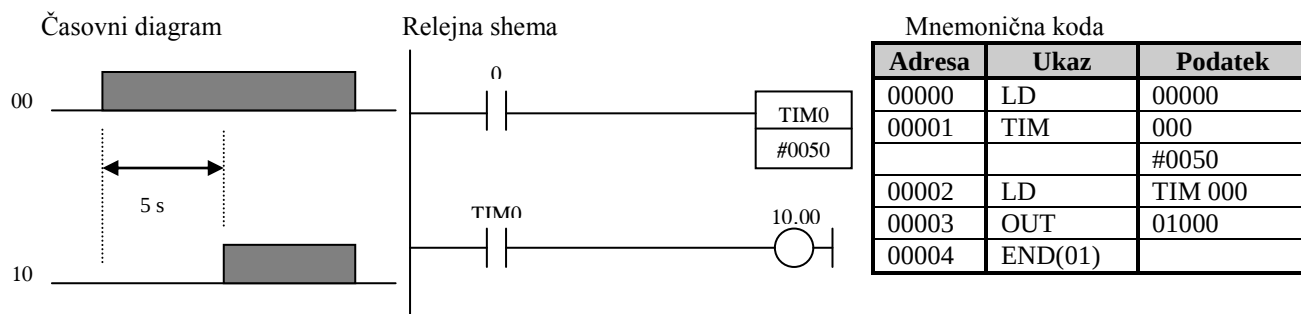


1.2. Časovne instrukcije - TIM

Časovnike uporabljamo za zakasnitvene aplikacije, pri čemer imamo več možnosti: zakasnitev vklopa, zakasnitev izklopa, impulzno delovanje,... TIM je odštevalna vklopna časovna instrukcija, v katero moramo vnesti številko časovnika (TIM0) in nastavitveno vrednost SV (set value) v območju od 0000 do 9999, kar pomeni časovno 0 do 999.9s. Vrednosti časovnikov nastavljamo v desetinkah sekunde. Nekateri PLC imajo možnost definiranja hitrih časovnikov (TIMH), katerih vrednosti pa lahko postavljamo v stotinkah sekunde.

Zakasnitev vklopa: On Delay



Zakasnitev vklopa in izklopa: On and Off Delay

Aplikacijo uporabimo za izvedbo zakasnitve ON/OFF danega vhodnega signala za določen in nastavljen čas.

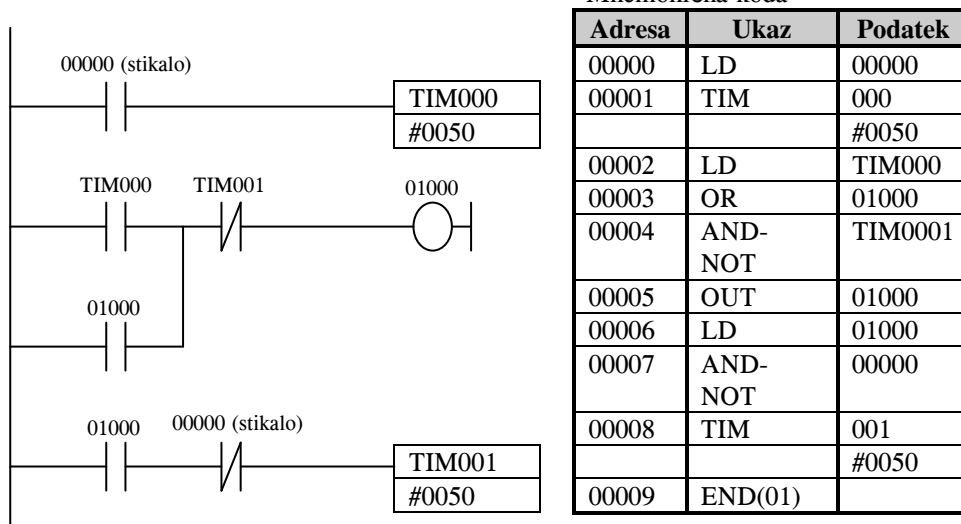
I/O tabela

Vhod	Enota	Izhod	Enota
00000	Stikalo	01000	Motor

Časovni diagram



Relejna shema



Primer: Krmiljenje tračnega transporterja

PLC bomo uporabili za krmiljenje vklopljanja in izklopljanja segmentnega tračnega transporterja. Vsak segment ima svoj pogon (motor z reduktorjem). Tračni transporter je namenjen transportu predmetov (bakrena plošča). Pozicioniranje plošče izvedemo z brezkontaktnimi senzorji (induktivna stikala), ki jih namestimo ob koncu vsakega transportnega segmenta. Pogon segmenta je aktiven, dokler je plošča v zaznavnem dosegu stikala. V primeru, ko pride plošča izven dosega senzorja, se aktivira časovnik. Ko poteče njegov nastavljeni čas, se bo pogon tega segmenta ustavil.

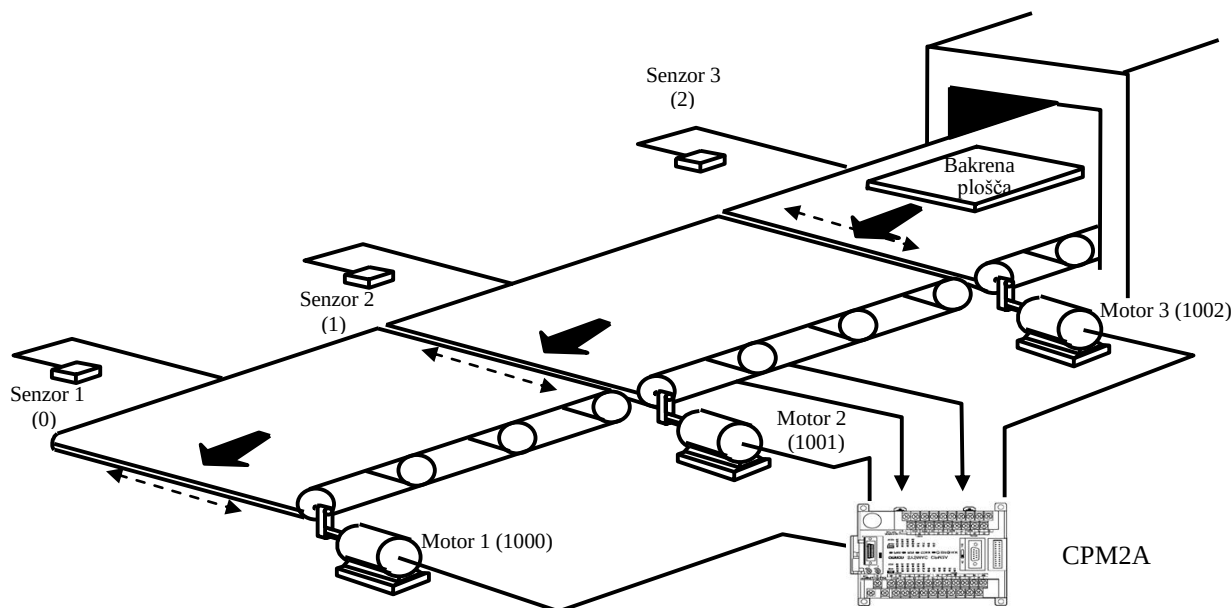


Tabela vhodnih in izhodnih signalov – adresni plan

Vhodi	Opis
00000	Senzor S1
00001	Senzor S2
00002	Senzor S3

Izhodi	Opis
01000	Motor M1
01001	Motor M2
01002	Motor M3

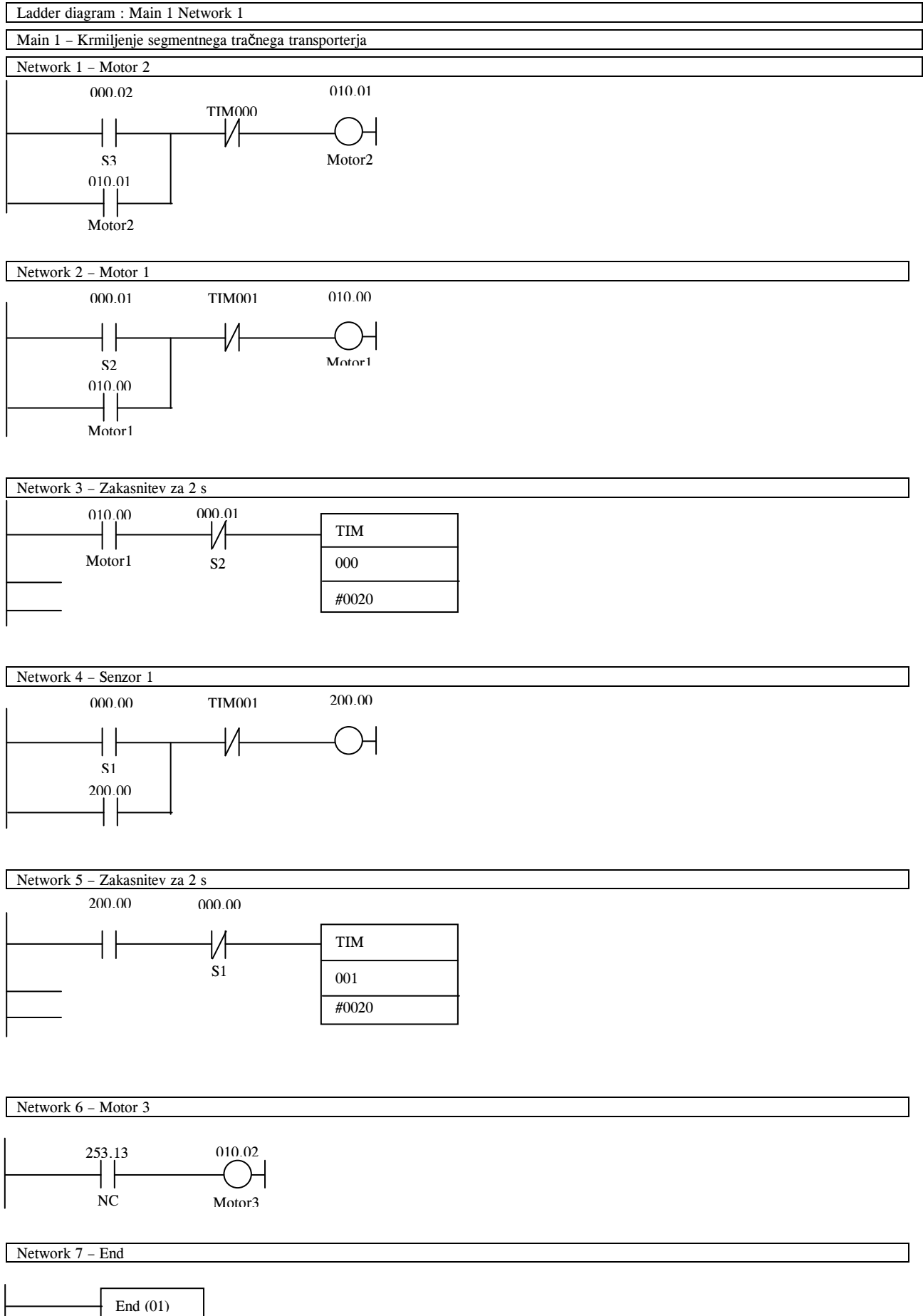
Opisni algoritem delovanja

1. Motor M3 je vklopljen ves čas (pogoj je vklopljeno krmilje ali krmilnik)
2. Motor M2 se vklopi, ko senzor S3 zazna ploščo
3. Motor M2 je vklopljen, dokler je vklopljen motor M1 in je plošča izven zaznavnega območja senzorja S2
4. Motor M1 se vklopi, ko senzor S2 zazna motor
5. Motor M1 je vklopljen, dokler plošča ni izven zaznavnega območja senzorja S1

Logične enačbe za kompenzacijski način delovanja segmentnega transporterja

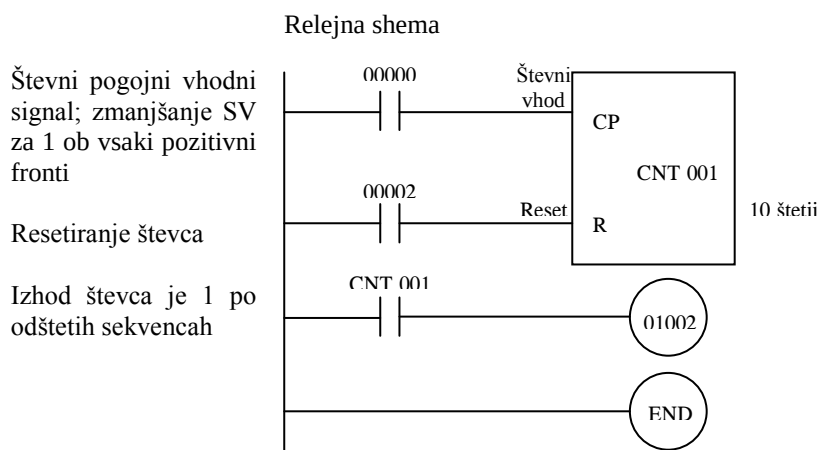
1. Vklop bit-M_n: $\text{bit-M}_n = (\text{Prehod}_{n-1} \cdot S_{n-1}) + (\text{Prehod}_n \cdot S_n)$
2. Izklop bit-M_n: $\overline{\text{bit-M}_n} = (\text{Prehod}_{n-1} \cdot \overline{\text{Prehod}_n} \cdot S_n) + (\text{Prehod}_n \cdot S_{n+1})$
3. $\overline{\text{Prehod}_{n-1}} = \overline{\text{Prehod}_n} \cdot S_{n-1} \cdot \overline{S_n}$; $\overline{\text{Prehod}_{n-1}} = \overline{\text{Prehod}_{n-1}} \cdot S_n$
4. $\overline{\text{Prehod}_n} = \overline{\text{Prehod}_{n-1}} \cdot S_n \cdot \overline{S_{n+1}}$; $\overline{\text{Prehod}_n} = \overline{\text{Prehod}_n} \cdot S_{n+1}$
5. Krmiljenje pogona (ročno/avtomatsko): $M_n = R \cdot S_{n-R} + A \cdot \text{bit-M}_n$

Relejna shema – lestvični diagram



1.3. Števec – CNT, CNTR

CNT je nastavljiva odštevalna instrukcija, kar pomeni, da vsakokrat, ko gre vhodni pogoj števca iz OFF na ON, se njegova vrednost odšteva za 1. Števec programiramo s števnim vhodom, reset vhodom, številko števca in vrednostjo števca SV, ki lahko zavzame vrednosti od 0000 do 9999.

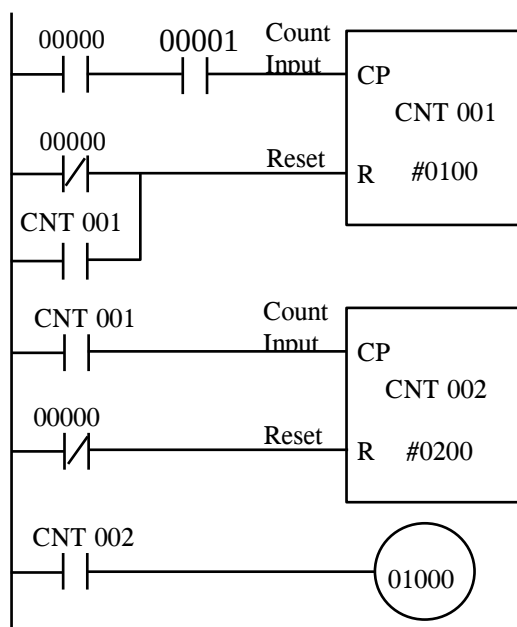


Mnemonična koda

Adresa	Ukaz	Podatki
0000	LD	00000
0001	LD	00002
0002	CNT	001
		#0010
0003	LD	CNT 001
0004	OUT	01002
0005	END(01)	

Številka števca se ne sme podvajati s številko časovnika, ker bi tako oba lahko koristila isti pomnilni prostor PLC. Trenutno vrednost števca (časovnika) lahko spremljamo preko PC ali programirne konzole.

