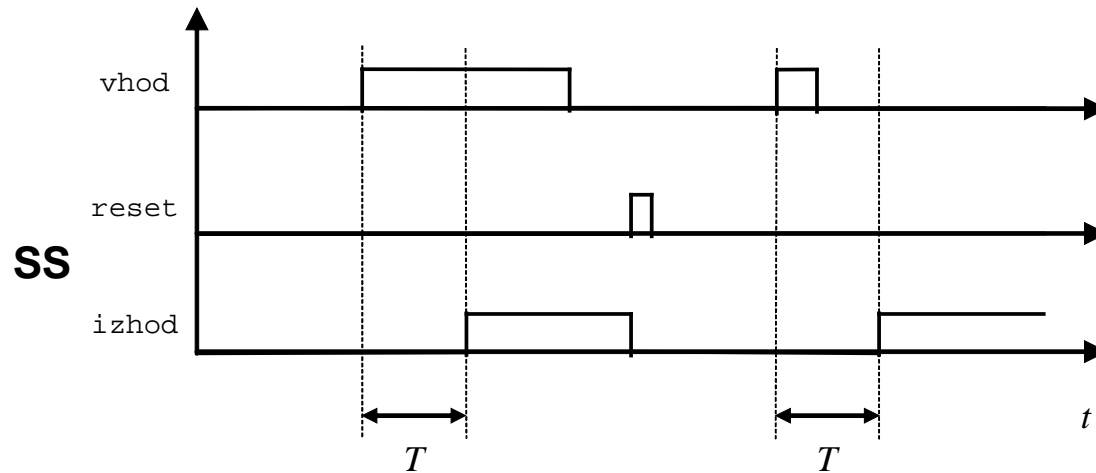
STEP 7 – Časovnik

Zakasnitev vklopa (SD)



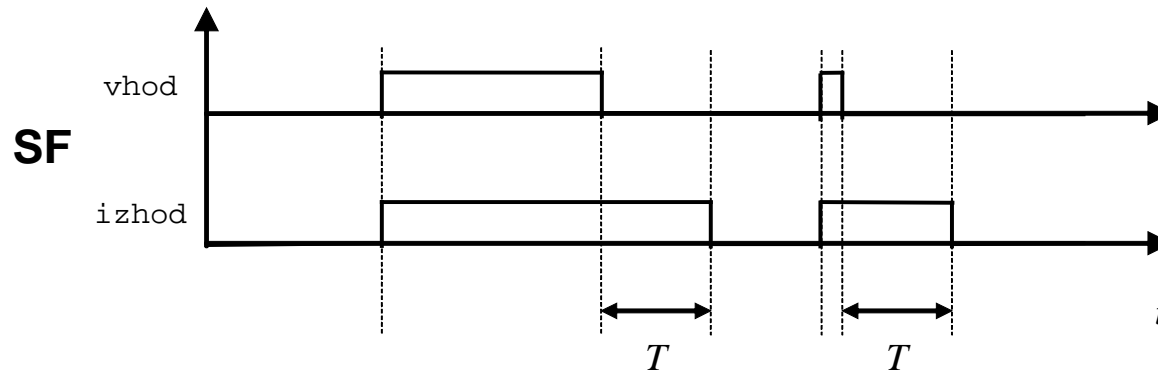


Zakasnitev vklopa s spominom (SS)



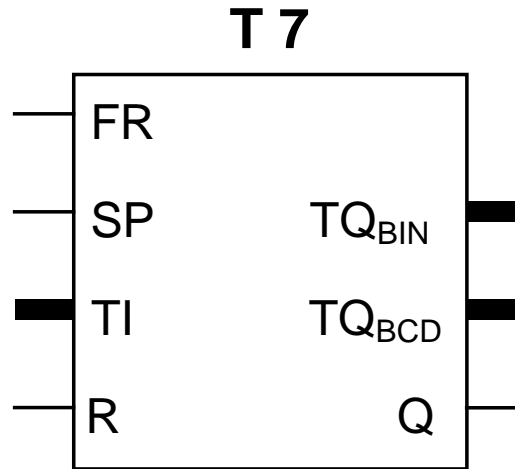


Zakasnitev izklopa (SF)





STEP 7 – Časovnik



Slika 7.11: Grafični simbol za časovnik





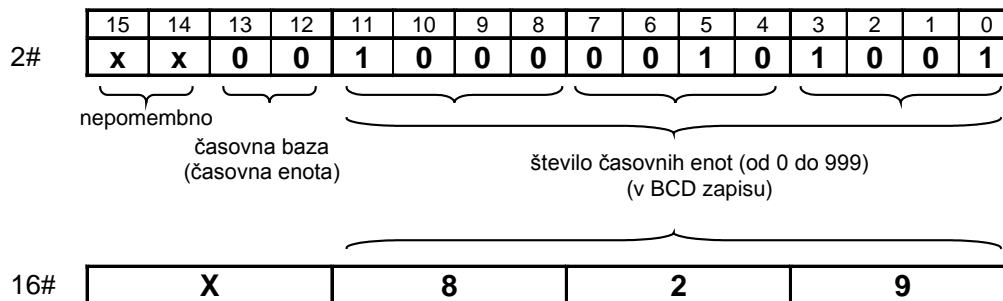
Postavitev časovnika v začetno stanje (1)

```

L S5T#8S290MS // naloži v AKU1 časovno konstanto 8,29 s
SP T 7 // naloži časovno konstanto v časovnik T 7 in
// ... sproži funkcijo "skrajšanje pulza"

```

S5T# časovniški format



Koda časovne baze (stanje bitov 13 in 12)	Časovna enota
0 0	10 ms
0 1	100 ms
1 0	1 s
1 1	10 s





STEP 7 – Časovnik

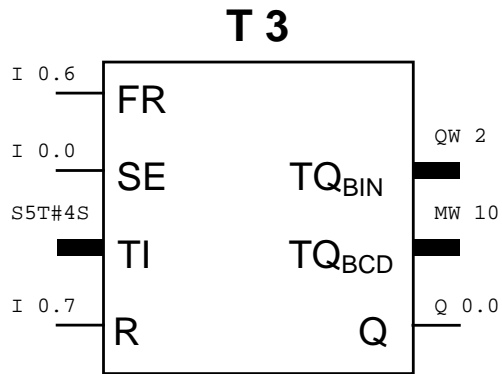
Postavitev časovnika v začetno stanje (2)

```
L S5T#2H46M30S // AKU1: 16#0000_3999 =  
                // 2#0000_0000_0000_0000_0011_1001_1001_1001  
SP T 7
```





STEP 7 – Časovnik



```
A I 0.6
FR T 3 // sprostitvev

A I 0.0 // vhodni signal
L S5T#4S // čas. konst.
SE T 3 // čas. funkcija

A I 0.7
R T 3 // resetiranje

A T 3 // povpraševanje
= Q 0.0 // izhodni signal

L T 3 // binarno
T QW 2

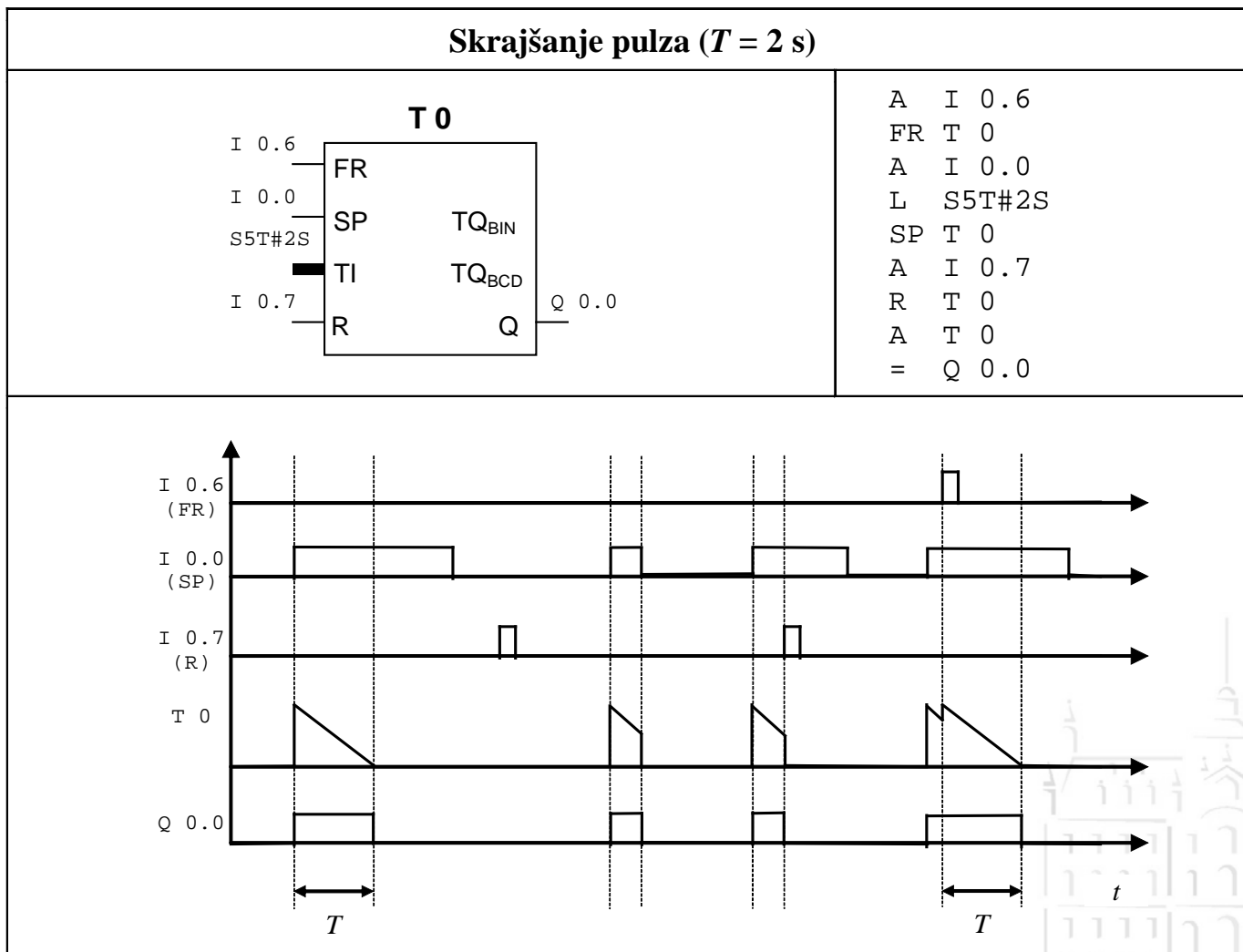
LC T 3 // BCD
T MW 10
```