Primer 1: Števec do vrednosti 20.000



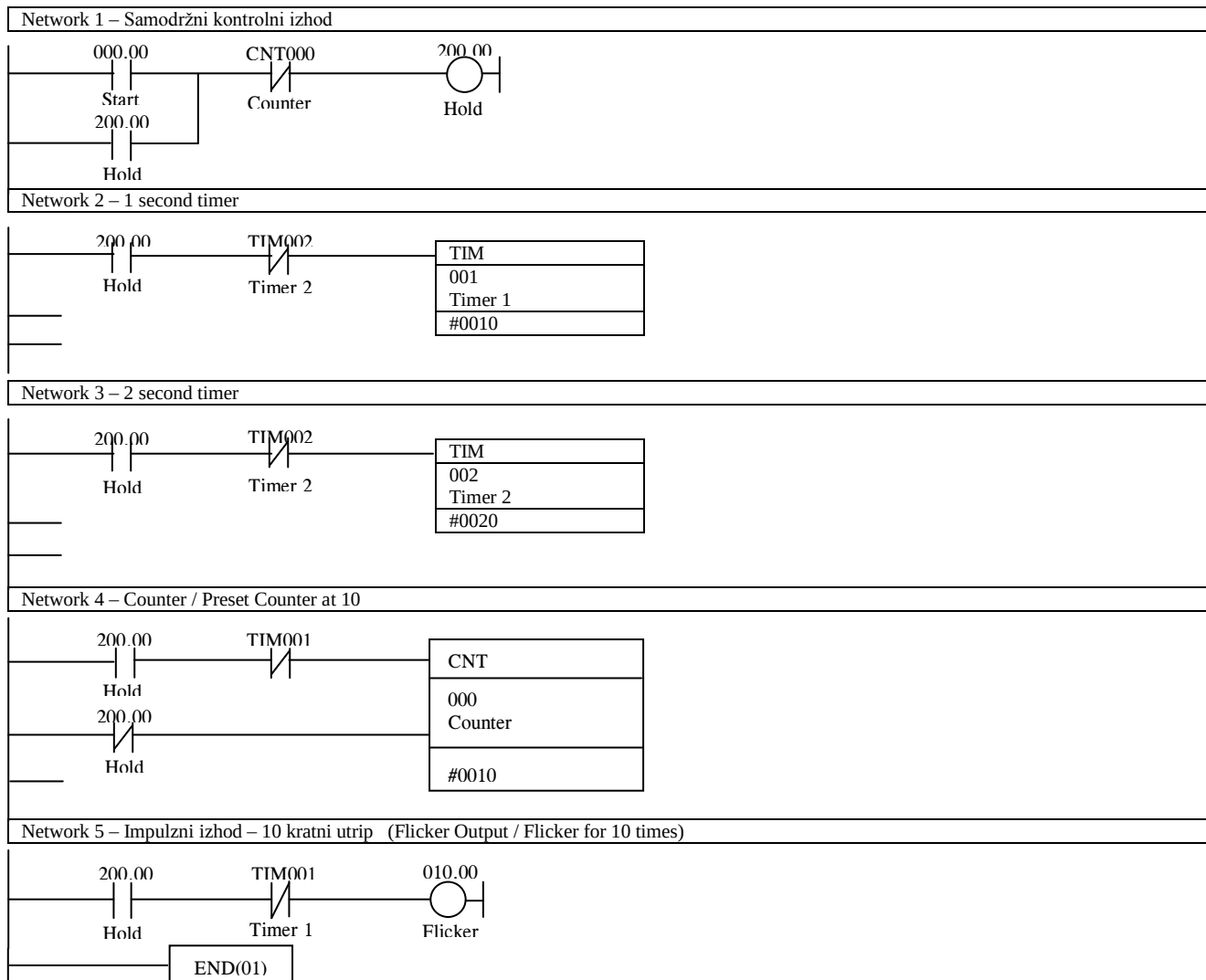
Adresa	Ukaz	Podatki
00000	LD	00000
00001	AND	00001
00002	LD-NOT	TIM 000
00003	OR	CNT 001
00004	CNT	001
		#0100
00005	LD	CNT001
00006	LD-NOT	00000
00007	CNT	002
		#0200
00008	LD	CNT002
00009	OUT	01000
00010	END(01)	

Primer 2: Povečanje časovnega obsega na 1.000 ur

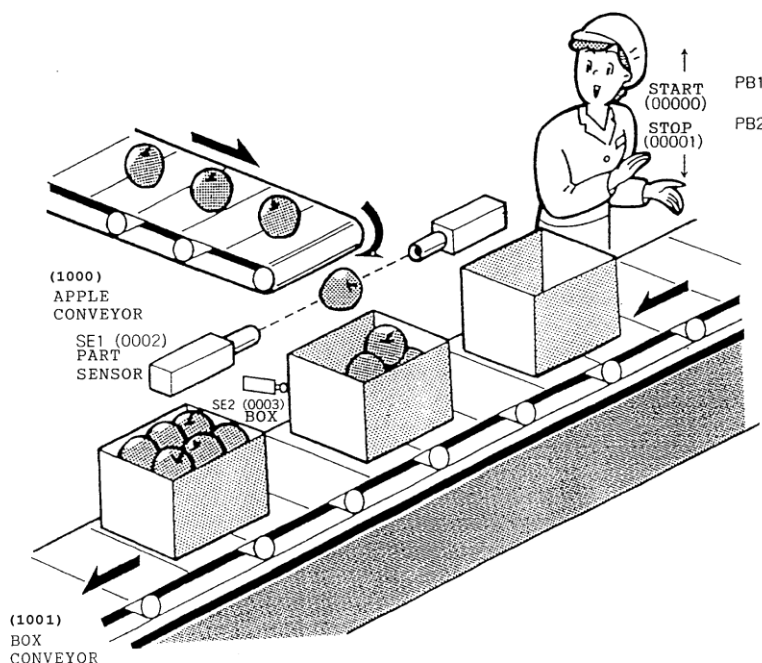
Adresa	Ukazi	Podatki
00000	LD	00000
00001	AND-NOT	TIM 001
00002	TIM	001
		#6000
00003	LD	TIM001
00004	LD	00001
00005	CNT	002
		#6000
00006	LD	CNT 002
00007	OUT	01000

Primer 3:

Impulzni izhod – Izhod 010.00 impulzno deluje (1s ON, 1s OFF) 10 krat po vklopu vhodnega signala 000.00



Primer 4: Krmiljenje pakirne linije



Opisni algoritem sekvenčnega delovanja v korakih:

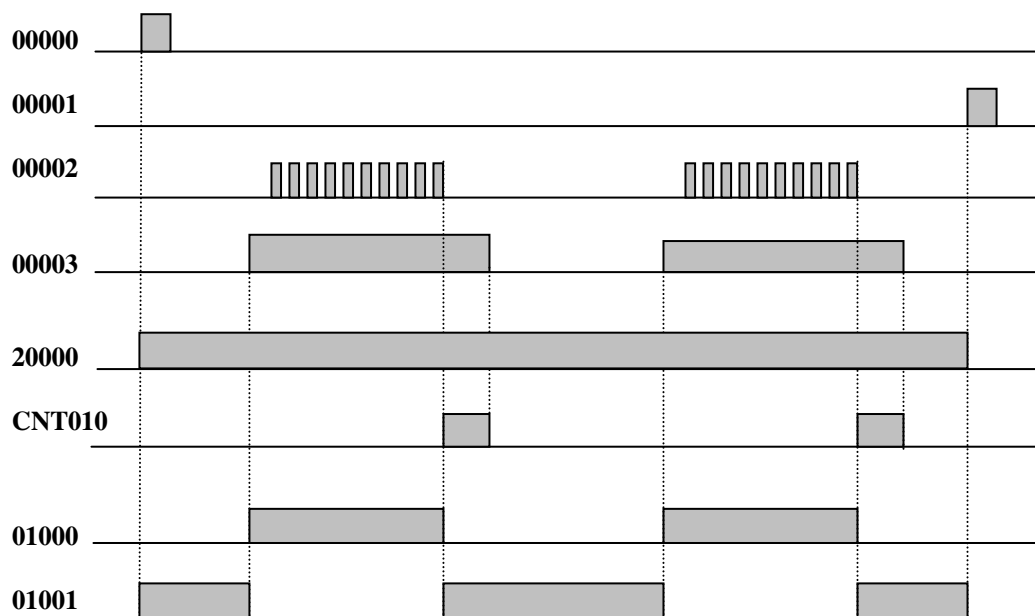
1. s pritiskom na tipko PB1 (start) startamo transporter embalaže
2. Ob aktiviranju SE2 (škatla na poziciji) se transporter embalaže ustavi
3. start transporterja jabolk, senzor SE1 (štetje jabolk) prešteje 10 komadov
4. po vložnih 10 komadih se transporter jabolk usatvi in ponovno starta transporter embalaže
5. števec se resetira in operacija se ponavlja tako dolgo, dokler ne aktiviramo tipke stop (PB2)

Popis vhodov in izhodov:

Vhodi	Opis
00000	START Tipka (PB1)
00001	STOP Tipka (PB2)
00002	Detekcija jabolk (SE1)
00003	Detekcija škatle (SE2)

Izhodi	Opis
01000	Transporter jabolk - A
01001	Transporter embalaže - B

Časovni diagram delovanja:



Zapis aplikacijskega programa v mnemonični kodi

Address	Instruction	Data
0000	LD	00000
0001	OR	20000
0002	AND NOT	00001
0003	OUT	20000
0004	LD	20000
0005	AND NOT	01001
0006	OUT	01000
0007	LD	00002

Address	instruction	Data
0008	LD NOT	00003
0009	CNT	010
		#0010
0010	LD CNT	010
0011	OR NOT	00003
0012	AND	20000
0013	OUT	01001
0014	END (01)	

2. Osnovne izhodne sekvenčne instrukcije

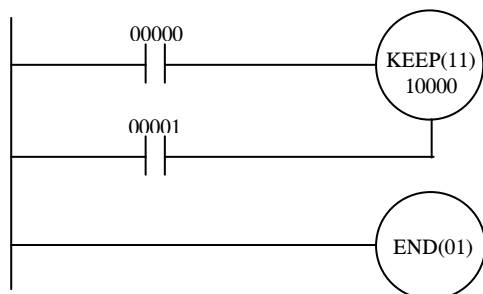
Izhodne instrukcije lahko definirajo izhodni bit ali kanal v programu samo enkrat. Kot vhodni podatek pa seveda lahko nastopi tudi večkrat. Brisanje izhodnega stanja ima prioriteto pred postavljanjem, zato bi lahko pri morebitnem večkratnem postavljanju izhodnih stanj (naslovov) prišlo do napačnega in nelogičnega delovanja programa.

- | | |
|----------------|--|
| 1. OUT | običajni logični izhod (OUTPUT), pri čemer je aktiven tako dolgo, dokler so prisotni vsi vhodni pogoji |
| OUT NOT | negiran logični izhod OUT |
| 2. SET | forsirana izhodna set logična funkcija, ki postavi bit na ON ob prisotnosti vseh vhodnih pogojev |
| RESET | forsirana reset logična funkcija, ki bo brisala postavljeni izhodni bit |

3. KEEP – Latch rele

Keep funkcijski blok uporabimo kot bistabilni element (latch). Ima dva krmilna vhoda, prvi za postavljanje izhodnega bita ali kanala, drugi pa za brisanje le tega. Če keep funkcijo uporabimo skupaj s HR pomnilno lokacijo, se bo izhodno stanje latched ohranilo tudi ob prekinitvi napajalne napetosti krmilnika!

Primer relejne sheme in mnemonične kode z uporabo funkcije KEEP:



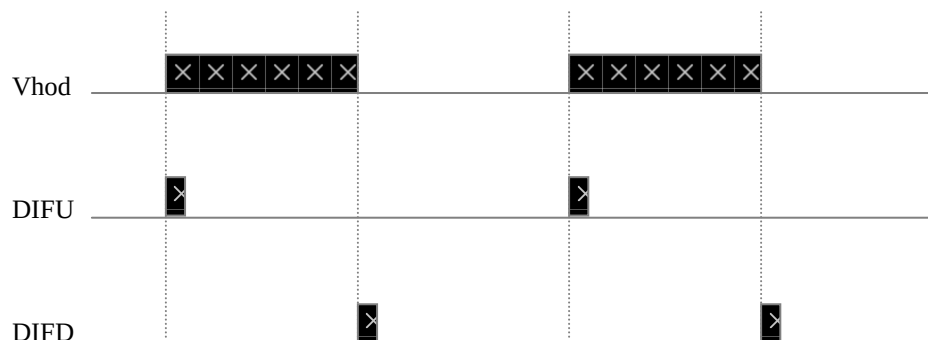
Programski naslov	Instrukcija	Podatek
0000	LD	00000
0001	LD	00001
0002	KEEP(11)	10000
0003	END(01)	

4. DIFU (Differentiate up) in DIFD (Differentiate down) – Instrukciji diferenciranja

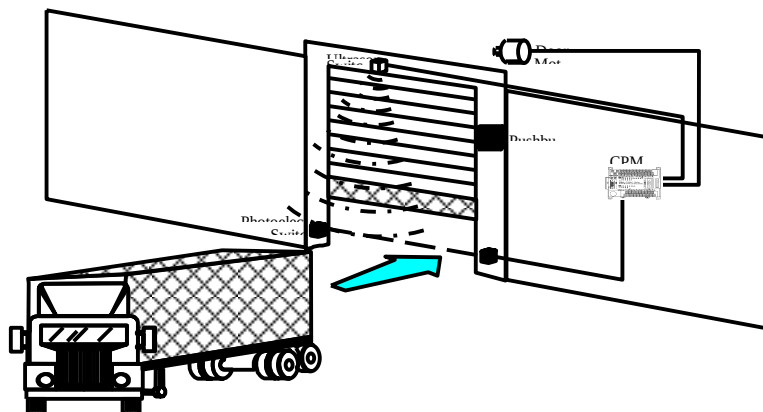
S pomočjo teh dveh instrukcij lahko na dokaj enostaven način opišemo funkcijo gibanja in spreminjanja vhodnih pogojev. Značilnosti instrukcij so:

- DIFU in DIFD vklopi določeni izhod na ON, vendar samo za čas enega scan-a
- DIFU vklopi določeni izhod ON pri detekciji pozitivne fronte (pri vklopu pogoja) na pogojnem vhodnem signalu
- DIFD vklopi določeni izhod ON pri detekciji negativne fronte (pri izklopu pogoja) na pogojnem vhodnem signalu

Časovni prikaz delovanja:



Primer 1: Avtomatsko krmiljenje vrat skladišča



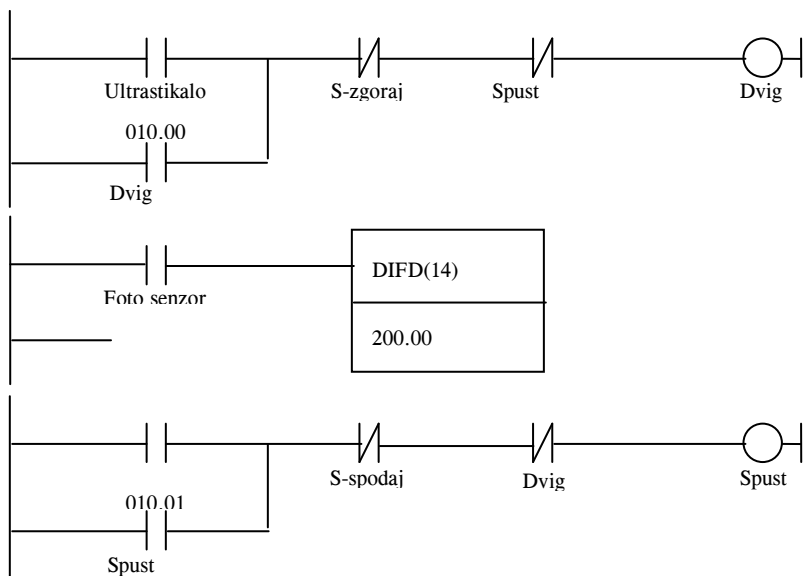
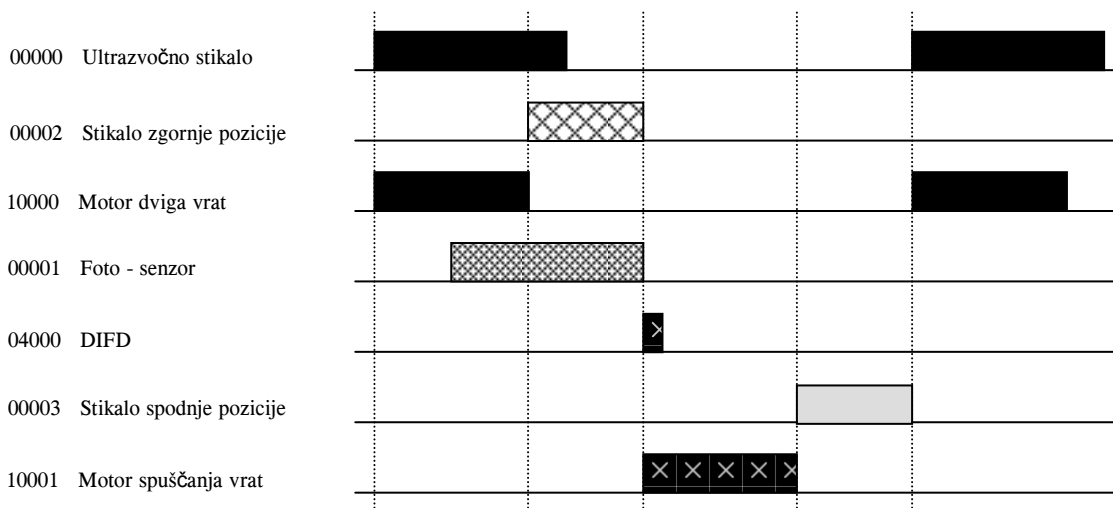
Vhodni ultrazvočni senzor je namenjen detekciji prisotnosti vlačilca. Foto senzor (sprejemnik – oddajnik) zaznava prehod vlačilca. V odvisnosti od teh dveh signalov krmilimo pogon za odpiranje (dvig) in zapiranje (spust) vrat.