Slika 7.12: Funkcijski načrt in program za zgled uporabe časovnika



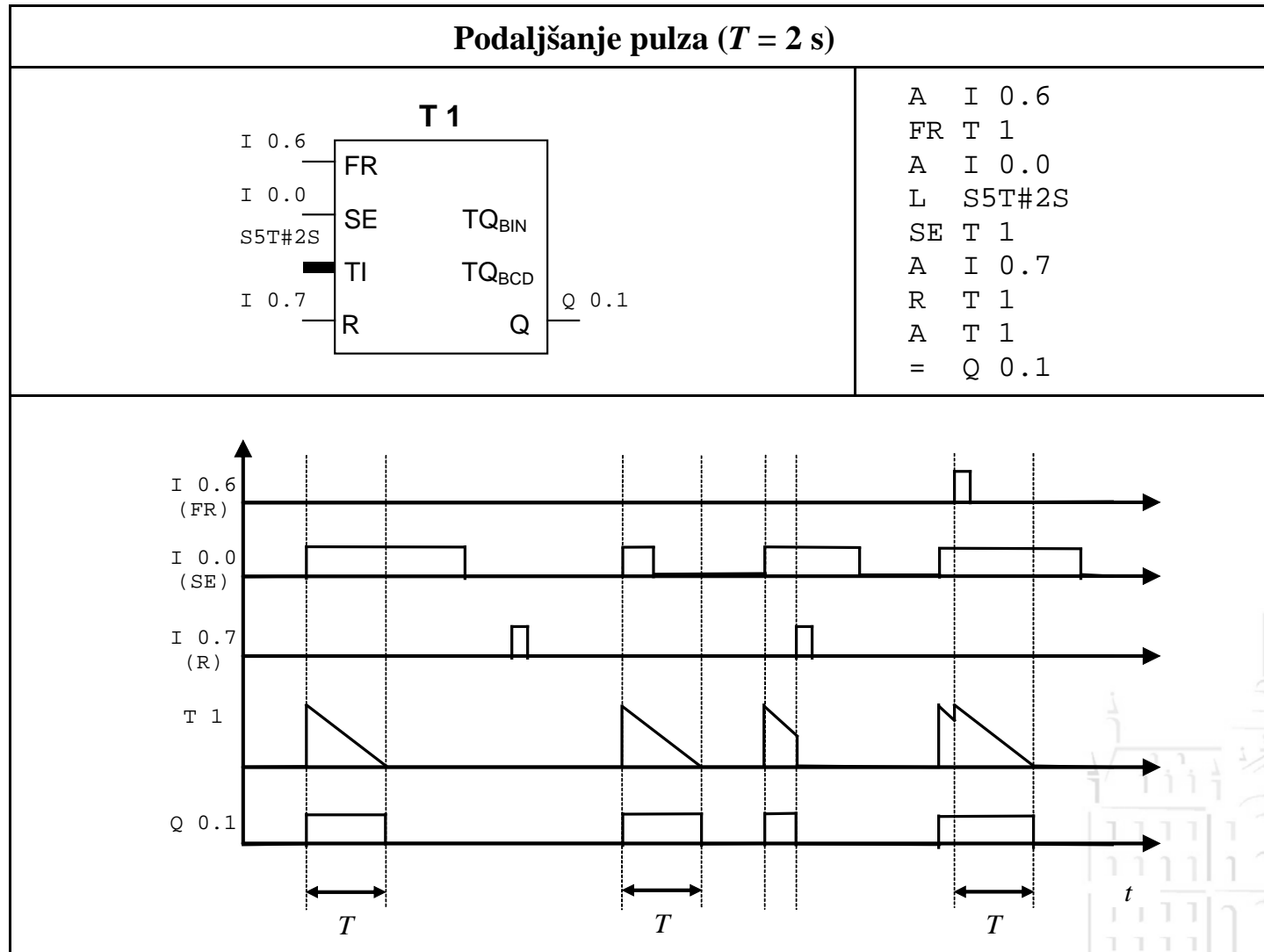


STEP 7 – Časovnik



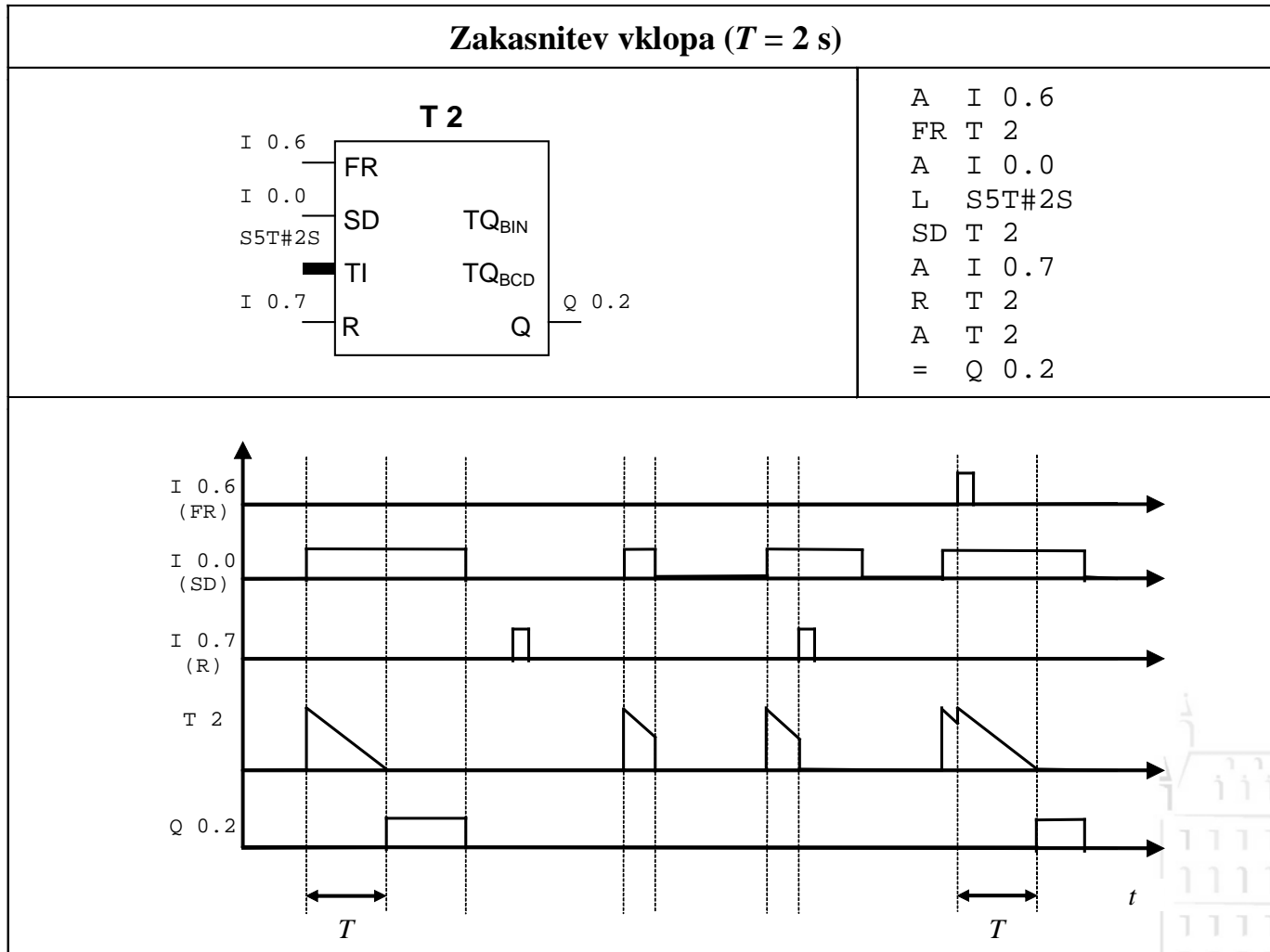


STEP 7 – Časovnik





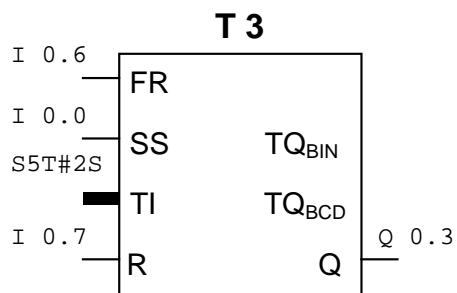
STEP 7 – Časovnik



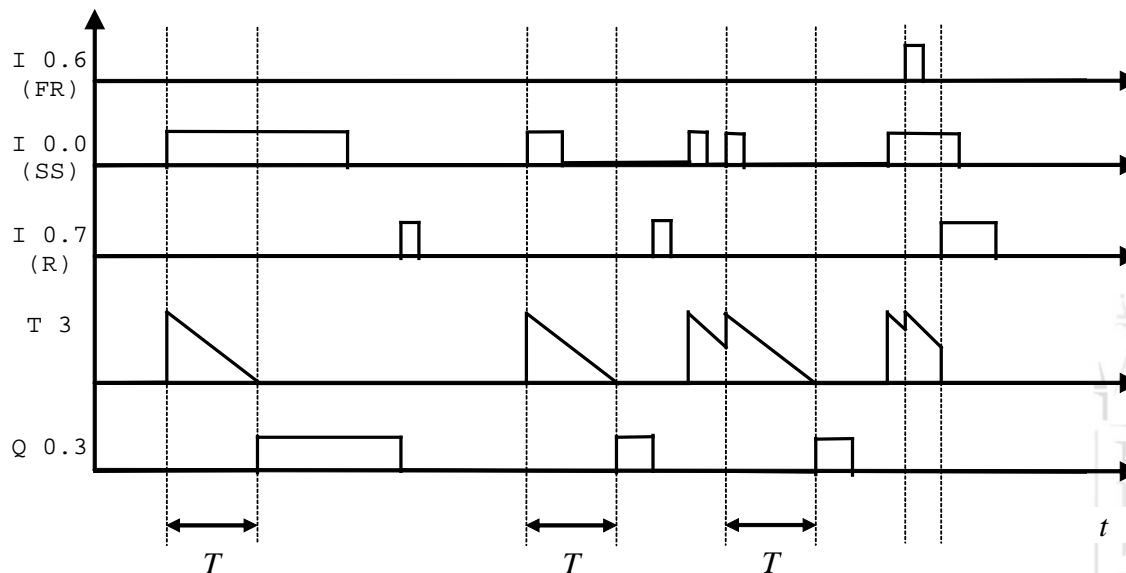


STEP 7 – Časovnik

Zakasnitev vklopa s spominom ($T = 2$ s) /obvezno resetiranje



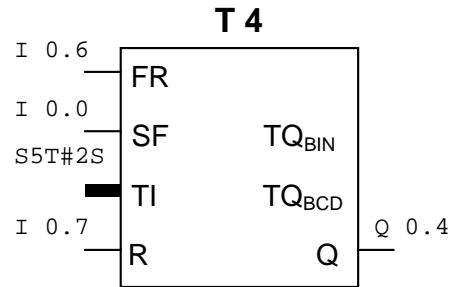
```
A I 0.6
FR T 3
A I 0.0
L S5T#2S
SS T 3
A I 0.7
R T 3
A T 3
= Q 0.3
```