Adresni plan

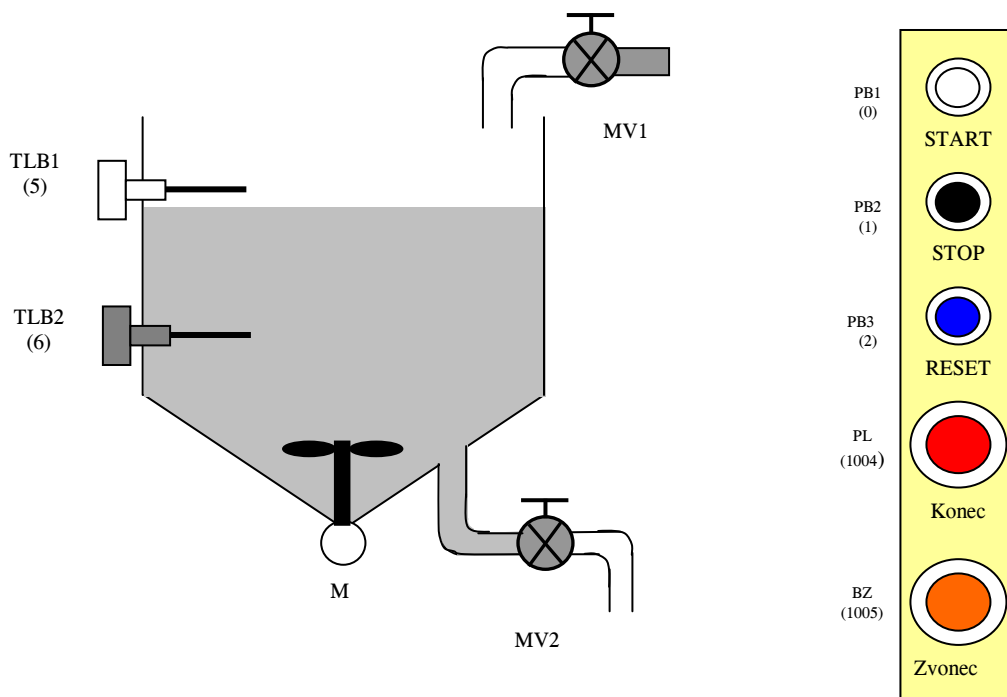
Vhodi	Opis
00000	Ultrasenzor
00001	Fotosenzor
00002	Končno stikalo-vrata zgoraj
00003	Končno stikalo-vrata spodaj

Izhodi	Opis
01000	Pogon – dvig vrat
01001	Pogon – spust vrat

Časovni diagram



Primer 2: Krmiljenje polnjenja in praznjenja rezervoarja



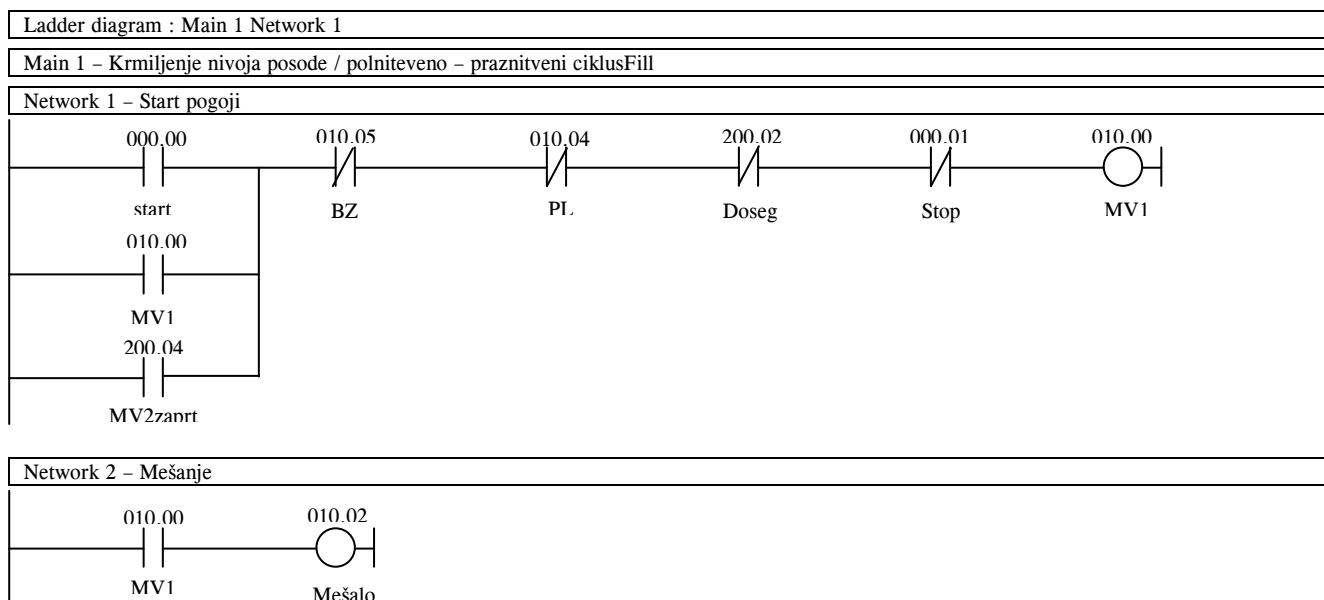
Vhodno – izhodna tabela:

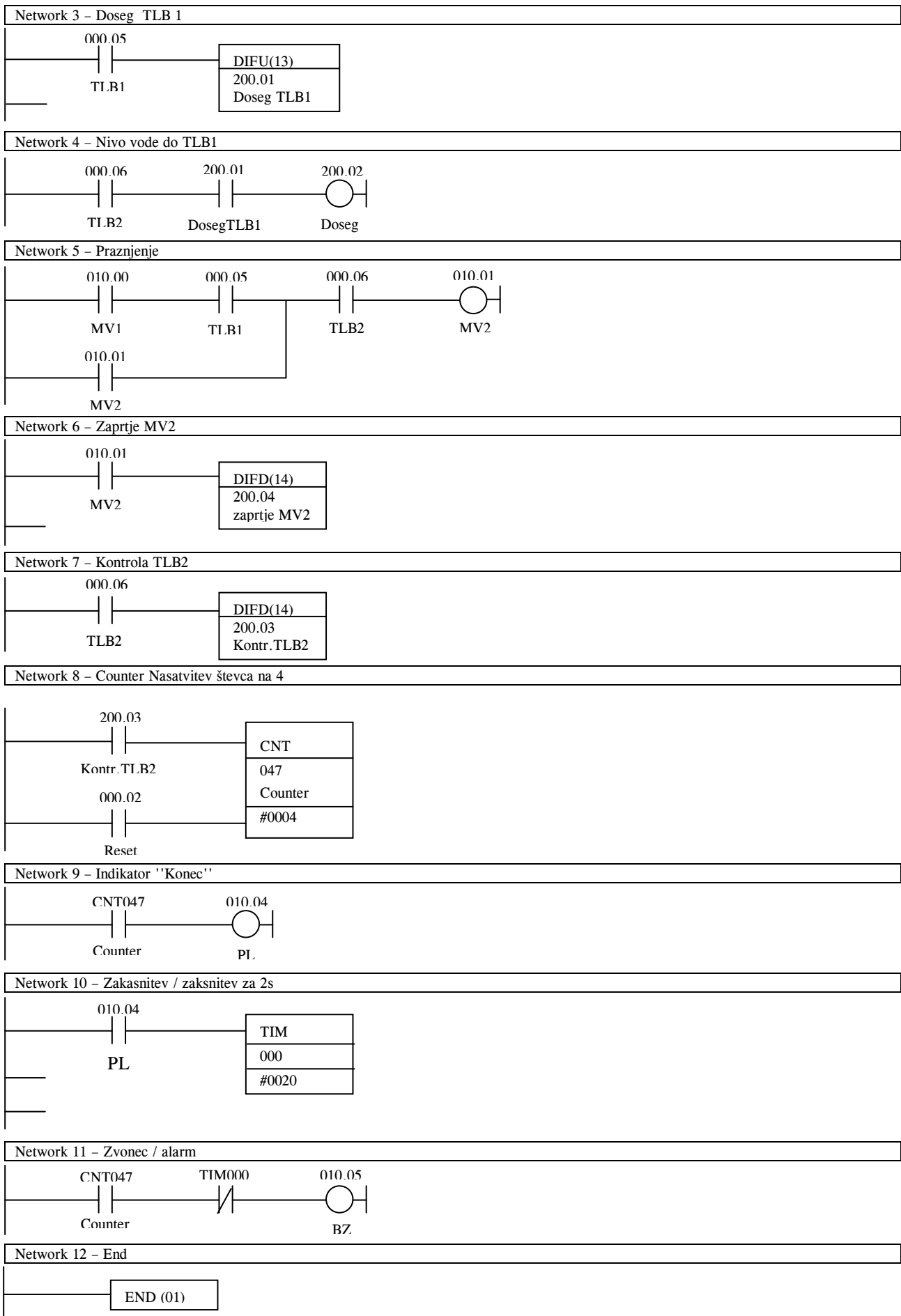
Vhod	Opis
00000	Start tipka (PB1)
00001	Stop tipka (PB2)
00002	Reset tipka (PB3)
00005	Stikalo – zgornji nivo (TLB1)
00006	Stikalo – spodnji nivo (TLB2)

Izhod	Opis
01000	Ventil – dotok vode (MV1)
01001	Ventil – izpust vode (MV2)
01002	Mešalni motor (M)
01004	Indikator – Konec
01005	Zvonec

Opisni algoritem delovanja:

1. Ob pritisku na PB1, se MV1 odpre in voda prične polniti cisterno; vklop mešanega motorja M
2. Ko nivo vode doseže TLB1 in TLB2, se MV1 zapre in mešalni motor se ustavi
3. MV2 se odpre in voda prične odtekati, ko nivo vode pade pod TLB2, se MV2 zapre
4. Ko se cikel operacije 4 krat ponovi, vklopimo indikator Konec in polnilno-spraznitvena operacija se ne ponovi več; tudi v primeru aktiviranja PB1 ne.

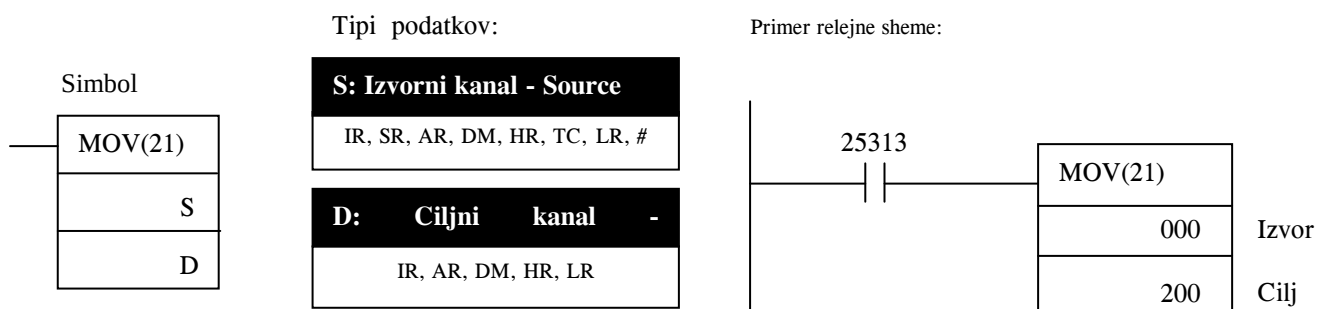




3. Ukazi za obdelavo podatkov

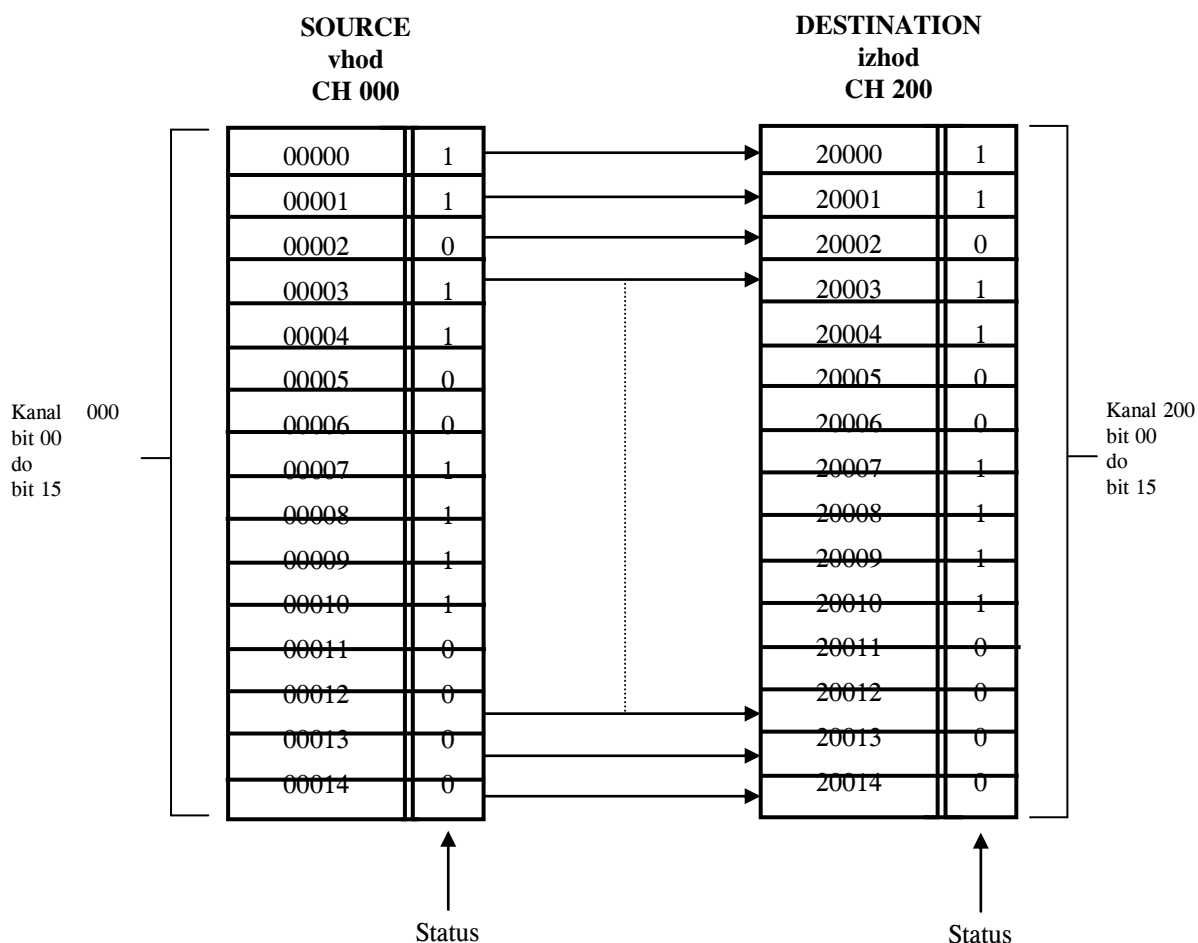
3.1. Prenos podatkov - MOV

MOV instrukcija prenese izvorni podatek (podatek določenega kanala ali štrbitno heksadecimalno konstanto) v ciljni kanal oz. register. Ukaz je definiran z dvema parametroma: izvorni kanal oz. konstanta in ciljni kanal. Instrukcijo uporabljamo pri inicializaciji krmilnikovega pomnilnika, matematičnih in primerjalnih operacijah,...