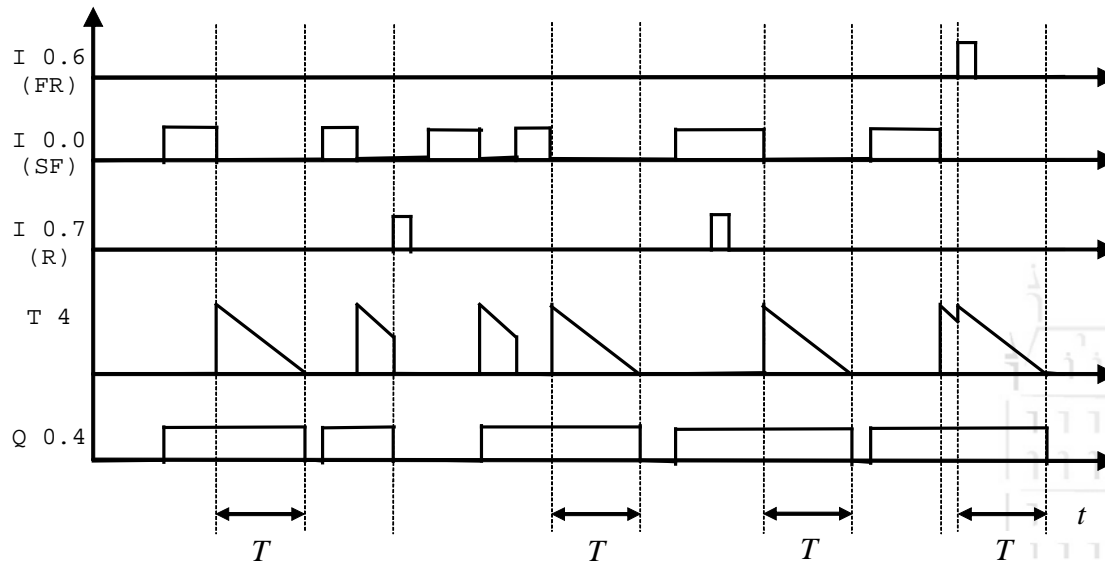


STEP 7 – Časovnik

Zakasnitev izklopa ($T = 2$ s) / čas steče po izklopu



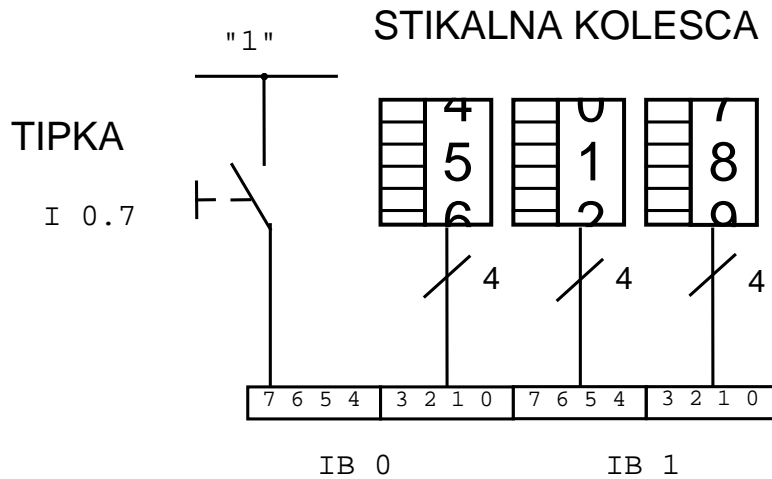
```
A I 0.6
FR T 4
A I 0.0
L S5T#2S
SF T 4
A I 0.7
R T 4
A T 4
= Q 0.4
```



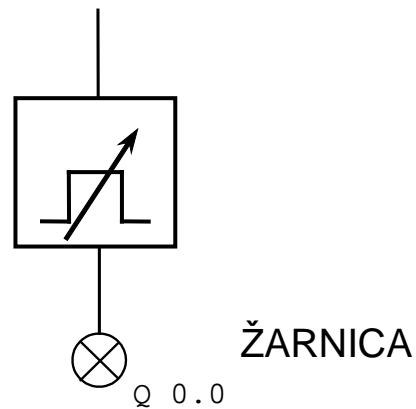


STEP 7 – Časovnik

Zgled



Slika 7.13: Shema za nastavljanje časa gorenja žarnice





STEP 7 – Časovnik

Zgled

```
L   IW 0           // v spodnjo besedo AKU1 vsebina z IW 0
AW  W#16#0FFF     // maska IN: pomembni so spodnji trije nibbli
OW  W#16#2000     // rezultatu "dodamo" časovno bazo (maska ALI)
A   I 0.7         // ob pritisku na tipko
SE  T 0           // steče časovnik z nastavljenno čas. konstanto
A   T 0           // povpraševanje po časovniku: če čas teče...
=   Q 0.0         // ... naj gori žarnica
```





STEP 7 - Statusna beseda in skočni ukazi

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC

Slika 7.14: Statusna beseda (STW) pri CPU





STEP 7 - Statusna beseda in skočni ukazi

Ukaz	OR	STA	RLO	/FC
A	x	x	x	1
AN	x	x	x	1
A(0	1	-	0
AN(0	1	-	0
O	0	x	x	1
ON	0	x	x	1
O(0	1	-	0
ON(0	1	-	0
X	0	x	x	1
XN	0	x	x	1
X(0	1	-	0
XN(0	1	-	0
=	0	x	-	0
CLR	0	0	0	0
FN	0	x	x	1
FP	0	x	x	1
NOT	-	1	x	-
R	0	x	-	0
S	0	x	-	0
SAVE	-	-	-	-
SET	0	1	1	0

Tabela 7.7: Vpliv nekaterih logično-povpraševalnih operacij na bite v statusni besedi





CC0	CC1	Pomen (splošno)
0	0	Rezultat operacije je enak nič (= 0)
0	1	Rezultat operacije je pozitiven (> 0)
1	0	Rezultat operacije je negativen (< 0)
1	1	Rezultat operacije je neveljaven

Tabela 7.8: Pomen pogojnih kod CC0 in CC1

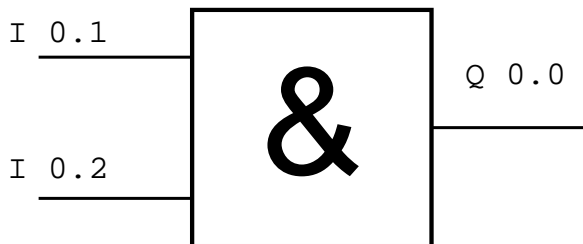
Ukaz		Stanje pogojnih kod		Pomen povpraševanja oz. kdaj se izvrši skok
povpraševalni	skočni	CC0	CC1	
A >0	JP	0	1	Pozitivno število (<i>Plus</i>)
A >=0	JPZ	0	x	Pozitivno število ali 0 (<i>Plus or Zero</i>)
A <0	JM	1	0	Negativno število (<i>Minus</i>)
A <=0	JMZ	x	0	Negativno število ali 0 (<i>Minus or Zero</i>)
A <>0	JN	CC0 ≠ CC1		Število, različno od 0 (<i>Non-zero</i>)
A ==0	JZ	0	0	Število 0 (<i>Zero</i>)
A UO	JUO	1	1	Neveljavno število UO (<i>UnOrdered</i>)

Tabela 7.9: Skočne in povpraševalne operacije glede na CC0 in CC1

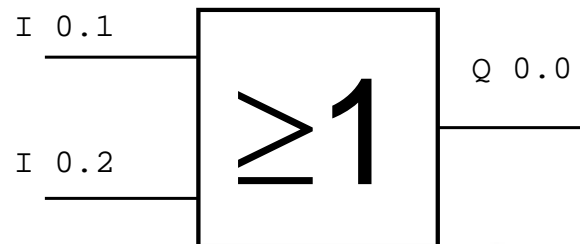


STEP 7 - Statusna beseda in skočni ukazi

I 0.0 = 0 :

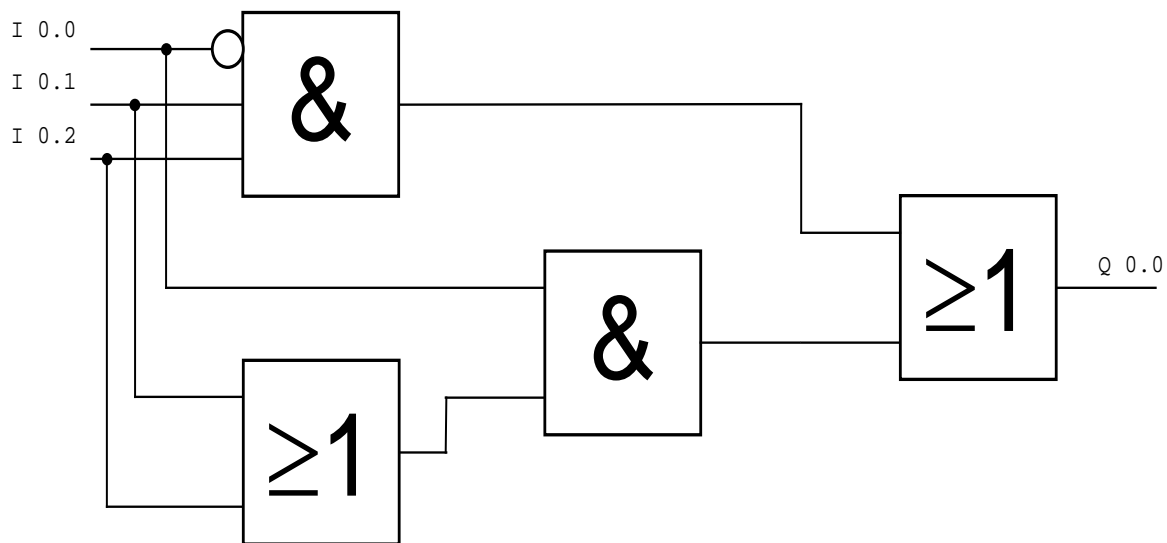


I 0.0 = 1 :