Prikaz delovanja operacije MOV:

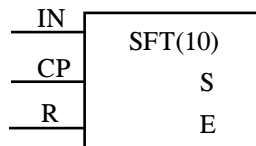
podatek na vhodnem kanalu 000 prenesemo na izhodni kanal 200



3.2. Pomikalni register – SFT

Ukaz SFT pomika 16-bitni podatek določenega kanala za en bit. Startni in končni kanal morata biti specificirana kot podatek. Pomikanje je lahko izvedeno v levo ali desno smer ali pa rotirajoče. Pomikalni register je možno izvesti tudi v asinhronskem delovanju. Ukaz je univerzalnega tipa in ga lahko uporabimo pri štetju, zaporednem sekvenčnem delovanju,...

Simbol:

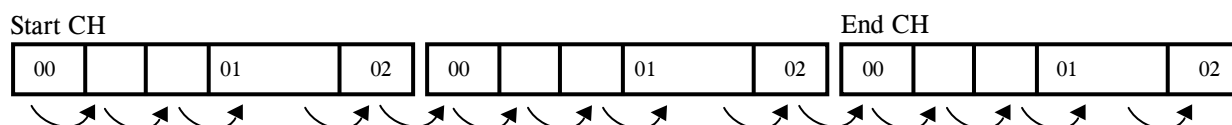


Tipi podatkov:

I/O, Internal auxiliary Relay IR
Holding Relay HR

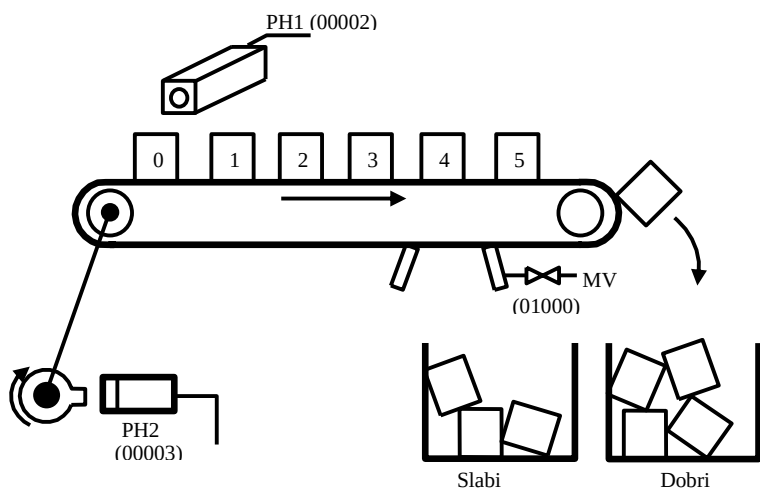
V primeru aktiviranja Reset vhoda, se resetira vseh 16 bitov. V primeru uporabe strukture HR, se ob izpadu napajalne napetosti podatki ohranijo.

Tehnologija delovanja pomikanja v desno:



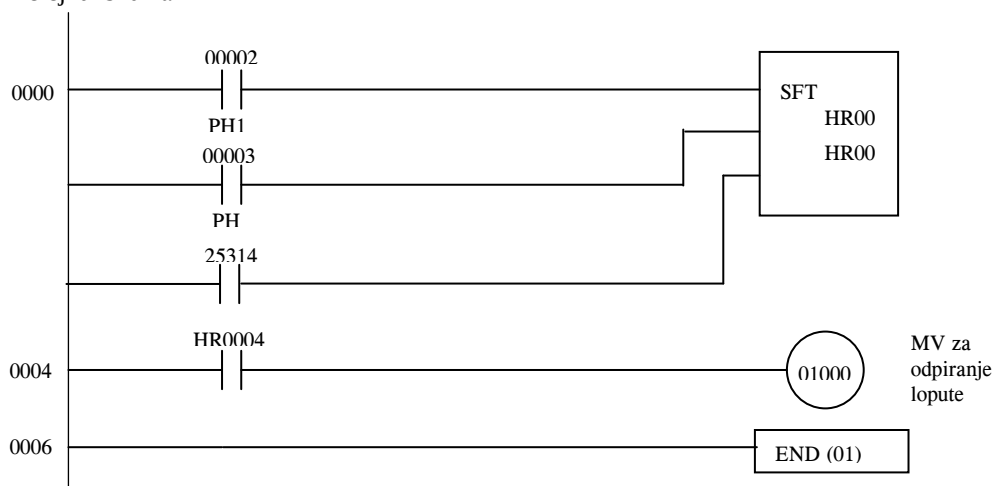
Primer: Sortiranje izdelkov

Slabe izdelke, ki prihajajo na transporter, zaznava senzor PH1(ON). Program mora aktivirati po določenem številu izdelkov ventil za odpiranje lopute v transporterju, tako da zaznan slab izdelek pade v posebno posodo. PH1 bomo uporabili kot podatkovni vhod v SFT, PH2 pa daje takt delovanju SFT v ritmu pomikanja transporterja za en izdelek. Od trenutka zaznave slabega izdelka, ga SFT sledi dokler ne prispe na pozicijo 4, kjer se aktivira ventil za odpiranje lopute.



Adresa	Ukaz	Podatek
0000	LD	00002
0001	LD	00003
0002	LD	25314
0003	SFT	HR00
		HR00
0004	LD	HR0004
0005	OUT	01000
0006	END(01)	

Relejna shema

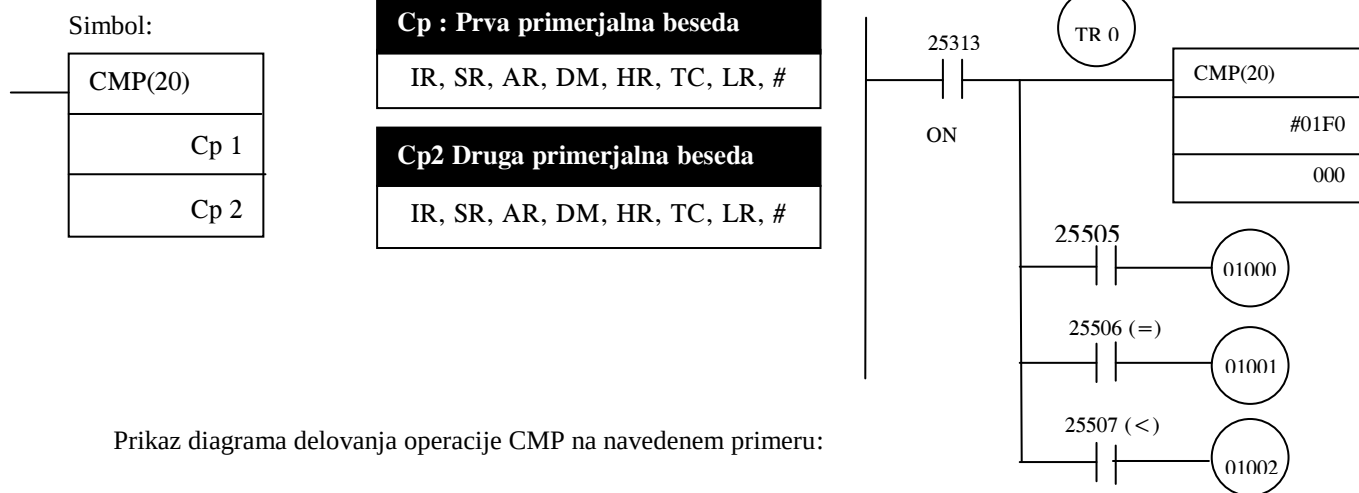


3.3. Primerjanje podatkov (Compare) – CMP

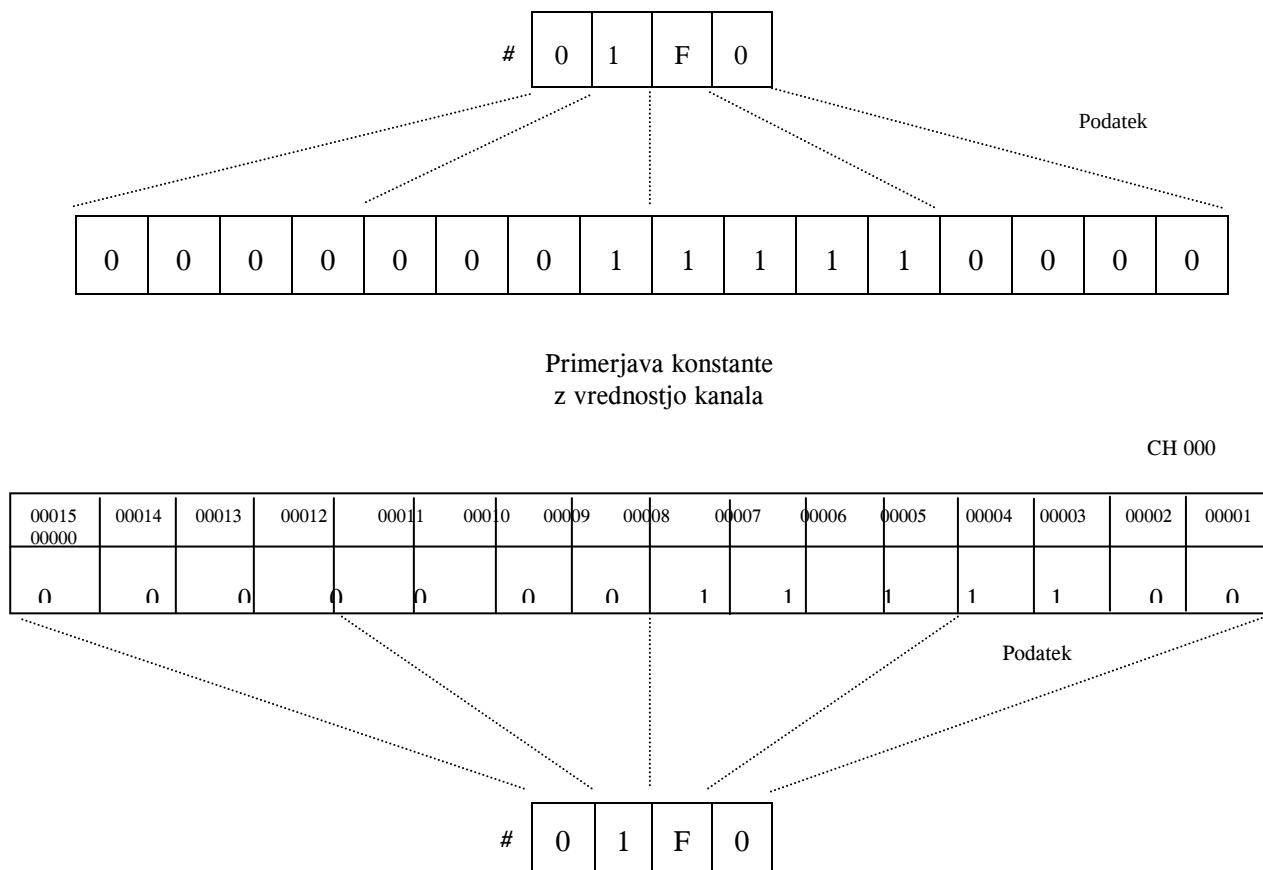
CMP instrukcija primerja podatek v določenem kanalu s podatkom v drugem kanalu ali pa s poljubno 4-bitno heksadecimalno konstanto. Oba podatka morata biti določena takoj po ukazu CMP, rezultat pa je zapisan v obliki zastavice: večje (bit 255.05), enako (bit 255.06) in manjše (bit 255.07).

Tipi podatkov:

Primer relejne sheme:



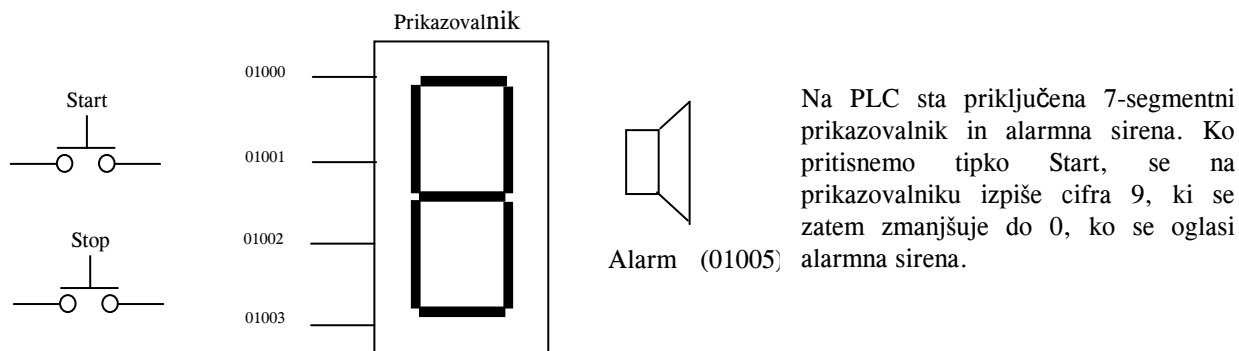
Prikaz diagrama delovanja operacije CMP na navedenem primeru:



Če je konstanta #01F0 enaka podatku na kanalu 000, se specialni bit 255.06 postavi na ON, v primeru manjšega podatka kanala 000 se postavi bit 255.05 in v primeru večjega podatka kanala 000 od konstante pa bit 255.07.

Instrukcije z oznako @ se izvršijo samo ob pozitivni fronti pogojnega signala, kar pomeni, da se instrukcija izvrši samo v enem preletu (scan) programa.

Primer: Alarm izteka časovnega intervala

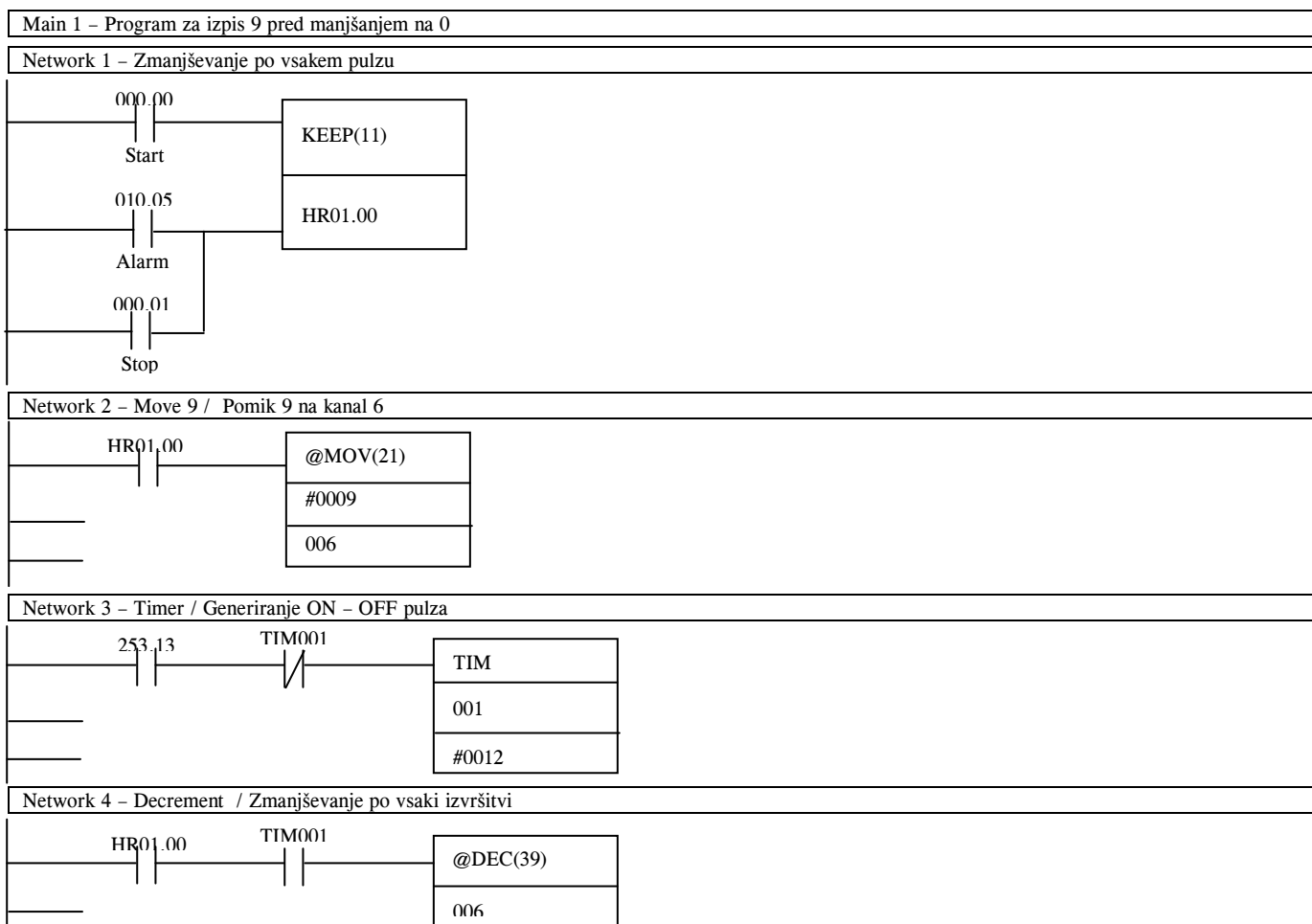


Vhodno izhodna tabela:

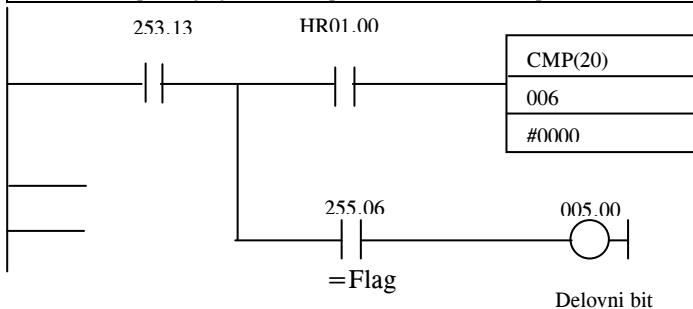
Vhod	Enota
00000	Start – tipka
00001	Stop – tipka

Izhod	Enota
01000 to 01003	7-segmentni prikaz
01005	Alarmna sirena

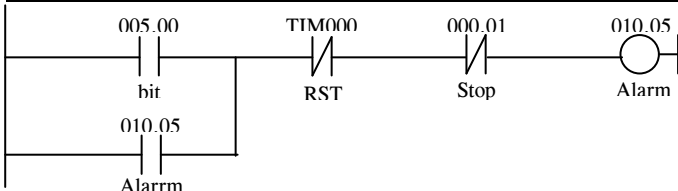
Relejna shema:



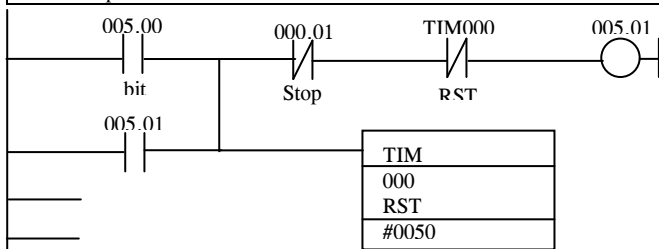
Network 5 – primerjanje vrednosti prikazovalnika z nič, v primeru ENAKO vklopimo 005.00 na ON



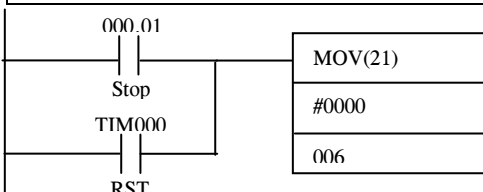
Network 6 – Alarm



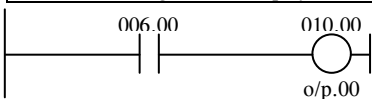
Network 7 –
Timer za prekinitev sirene alarma



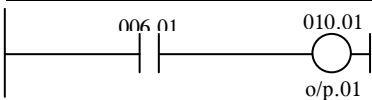
Network 8 – Stop
Reset kanala 10



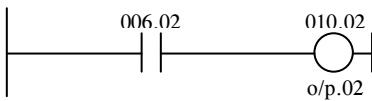
Network 9 – Segment 1 - display



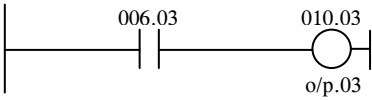
Network 10 – Segment 2 - display



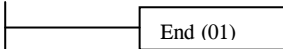
Network 11 – Segment 3 - display



Network 12 – Segment 4 - display

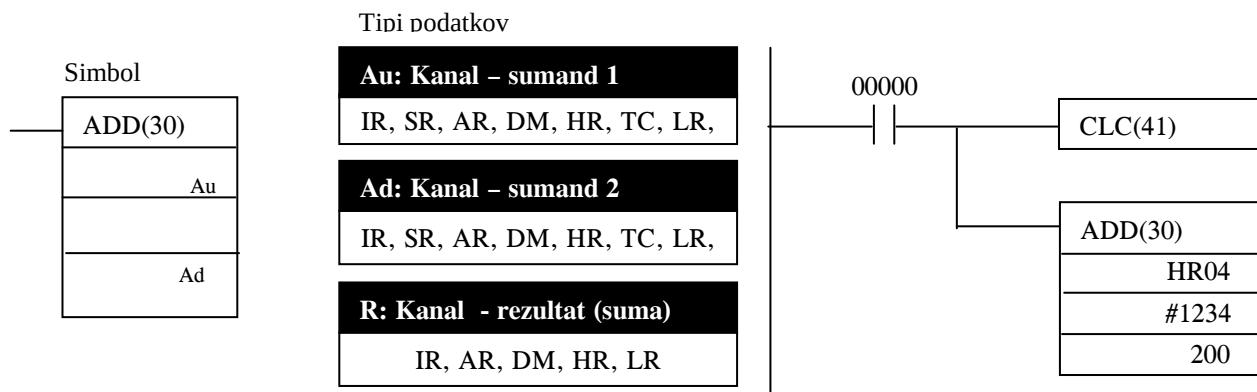


Network 13 – End



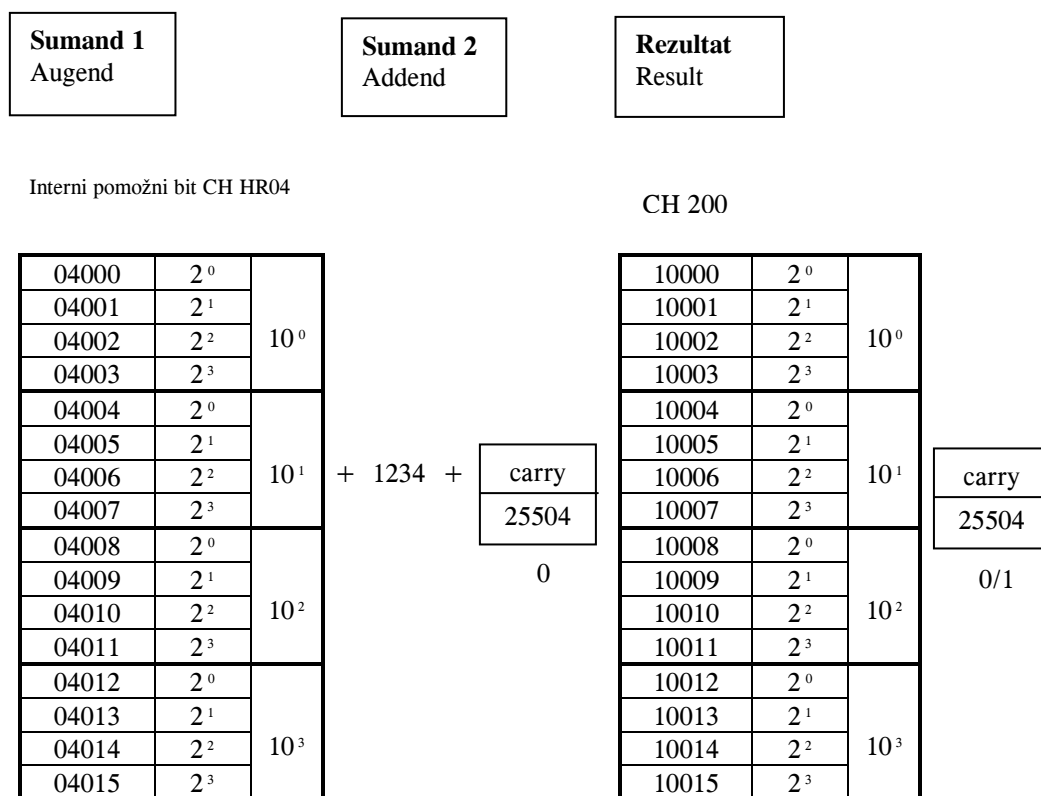
Instrukcija seštevanja – ADD

Instrukcija sešteje vsebini dveh kanalov ali enega kanala in konstanto ter vsoto prenese v tretji kanal (naslov). Določiti je potrebno tri parametre: oba seštevanca (sumanda) in vsoto (sumo).



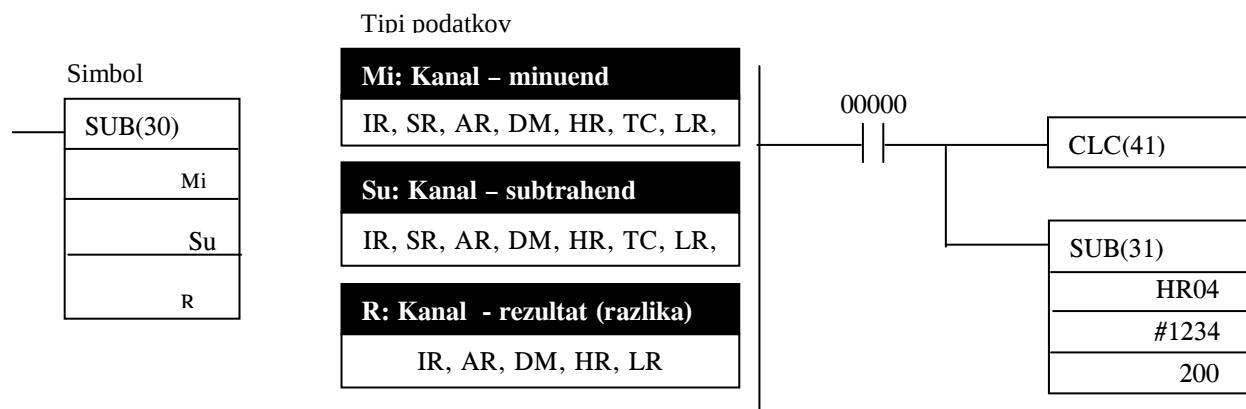
Na začetku operacije izvedemo brisanje prenosa z instrukcijo CLC (Clear Carry), s katero postavimo bit prenosa 255.04 na OFF. Ob pogoju 000.00 (ON) se podatki internega pomnilnega kanala HR040 prištejejo konstanti 1234. Rezultat seštevanja je izhod na naslovu CH 200. V primeru pojava prenosa (carry) kot posledice seštevanja teh dveh števil, se postavi zastavica prenosa (carry flag – SR255.04) na ON. Če je rezultat seštevanja 0000, se posatvi SR 255.06 na ON. Sumanda morata biti v BCD obliki zapisa števila, sicer se postavi specialni bit 255.03 na ON in ADD ukaza ni mogoče izvršiti. ADD se izvede ob vsakem preletu programa, za enkratno izvrševanje uporabimo ukaz @ADD.

Diagram operacije seštevanja:



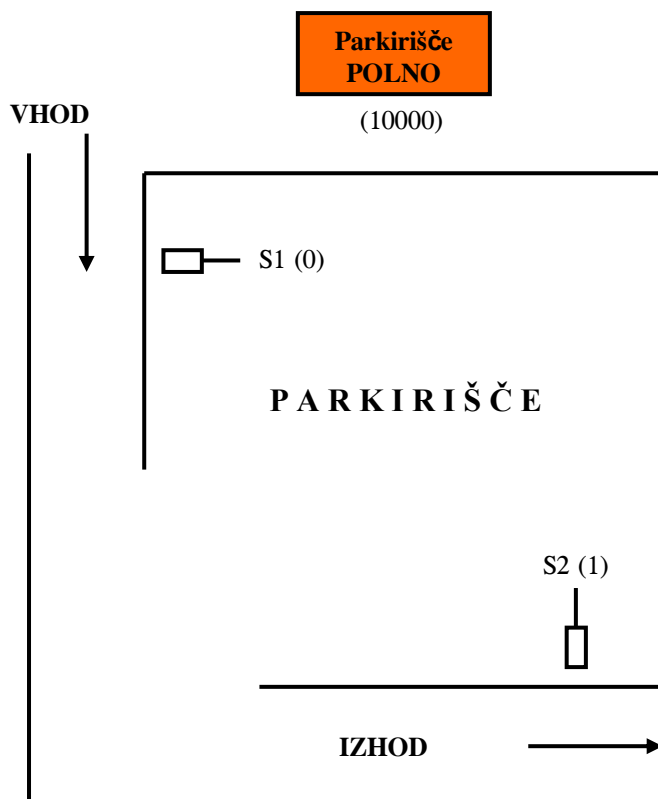
3.4. Instrukcija odštevanja – SUB

Instrukcija poišče razliko številskih podatkov dveh kanalov ali enega kanala in konstanto ter razliko prenese v tretji kanal (naslov). Določiti je potrebno tri parametre: zmanjševanec (minuend), odštevanelec (subtrahend) in razliko (diferenco) oz. rezultat.



Pogoji delovanja instrukcije SUB so enaki pogojem instrukcije ADD.

Primer: Nadzor zasedenosti parkirišča



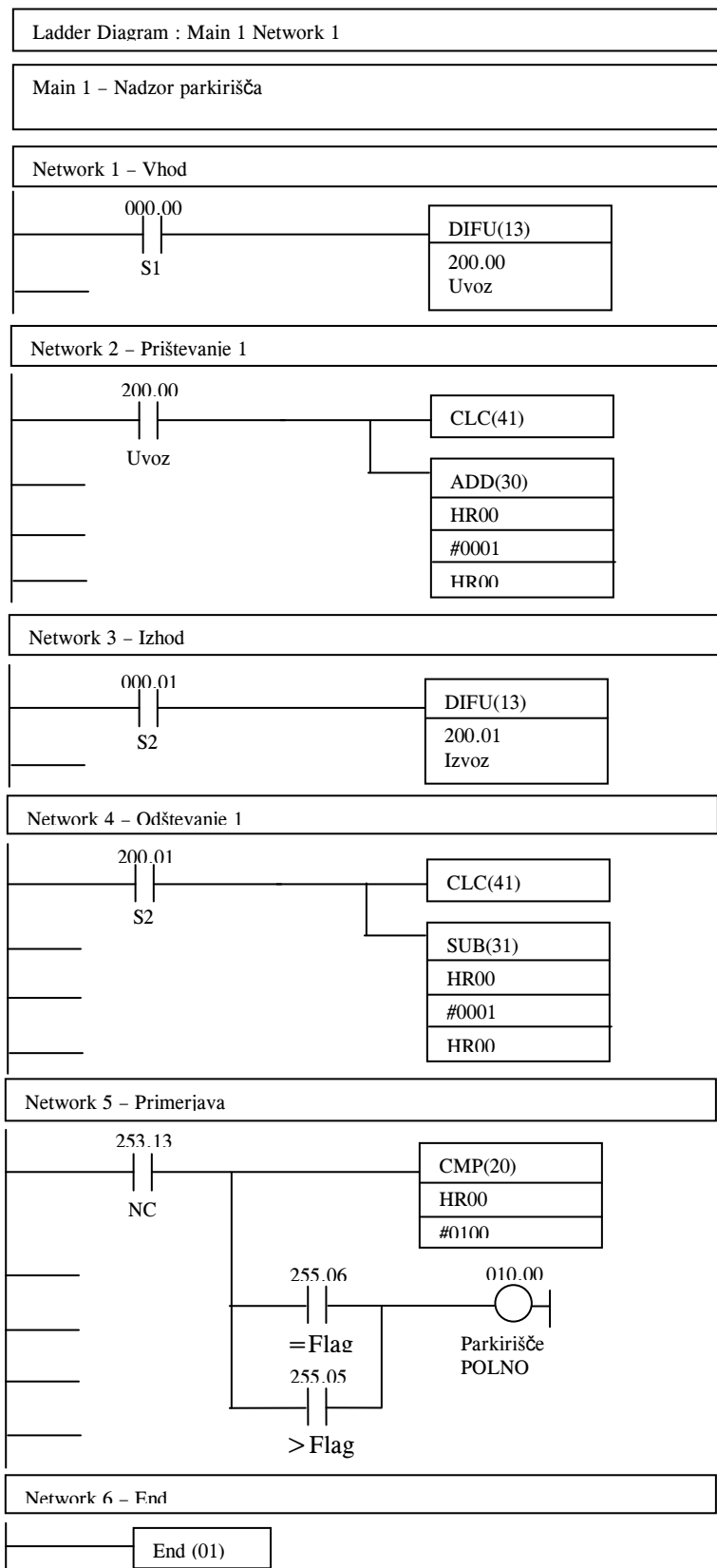
Parkirišče sprejme največ 100 parkiranih avtomobilov. Prihod vsakega avtomobila nadzorujemo s senzorjem S1, odhod vsakega avtomobila pa s senzorjem S2. Ob registraciji 100-tega avtomobila, se prižge oznaka za polno parkirišče, kar pomeni stop za voznike naslednjih avtomobilov. Ob zmanjšanju števila parkiranih avtomobilov se ponovno sprosti vhod,.