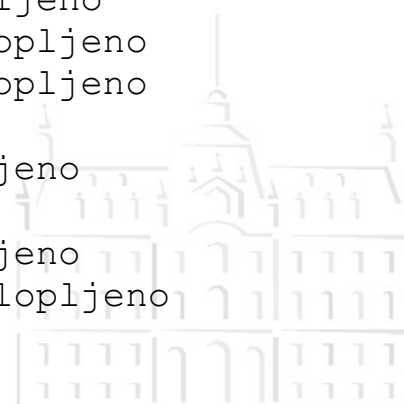


STEP 7 - Statusna beseda in skočni ukazi



Slika 7.15: Funkcijski načrt za rešitev brez skočnih ukazov

```
AN I 0.0 // če je I 0.0 izklopljeno
A I 0.1 // IN če je I 0.1 vklopljeno
A I 0.2 // IN če je I 0.2 vklopljeno
O // ALI
A I 0.0 // če je I 0.0 vklopljeno
A( // in
O I 0.1 // če je I 0.1 vklopljeno
O I 0.2 // ALI če je I 0.2 vklopljeno
)
= Q 0.0 // naj žarnica sveti
```





```
A    I 0.0    // če je I 0.0 vklopljeno
JC   sko1    // pogojno skoči na operacijo ALI
      // sicer nadaljuj...
A    I 0.1    // če je I 0.1 vklopljeno
A    I 0.2    // IN če je I 0.2 vklopljeno
=    Q 0.0    // naj žarnica sveti
BEU                                     // POZOR: ne sme nadaljevati!
```

```
sko1: O    I 0.1    // če je I 0.1 vklopljeno
      O    I 0.2    // ALI če je I 0.2 vklopljeno
      =    Q 0.0    // naj žarnica sveti
      BE
```





STEP 7 - Statusna beseda in skočni ukazi

```
A    I 1.0
FP   M 0.0    // dinamični člen za tipko I 1.0
=    M 0.1    // priredimo M 0.1 za nadaljnjo uporabo

A    M 0.1    // če je na I 1.0 sprememba z "0" na "1"
AN   M 20.0   // in če je pomnilnik na "0"
S    M 20.0   // setiraj pomnilnik
JC   SK01    // ob RLO = "1" preskoči resetiranje!!!

A    M 0.1    // če je na I 1.0 sprememba z "0" na "1"
A    M 20.0   // in če je pomnilnik na "1"
R    M 20.0   // potem resetiraj pomnilnik

SK01: A    M 20.0 // če želimo prenos v dvojiški obliki
JC   SK02    // pogojno skoči na ukaz L
      // sicer nadaljuj...
LC   C 6     // naloži vsebino števca v BCD obliki
```





STEP 7 - Statusna beseda in skočni ukazi

Ukaz	Opis	Pogoj za skok	Vpliv na STW								
			BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
JU	brezpogojni skok	–	–	–	–	–	–	–	–	–	–
JC	pogojni skok, če je RLO 1	RLO = 1	–	–	–	–	–	0	1	1	0
JCN	pogojni skok, če je RLO 0	RLO = 0	–	–	–	–	–	0	1	1	0
JCB	pogojni skok, če je RLO 1; vedno shrani še RLO na BR	RLO = 1	x	–	–	–	–	0	1	1	0
JNB	pogojni skok, če je RLO 0; vedno shrani še RLO na BR	RLO = 0	x	–	–	–	–	0	1	1	0
JBI	pogojni skok, če je BR 1	BR = 1	–	–	–	–	–	0	1	–	0
JNBI	pogojni skok, če je BR 0	BR = 0	–	–	–	–	–	0	1	–	0
JZ	pogojni skok, če je rezultat operacije enak 0	CC0 = 0 CC1 = 0	–	–	–	–	–	–	–	–	–
JN	pogojni skok, če je rezultat operacije različen od 0	CC0 ≠ CC1	–	–	–	–	–	–	–	–	–
JP	pogojni skok, če je rezultat operacije pozitiven	CC0 = 0 CC1 = 1	–	–	–	–	–	–	–	–	–
JPZ	pogojni skok, če je rezultat operacije pozitiven ali enak 0	CC0 = 0 CC1 = x	–	–	–	–	–	–	–	–	–
JM	pogojni skok, če je rezultat operacije negativen	CC0 = 1 CC1 = 0	–	–	–	–	–	–	–	–	–
JMZ	pogojni skok, če je rezultat operacije negativen ali enak 0	CC0 = x CC1 = 0	–	–	–	–	–	–	–	–	–
JUO	pogojni skok, če je rezultat operacije neveljaven	CC0 = 1 CC1 = 1	–	–	–	–	–	–	–	–	–
JO	pogojni skok, če je pri operaciji prišlo do preliva	OV = 1	–	–	–	–	–	–	–	–	–
JOS	pogojni skok, če je bit shranjenega preliva enak 1	OS = 1	–	–	–	–	–	–	–	–	–
JL	izbirni skok iz seznama**	–	–	–	–	–	–	–	–	–	–
LOOP	zančni skok***	–	–	–	–	–	–	–	–	–	–

Tabela 7.10: Pregled vseh skočnih ukazov





Zgled skočnega ukaza JL

```
L      +2      // želimo opraviti 2. skok iz seznama
JL    NAPR    // NAPR označuje konec seznama skokov
JU    SK00    // 0. skok
JU    SK01    // 1. skok
JU    SK02    // 2. skok
JU    SK03    // 3. skok
JU    SK04    // 4. skok
      ...
NAPR: ...     // konec seznama skokov; nadaljnji ukazi
      ...
      ...
SK00: ...     // ukazi za 0. skok
      ...
      ...
SK01: ...     // ukazi za 1. skok
      ...
      ...
```