Vhodno – izhodni kanali:

Vhod	Enota
00000	Senzor S1
00001	Senzor S2

Izhod	Enota
01000	Signalizacija Parkirišče POLNO

Relejna shema:



Mnemonična koda

Adresa	Ukaz	Podatki
0000	LD	00000
0001	DIFU(13)	20000
0002	LD	20000
0003	CLC(41)	
0004	ADD(30)	
		HR00
		#0001
		HR00
0005	LD	00001
0006	DIFU(13)	20001
0007	LD	20001
0008	CLC(41)	
0009	SUB(31)	
		HR00
		#0001
		HR00
0010	LD	25313
0011	OUT	TR 0
0012	CMP(20)	
		HR00
		#0100
0013	AND	25506
0014	LD	TR 0
0015	AND	25505
0016	OR LD	
0017	OUT	01000
0018	END(01)	

3.6. Instrukciji INC in DEC

Instrukciji povečata (INC – increment) ali zmanjšata (DEC – decrement) določeno vsebino BCD podatkovne besede (Wd) za 1 brez vpliva na prenos (CY – carry). Vsebina podatka se bo povečala oz. zmanjšala ob postavljenem vhodnem pogoju pri vsakem ciklu preleta programa. Za enkratno povečanje ali zmanjšanje vsebine podatka uporabimo možnost instrukcije v obliki @INC (@DEC) ali pa v kombinaciji z ukazoma DIFU ali DIFD.

Simbol	Tipi podatkov:	
	INC, @ INC	Wd: povečanje BCD-INC IR, SR, AR, DM, HR, LR,
	DEC, @ DEC	Wd: zmanjšanje BCD-DEC IR, SR, AR, DM, HR, LR,
Wd		

3.7. Logične instrukcije

Logične instrukcije omogočajo izvedbo logičnih funkcij nad biti v podatkovni besedi.

Komplement – COM

je funkcija negacije, ki ob vsakem ciklu in postavljenem pogoju briše vse ON bite in postavlja vse OFF bite podatkovne besede, ki je lahko tipa: IR, SR, AR, DM, HR, LR. Za enkratno negacijo vsebine podatka uporabimo možnost instrukcije v obliki @COM ali pa v kombinaciji z ukazoma DIFU ali DIFD.

Podatek	15	00
	1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1	
		↓
Komplement	15	00
	0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0	

Logični IN – ANDW

je funkcija, ki nad vsakim bitom dveh vhodnih podatkovnih besed izvede logično operacijo konjunkcije in postavi rezultat v register R. Podobno se izvajajo tudi ostale logične funkcije (ALI-ORW, antivalenca-XORW, ekvivalenca-XNRW).

Simbol	Vhod 1	
	ANDW (34)	IR, SR, AR, DM, HR, TC, LR, #
	vhod1 I1	
	vhod2 I2	
Vhod 1	Vhod 2	
	IR, SR, AR, DM, HR, TC, LR, #	
Rezultat R	Rezultat	
	IR, AR, DM, HR, LR	
Vhod 1	15	00
	1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1	
Vhod 2	15	00
	0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 1	
		↓
Rezultat R	15	00
	0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 1	

4. Obdelava analognih podatkov

Analogne podatke iz procesa lahko zajemamo preko posebnih modulov ali pa direktno z namenskimi PLC oz. njihovimi CPU. Ravno tako lahko generiramo analogni izhodni signal preko posebne aparature opreme. Spodaj je navedenih nekaj tehničnih podatkov A/D in D/A enote krmilnika CQM1 (Omron). Posamezne opcije in karakteristike delovanja A/D in D/A modulov oz. kanalov lahko nastavljamo s pomočjo stikal ali pa programsko s postavitvijo določenih flag-ov.

CQM1-AD041 analogna vhodna enota

CQM1-AD041 enota ima štiri analogne vhode. Z njimi lahko zajamemo analogne signale iz senzorjev oz. merilne opreme. Enoto lahko uporabljamo samo v kombinaciji s posebno napajalno enoto IPS01 ali IPS02.

Število analognih vhodov: 4

Napetostni vhodi: -10V do +10V; 0 do 10V; 1 do 5V

Tokovni vhod: 4 do 20 mA (0 do 20 mA)

Vhodna impedanca - napetostni vhod: 1 M Ω ; tokovni vhod: 250 Ω

Resolucija: 1 / 4000; točnost: $\pm 1\%$; hitrost konverzije: 2,5 ms / točko



CQM1-DA021 analogna izhodna enota

Enota ima tokovni analogni izhod (0 do 20 mA) in napetostni analogni izhod (-10 do +10V). Enoto lahko uporabljamo samo v kombinaciji s posebno napajalno enoto IPS01 ali IPS02.

Število analognih izhodov: 2

Zunanja obremenitev napetostnega izhoda: 1 k Ω min.

Zunanja obremenitev tokovnega izhoda: 520 Ω max.

Resolucija – U: 1 / 4096

Resolucija – I: 1 / 2048

Točnost: $\pm 1\%$; hitrost konverzije: 0.5 ms / 2 točki



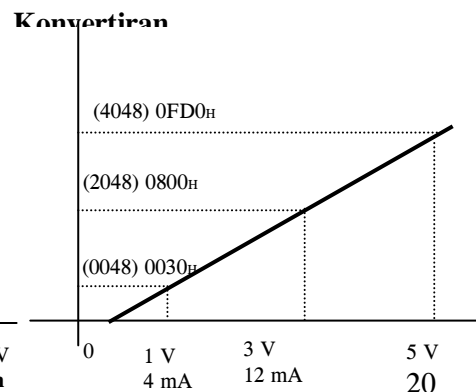
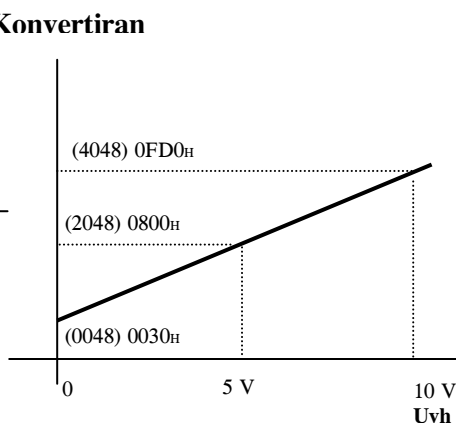
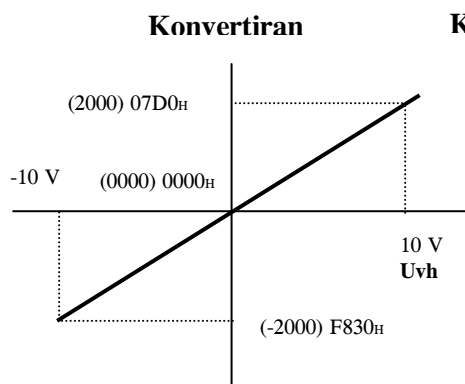
4.1. Obdelava vhodnih analognih podatkov - A/D pretvorba

A/D vhodna enota konvertira analogni signal v digitalni (8, 12 ali 16 bitni) podatek. Metoda konverzije različnih analognih signalov v 12 bitni podatek za naveden A/D modul je naslednja:

- 10V do +10V

0 do 10V

1 do 5V (4 do 20 mA)



Območje pretvorbe hex: F800 do 07FF
Decimalno območje: -2048 do 2047
Konverzija anal.signala: -10 do 10 V
Heksadecimalno: F830 do 07D0
Decimalno: -2000 do 2000

Območje pretvorbe hex: 0030 do 0FFF
Decimalno območje: 48 do 4095
Konverzija anal.signala: 0 do 10 V
Heksadecimalno: 0030 do 0FD0
Decimalno: 48 do 4048

Območje pretvorbe hex: 0000 do 0FFF
Decimalno območje: 0 do 4095
Konverzija anal.signala: 1 do 5V
4 - 20 mA
Heksadecimalno: 0030 do 0FD0
Decimalno: 48 do 4048

V primeru tokovnega vhoda 4 do 20 mA ali napetostnega vhoda 1 do 5 V je spodnja meja dovoljena meja velikosti signala 3,8 mA (0,95V) sicer CPU na modulu aktivira alarm prekinitev povezave (broken wire detection). Ta alarm

lahko uporabimo tudi pri dejanskem preverjanju fizične povezave senzorjev z A/D modulom. Povprečenje vhodnega podatka lahko postavimo z ustreznim stikalom na modulu ali pa programsko (ukaz AVG). Konverzija analognega signala v digitalni podatek se izvrši osemkrat, čas merjenja signala traja 72 ms in šele nato se povprečna vrednost podatka prenese v delovni pomnilnik.

Posamezne analogne kanale naslavljamo s celimi besedami in ne biti. Primer naslavljanja navedenega vhodnega modula v primeru, da imamo pred njim dva digitalna 16 – kanalna vhodna modula, je:

Vhodni kanal 1:	0 do 10 V	Wd 002	-	biti 0 do 11 so uporabljeni za podatek (12 bitna resolucija)
Vhodni kanal 2:	4 do 20 mA	Wd 003	-	bit 12 je uporabljen pri detekciji prekinitve povezave ali
Vhodni kanal 3:	1 do 5 V	Wd 004	-	prenizke vrednosti signala
Vhodni kanal 4:	- 10 do 10 V	Wd 005	-	bit 13 je bit , ki se postavi ob napaki (napačno postavljanje ali programiranje kanal, napačna priključitev....

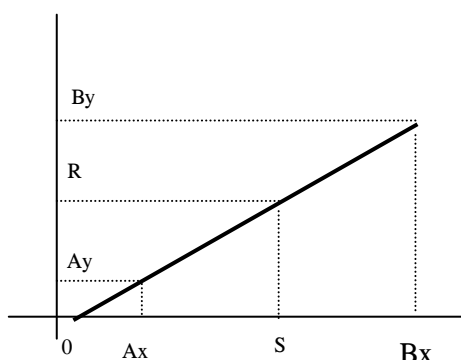
Po priključitvi in zagonu analognega vhodnega kanala se izvede najprej merilni postopek (10 ms) in še povprečenje podatka (72 ms), zato začetne merilne podatke in aktiviranje ostalega aplikativnega dela programa zakasnimo za vsaj 0,1s (ukaz TIM)!

Ukaz SCL

Konverzijo analognih signalov izvedemo z ukazom **SCL** (Scaling – skaliranje). Funkcija linearno konvertira 4 – bitno heksadecimalno vrednost v 4 – mestno decimalno vrednost (BCD). Pretvorba je definirana z dvema točkama A(x,y) in B(x,y), kateri opišemo s parametri P1 do P1+3. Funkcija SCL se zvede ob vhodnem pogoju ON in povzroči pretvorbo 4 bitne HEX vrednosti, ki je zapisana v S, v 4 mestno BCD vrednost, ki jo zapišemo v R. Vrednost BCD je odvisna od definiranih parametrov (P1, P1+1) in (P1+2, P1+3). Rezultat pretvorbe je celoštevilčen, vrednosti manjše od 0000 se zapišejo v R kot 0000 in vrednosti večje od 9999 kot vrednosti 9999.

Simbol	Tipi podatkov	Funkcija in območje parametra P1:
SCL (66)	S – Izvorna beseda	Paramter Funkcija Območje
S	IR, SR, AR, DM, HR, TC, LR,	P1 Ay: BCD točka #1 0000 – 9999
P1	P1 – parameter 1	P1+1 Ax: HEX točka #1 0000 – FFFF
R	IR, SR, AR, DM, HR, TC, LR,	P1+2 By: BCD točka #2 0000 – 9999
	R - rezultat konverzije	P1+3 Bx: HEX točka #2 0000 –FFFF
	IR, AR, DM, HR, LR	

Vrednost BCD po pretvorbi



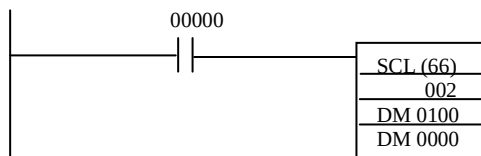
Vrednost HEX pred pretvorbo

$$\frac{By - R}{Bx - S} = \frac{By - Ay}{Bx - Ax}$$

$$R = By - \frac{By - Ay}{Bx - Ax} \cdot (Bx - S); \quad S = Bx - \frac{Bx - Ax}{By - Ay} \cdot (By - R)$$

$$y = k \cdot x + n = \frac{y_1 - y_0}{x_1 - x_0} \cdot x + n$$

Primer uporabe ukaza SCL



- naslov vhodnega analognega signala: Wd = 002
- parametri:

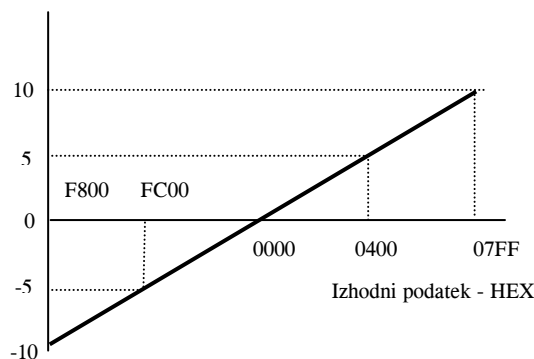
P1 = DM 0100 0000 BCD
P1 + 1 = DM 0101 0030 HEX
P1 + 2 = DM 0102 0100 BCD
P1 + 3 = DM 0103 0FD0 HEX

Analogni vhodni signal iz prvega kanala A/D (naslov Wd = 002) se pretvori v HEX vrednost in ob postavljenem pogoju 000.00 skalira sorazmerno podatkom v DM 0100 in DM 0102 (od 0 (%) do 100 (%) vrednosti). Rezultat se shrani na lokacijo DM 0000. Skaliranje se je izvršilo tako, da heksadecimalne vrednosti binarnih podatkov med 0030 in 0FD0 linearno odgovarjajo decimalnim vrednostim med 0 in 100 (BCD). Ta podatek pa lahko koristno uporabimo naprej v odstotkih.

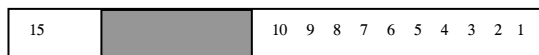
4.2. Obdelava izhodnih analognih podatkov - D/A pretvorba

D/A izhodna enota pretvori podatek iz CPU v napetostni (- 10V do +10V) ali tokovni signal (0 do 20mA), ki je doveden na izhodne priključne sponke modula.

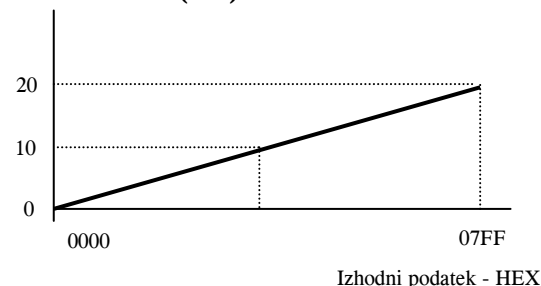
Napetostni izhod (V)



Podatki med 0000 in 07FF_{hex} iz CPU se pretvorijo v izhodno napetost 0 do 10V. Negativno napetost dobimo s pretvorbo preko dvojnega komplementa z MSB. Podatki med F800 in FFFF se pretvorijo v napetostni izhod -10V do 0V. Če negativna napetost ni dopuščena, potem je izhodni signal 0V. Pretvorba je 11 bitna, podatka 0000 in 7800_{hex} oba predstavljata izhod 0V.



Tokovni izhod (mA)



Podatki med 0000 in 07FF_{hex} iz CPU se pretvorijo v izhodni tok 0 do 20mA. Bit 15 mora biti vedno posatvljen na 0. Pretvorba je 11 bitna, podatka 0000 in 7800_{hex} predstavljata izhod 0mA. Razporeditev bitov med LSB in MSB je enaka kot pri napetostnem izhodu.

Analogni izhodni kanali zasedejo naslov celotne besede, zato jih v tem smislu tudi naslavljamo. Paziti je potrebno, na katero mesto vgradimo D/A modul. V primeru vgradnje takoj za vhodnimi kanali, dodelimo obema kanaloma naslova Wd = 100 in 101. Pretvorbo in skaliranje podatkov izvedemo podobno kot pri vhodnih moduli, uporabimo pa lahko skaliranje med BCD vrednostjo podatka in analogni vrednostjo podatka. Primerni ukazi za izvajanje teh aplikacij so za krmilnike družine Omron SCL, SCL3 in APR.

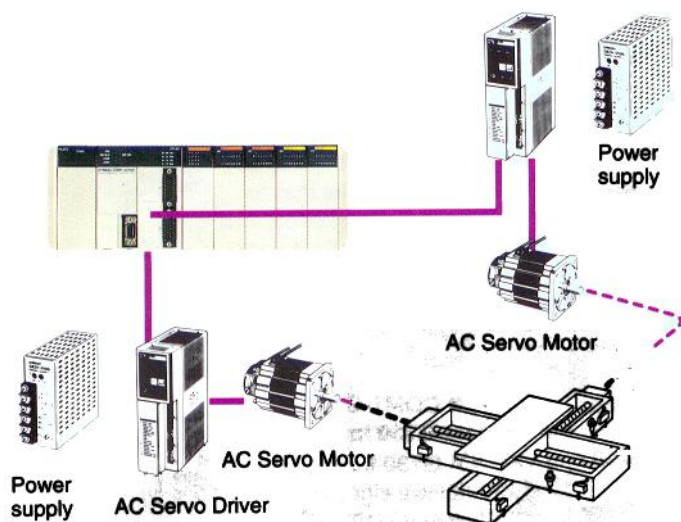
5. Hitro procesiranje podatkov

Pojem pomeni obdelavo vhodno – izhodnih signalov, katerih frekvenca pojavljanja je večja od 1 kHz. Sem prištevamo detektorje hitrega spreminjanja (foto senzorji), inkrementalne dajalnike (nadzor vrtenja, pozicije) kot vhodne elemente in koračne motorje kot izhodne elemente.

Zajemanje in posredovanje hitrih signalov lahko v PLC sistemih izvajamo samo s pomočjo prekinitve delovanja osnovnega programa, obdelavo hitrega signala preko posebnih hitrih modulov in po obdelavi vrnitev v osnovni glavni program. Takšen postopek delovanja imenujemo prekinitveno procesiranje (Interrupt Processing).

Posebne procesorske enote omogočajo preprosto kontrolo hitrosti in pozicije motorja. Enota ima na vhodu dvojne hitrih pulzno vhodno/izhodnih vrat z vhodnim frekvenčnim obsegom 50kHz in izhodnim frekvenčnim obsegom 50kHz. To omogoča preprosto dvo osno pozicioniranje in kontrolo hitrosti s frekvenčno pretvorbo.

Primer dvo-osnega krmiljenja:



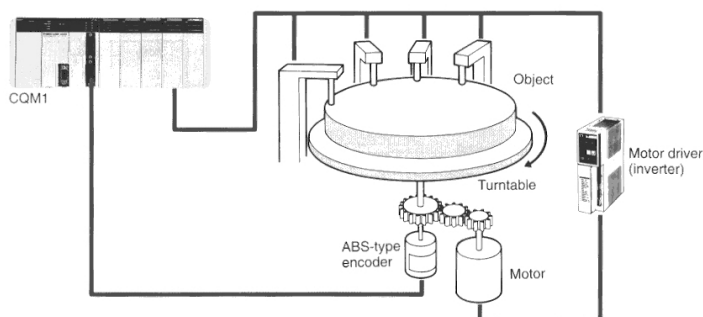
Tehnične značilnosti impulznega vhoda :

- vhodna napetost: 12 VDC, 24 VDC
- vhodni tok: 5mA do 12 mA
- hitrost štetja: 50 kcps (25 kcps)

Tehnične značilnosti impulznega izhoda:

- impulzni izhod: 30 mA NPN, OC, 5-24VDC

Primer zajemanja podatkov iz inkrementalnega dajalnika



- | | |
|-------------------|-------------------|
| Vhodna napetost: | 24 VDC |
| Vhodna impedanca: | 5,4 kΩ |
| Vhodni tok: | 4 mA |
| Hitrost štetja: | 4 kHz |
| Vhodna koda: | binarna Gray koda |

5.1. Tipi hitrega procesiranja

Vhodno prekinitveno procesiranje – Interrupt Input Processing;

se izvrši v primerih, ko vhodni signal pripeljemo na točno določen hitri vhodni kanal (000.00 – 000.03). V trenutku, ko se postavi hitri prekinitveni zunanji signal na logično 1, CPU začasno prekine izvajanje glavnega programa in skoči na ustrezeni podprogram (subroutine), ki se ob danih pogojih izvrši. Po izvršitvi podprograma se nadaljuje glavni program tam, kjer je bil prekinjen.

Časovno prekinitveno procesiranje – Interval Timer Interrupt Processing;

se izvaja pogojno v časovnih presledkih (interval 0,1 ms). Prekinitveni način je primeren za zajemanje signalov pri ponavljajočih procesih. Zajemanje podatkov se mora izvršiti znotraj skeniranega časa izvajanja programa.

Prekinitiv hitrega števca – High-speed Counter Interrupts (HSC);

impulze iz inkrementalnega dajalnika pripeljemo na določene vhodne kanale (000.04, 000.05 in 000.06), ki jih uporabimo kot interne hitre števce. Impulze štejemo direktno v inkrementalni obliki ali obliki gor – dol. Resetiranje števca je možno v SW ali HW načinu.

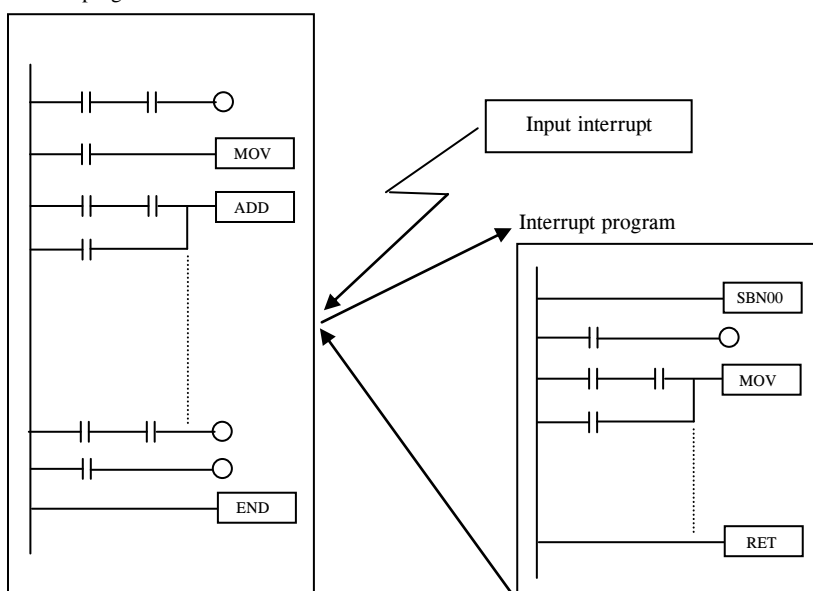
5.2. Vhodne prekinitve (interrupts)

Prekinitveni vhod je lahko postavljen kot vhodni prekinitveni vhod ali kot hitri števeni vhod.

V primeru aktiviranja hitrega prekinitvenega vhoda, se izvajanje glavnega programa zaustavi in se takoj izvede prekinitveni program. Ko je prekinitveni program izveden, se nadaljuje izvajanje glavnega programa od mesta prekinitve.

Hitri impulzi (1 kHz, 5kHz) prihajajo v primeru uporabe hitrega števnega vhoda na hitri števeni vhod. Prekinitve se pojavi takrat, ko število impulzov doseže nastavljeno vrednost (set value SV), glavni program se zaustavi in izvede se prekinitveni program. Po izvedbi prekinitvenega programa se nadaljuje izvajanje glavnega programa. SV števca lahko nastavimo v mejah 0

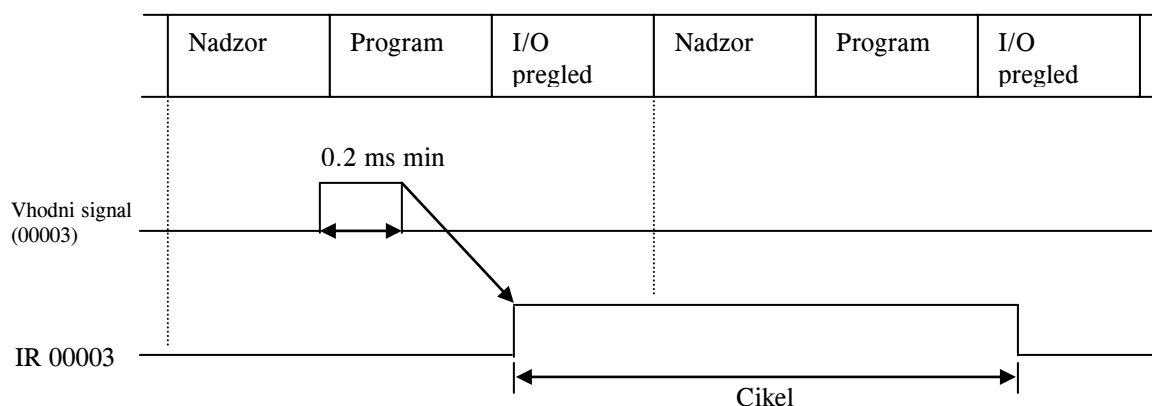
Glavni program



5.3. Funkcija hitrega vhoda

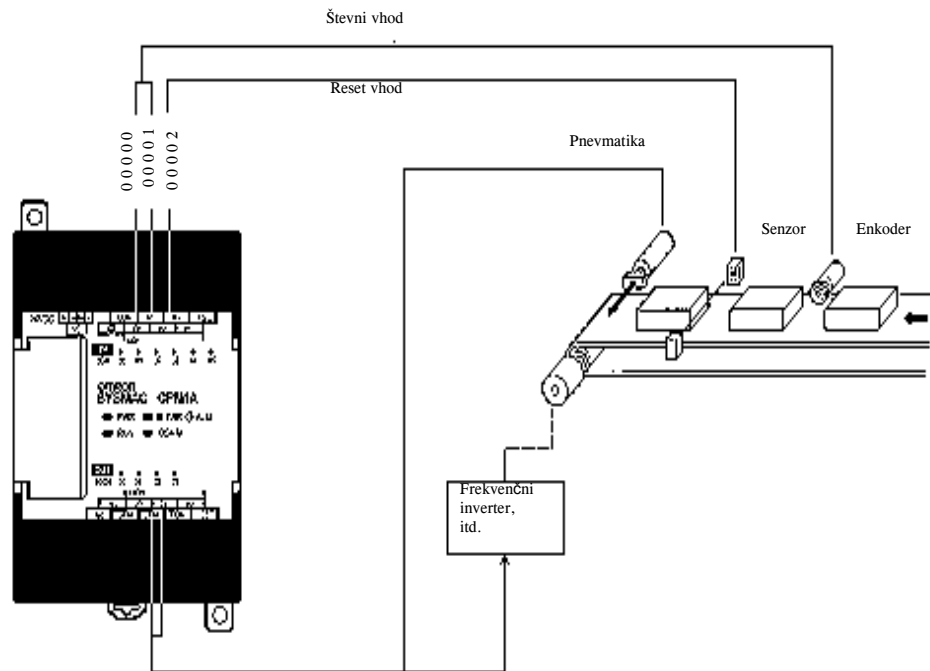
5.3. Funkcija hitrega vhoda

Za hitre vhode uporabljamo iste vhodne kanale kot prekinitveni vhodi. CPM1 ima lahko do 4 takšne vhode. Hitri vhod ima interni buffer, tako da lahko zajemamo tudi signale, ki so krajši od enega delovnega cikla PLC. Minimalna dolžina vhodnega impulza je 0,2 ms. Za CPM1 so kot hitri vhodi postavljeni vhodi IR 000.03, 000.04, 000.05, 000.06.



5.4 Hitri števec

CPM1 ima funkcijo hitrega števca , ki ga lahko uporabimo v inkrementalni obliki (prištevanje impulzov ne glede na smer vrtenja en-koderja) ali obliki gor/dol (up/down mode), ki pri vrtenju en-koderja v eno smer impulze prišteva, pri vrtenju v drugo smer pa odšteva.



Delovanje		Inkrementalni	Up/Down
Vhodni kanal	00000	Števni vhod	A-fazni vhod
	00001	- - -	B-fazni vhod
	00002	Reset vhod	Z-fazni vhod
Vhodna metoda		eno-fazni vhod	fazna diferenca, 4 x vhodi
Števna frekvenca		5.0 kHz	2.5 kHz
Območje števca		0 to 65535	-32767 to 32767
Krmilna metoda	Target value control	Uporabimo lahko do 16 vrednosti TV in s tem prekinitvenih podprogramov	
	Zone comparison control	Uporabimo lahko do 8 spodnjih in zgornjih limitnih vrednosti in 8 prekinitvenih podprogramov	

5.5 Osnovni hitri števeni vhod HSC

Hitri števeni vhod podatkovno obdelamo s funkcijo CTBL (Register Comparison Table), s katero formiramo tabelo, v katero zapišemo vse potrebne podatke za zajemanje in delovanje hitrega števca.

Simbol	Tipi podatkov	CTBL tabela:
CTBL (63)	P – specifikacija vh.kanala	DM 0000 – 0002 število podprogramov
P	000, 001 ali 002	DM 0001 – 0100 TV1, spodnja
C	C – kontrolni podatek	DM 0002 – 0000 TV1, zgornja
TB	000 do 003	DM 0003 – 0101 naslov podprograma 1
	TB – prva beseda tabele	DM 0004 – 0100 TV2, spodnja
	IR, SR,AR, DM, HR, LR	DM 0005 – 0000 TV2, zgornja
		DM 0006 – 0102 naslov podprograma 2

Potek izdelave aplikacije:

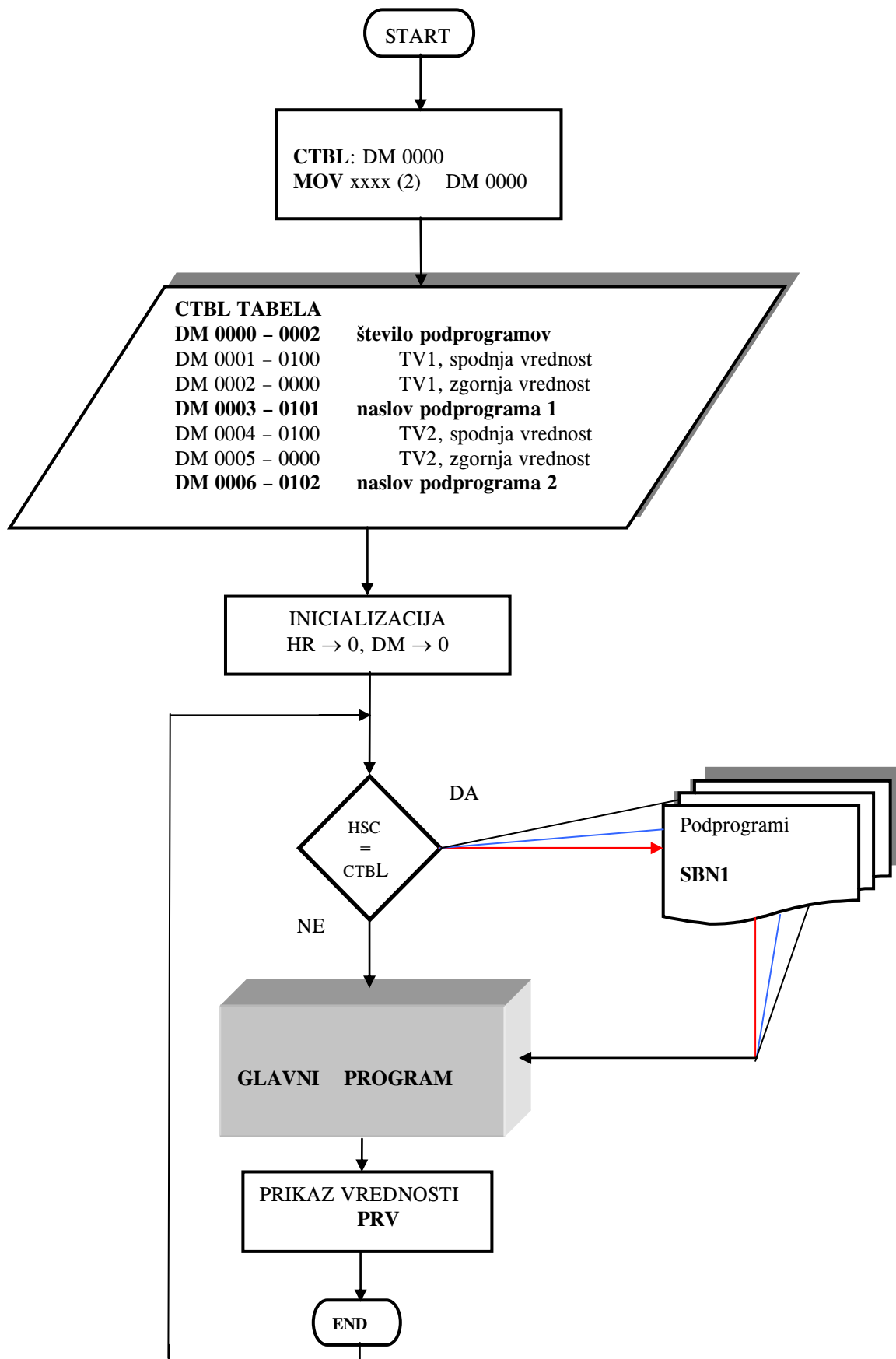
- inkrementalno delovanje: 0 do 65535 impulzov
- PLC nastavitev: DM 6642 = 0114
- določitev vrednosti štetja TV (Target Value): #100
- določitev reseta : SW reset števca – SR 252.00 = 1
- prikaz trenutne vrednosti števca (funkcija PRV)
- izdelava primerjalne tabele CTBL

Koncept programa:

FSF 253.15	CTBL: DM 0000	Definiranje primerjalne tabele
	MOV xxxx (2) DM 0000	Formiranje primerjalne tabele
	HR → 0, DM → 0	Inicializacija krmilnika
ON 253.13	GLAVNI PROGRAM	Pogon naprej, nazaj, A/R, časovniki
	PRV DM 0100	Prikaz trenutnih vrednosti HSC0
	SBN 101 ... RET	Podprogram 1
	SBN 102 ... RET	Podprogram 2
	END	

5.3.