Zgled skočnega ukaza LOOP

```
L      +15      // zanko želimo ponoviti 15-krat
ZANK  T      MW 20 // število ponovitev shrani na MW 20
...    // ukazi, ki se bodo ponavljali
...
...
L      MW 20    // naloži število ponovitev v AKU1
LOOP  ZANK     // zmanjšaj št. ponovitev in ponovi
...    // nadaljnji ukazi
```





Zgled klica blokov

OB 1:

```
UC   FB 1           // brezpogojno kliči FB 1
A    I 1.0         // če je vklopljeno stikalo I 1.0
CC   FB 2         // kliči FB 2
BE
```

FB 1:

```
A    I 0.0         // če je I 0.0 vklopljeno
JC   skol         // pogojno skoči na operacijo ALI
                        // sicer nadaljuj...
A    I 0.1         // IN če je I 0.1 vklopljeno
A    I 0.2         // IN če je I 0.2 vklopljeno
=    Q 0.0         // naj žarnica sveti
BEU                          // POZOR: ne sme nadaljevati!
```

```
skol: O   I 0.1         // če je I 0.1 vklopljeno
      O   I 0.2         // ALI če je I 0.2 vklopljeno
      =   Q 0.0         // naj žarnica sveti
      BE
```

FB 2:

```
A    Q 0.0         // če žarnica sveti
BEC                          // končaj z izvajanjem bloka FB 2
                        // sicer nadaljuj...
L    W#16#FFFF
T    QW 4          // prižgi vse lučke na QW 4
BE                          // konec bloka
```



STEP 7 – Naslavljanje

Takojsnje naslavljanje (angl. immediate addressing)

Primer	Pomen
OW W#16#ABCD	Besedo v ACCU1 poveži s številom ABCD _{HEX} po logični funkciji ALI
L 123	Naloži celoštevilčno vrednost 123 v ACCU1
L 'ABCD'	Naloži ASCII niz ABCD v ACCU1
L B#(12,13)	V spodnja zloga ACCU1 naloži vrednosti 12 in 13
L C#0100	Naloži BCD vrednost v ACCU1

Neposredno naslavljanje (angl. direct addressing)

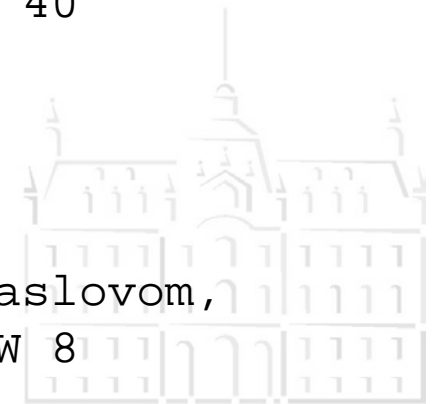
Primer	Pomen
A I 10.0	In povezava z vhodnim bitom 10.0
= Q 11.0	Dodeljevanje RLO izhodnem bitu 11.0
L IW10	Vhodno besedo 10 naloži v ACCU1
T QW11	Spodnjo besedo iz ACCU1 pošlji na izhodno besedo 11



STEP 7 – Posredno naslavljanje

MW 20: 2#0000_0000_0010_1001 = 16#0029 = 41

```
...  
CD C[MW 20] // štetje navzdol števca, katerega naslov...  
// ...je definiran z vsebino MW 20  
...  
...  
A I 0.1  
SE T[MW 20] // podaljšanje pulza na časovniku z naslovom,  
// ... definiranim z vsebino MW 40  
...  
...  
A M 14.1  
CC FB[MW 20] // klic funkcijskega bloka z naslovom,  
// ... definiranim z vsebino MW 8
```





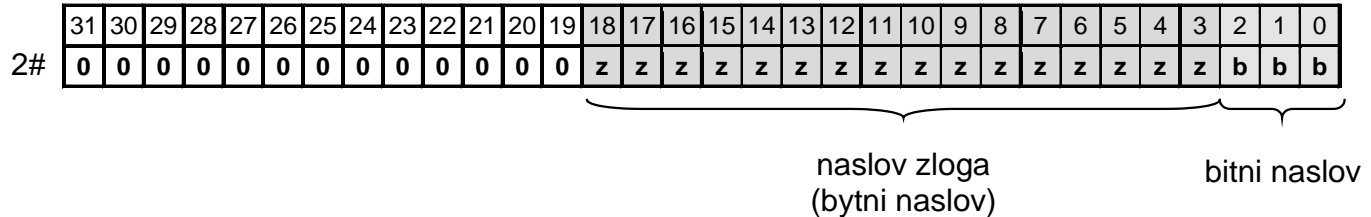
STEP 7 – Posredno naslavljanje

I 1.6

Q 124.2

M 16.0

P# kazalčni format



Slika 7.16: Format 32-bitnega kazalca (pointerja)

P#1.6 = 16#0000_0000_0000_000E =
 2#0000_0000_0000_0000_0000_0000_0000_1110

P#124.2 = 16#0000_0000_0000_03E2 =
 2#0000_0000_0000_0000_0000_0011_1110_0010

P#16.0 = 16#0000_0000_0000_0080 =
 2#0000_0000_0000_0000_0000_0000_1000_0000

I [MD 40]

Q [MD 44]

M [MD 48]