5.6 Diagram poteka hitrega števca



IV. PROGRAMIRLJIVI RELEJNI MODULI - PRM

1. Vloga programirljivih relejnih modulov

Namen programirljivih relejnih modulov PRM je zamenjava klasičnih enostavnejših elektromehanskih relejskih krmilij. Z uporabo PRM pa bistveno skrajšamo in poenostavimo načrtovanje krmiljenja, saj lahko posamezne segmente sproti preizkušamo in testiramo. Rekonstrukcije krmilij ali naknadne spremembe ne predstavljajo posebnih težav, ker so PRM programirljivi preko priročne tastature na samem modulu ali pa preko PC.

Kot izhodna stopnja nastopajo releji ali tranzistorji, ki imajo samo še funkcijo aktuatorja. Vhodni kanali so v večini primerov digitalni, galvansko ločeni od periferije. V nekaterih primerih PRM imamo možnost uporabiti tudi analogne vhode z nižjo resolucijo (7 – 10 bitov). Nekateri tipi modulov imajo tudi hitre šteвне digitalne vhode (5 kHz), ki jih običajno uporabimo za priključevanje inkrementalnih impulznih dajalnikov.

2. Primerjava tehničnih lastnosti različnih PRM

Lastnosti – parametri	Moeller - EASY	Siemens - LOGO	Crozet - MILLENIUM
Napajanje – vhodi	12 VDC 24 VDC 230 VAC	12 VDC 24 VDC, 24 VAC 230 VAC	24 VDC 230 VAC
Poraba	2 – 10 W	0,3 – 9 W	2 – 9 W
Digitalni vhodi			
- DC modeli	8 – 12	8 – 12	6 – 12
- AC modeli	8 – 12	6 – 12	4 – 12
Analogni vhodi			
0 – 10 V samo DC verzije	2	2	6 – 8
ločljivost	1 %	0,1 %	0,5 %
Hitri števní vhod	/	2	/
Relejski izhodi	4 – 6 (8 A)	4 – 8 (10 A)	2 – 8 (8 A)
Tranzistorski izhodi	4 – 8 (0,5 A)	4 – 8 (0,3 A)	/
Možne razširitve			
Vhodov – izhodov	12 vh / 8 tr.izhodov	/	/
S posebnimi moduli	12 vh / 6 rel.izhodov	/	/
Možnost razširitve preko podatkovnih vodil	4 vh / 4 izh	4 vh / 4 izh	4 vh / 4 izh
Preklopna ura	tedensko	Model C	Vsi modeli
Tedenska mesečna			
Nastavitev letni – zimski čas	/	Ročno	Avtomatsko
Točnost ure	± 5 sek / dan	± 5 sek / dan	
Možnost korekcije	/	/	+20 do – 10 sek / dan
Delovanje ure brez zunanjšega napajanja	do 64 ur	do 80 ur	do 20 dni
Programiranje	grafično/relejne sheme	Grafično/funkc. sheme	Grafično/funkc. sheme
Vgrajene funkcije	več kot 20	8 osn.log in 21 posebnih	7 osn.log in 15 posebnih
Število blokov	do 121 povezav	do 56 povezav	do 64 povezav
Zaščita programa	4 števní geslo	Posebni spom. modul	4 števní geslo
Nadzor delovanja programa	Da	Ne	Da
Možnost priklopa modula	Ne	Ne	Da
Možnost blokade tipk	Ne	Ne	Da
Parametri med delovanjem	Da	Da	Ne
Ločljivost časovnih relejev	10 ms	100 ms	100 ms

3. Uporaba, vezava in programiranje PRM

Enostavno "easy"

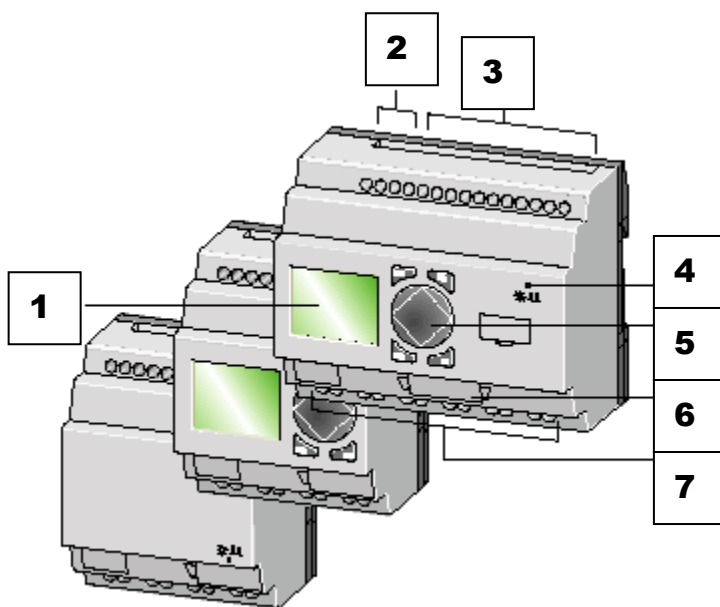
Pametno preklapljanje in krmiljenje

1. "easy" je kompakten, uporabniku prijazen in poceni krmilno relejni modul za enostavne krmilne aplikacije.
2. Uporabimo ga lahko tako v hišnih inštalacijah kakor tudi pri strojih in napravah. "easy" ima na sebi uporabniku prijazne upravljalne elemente in prikazovalnik.
3. "easy" enostavno priključimo in s pomočjo ustreznih tipk narišemo na displej vezalni načrt. Pri tem uporabljamo mirovne in delovne kontakte ter releje.
4. Vezalno shemo vnesemo v "easy" podobno, kot da bi narisali skico sheme na papir. "easy" lahko nadomesti releje, časovnike, zaščitne elemente in še več; spremembe lahko naredimo zelo enostavno s pritiski na ustrezne tipke. Tako odpade čas, ki je sicer potreben za fizične prevezave.

Uporaben je skoraj povsod:

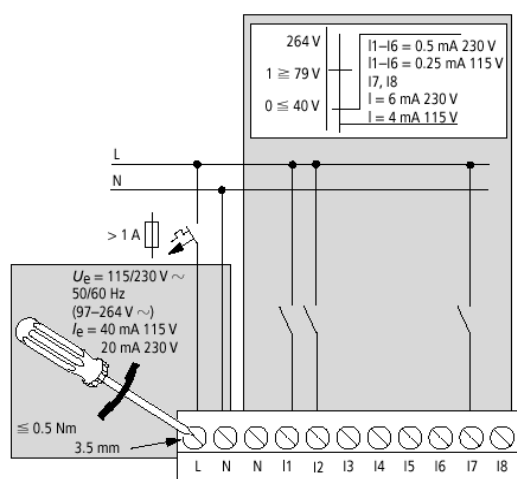
- Za hišne in ostale instalacije, avtomatiko za luči, vrata, okenske rolete,
- Krmiljenje ventilatorjev, vrtljivih vrat, zimskih vrtov, zunanje razsvetljave, odpiranja oken, razsvetljave razstavnih prostorov v trgovinah.
- Krmiljenje strojev in naprav kot so stiskalnice, transportni trakovi, nihalni transporterji, črpalke, sortirni stroji, ...

Modul EASY

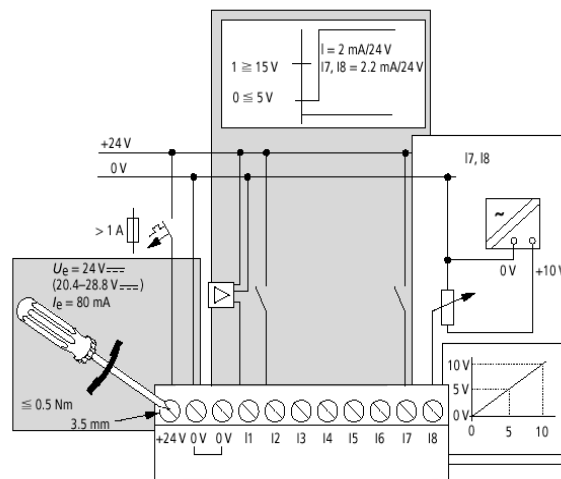


1. LCD zaslon
2. Napajanje
3. Vhodi
4. Signalna lučka
5. Tipke
6. Priključek za spominsko kartico ali povezavo z računalnikom
7. Izhodi

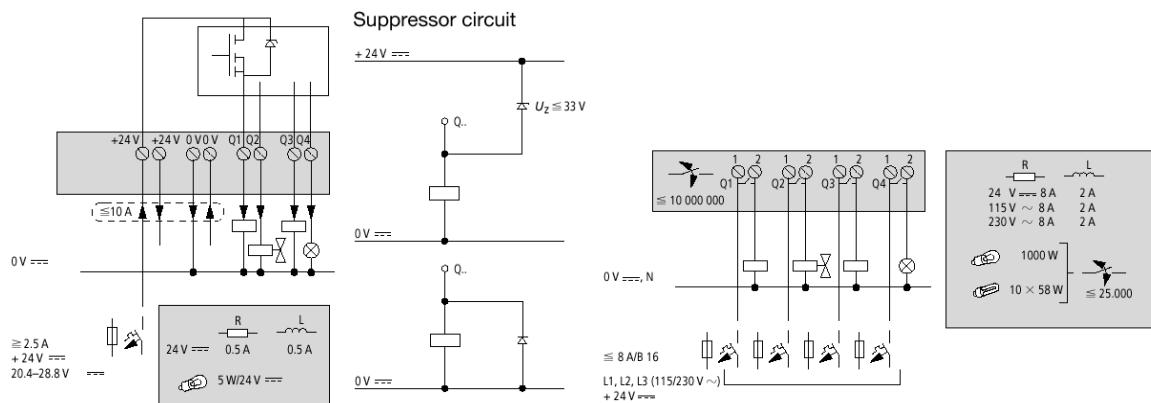
Priključitev vhodov na EASY 412-DC-...



Priključitev vhodov na EASY 412-AC-...

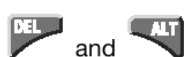


Priključitev izhodnih kanalov



Premikanje po menutih in izbiranje vrednosti

Pokaže se sistemski menu(aktiviranje gesla, ...)



and



Gremo na naslednji menu Izberemo točko menuja Shranimo vrednost

Vračanje na prejšnji menu Prekinemo vnos od zadnjega **OK**

^v Izberemo točko menuja Zamenjamo vrednosti

<> Premikamo kurzor

Funkcije P-tipk (če so omogočene):

< Vhod P1, ^ Vhod P2

> Vhod P3, v Vhod P4

12-.. Prikaz statusa

Vhodi

I12345678

■■■■■■■■ MO

■■■■ 12:50

Izhodi

Q1234 RUN

Dan v tednu

Trenutni čas

Stanje RUN/STOP

■■ On/ ■ Off

EASY 618-..., EASY 620-... Prikaz statusa

Vhodi

1...5...8...

Neizbrisljiv

RE I P

Dan, čas

MO 02:00 ST

Izhodi

.2...5...8 RUN

P gumb

STOP način ob priklopu

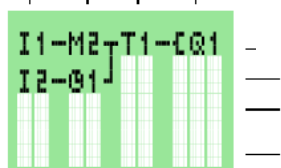
Način

1, 2, 5, 8 On/ . Off

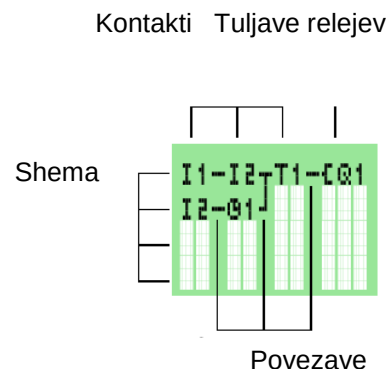
Simboli v vezalni shemi

P	Kontakt : Kurzorske tipke kot vhod
I	Kontakt : Vhod na "easy"-ju
Q	Kontakt : Izhod na "easy"-ju
M	Kontakt : Pomožni rele(marker)
T	Kontakt : Časovnik
C	Kontakt : Števec
Q	Kontakt : Preklopna ura
A	Kontakt : Analogni komparator
O	Kontakt : Tekstovni marker ¹
:	Kontakt : Skok ¹
R	Kontal Tuljave relejev
S	Kontal

1. veja
2. veja
3. veja

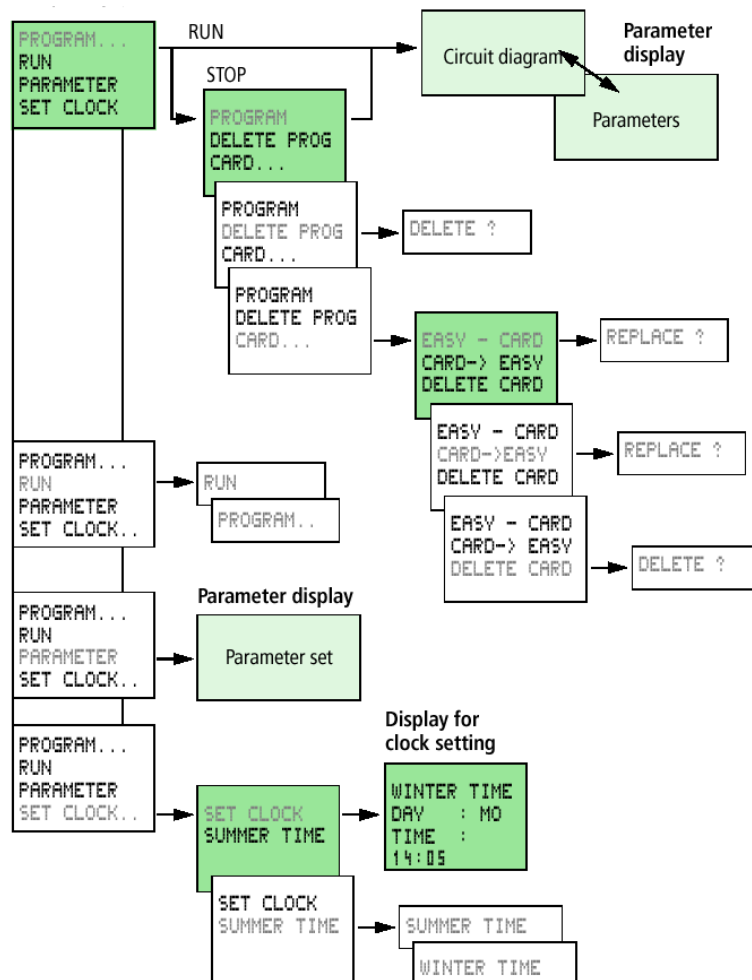


Prikaz vezalne sheme



¹ samo za EASY 618/620

Struktura menuejev



Izdelava vezalne sheme na EASY-ju

“easy” - upravljanje

Tipke, ki so potrebne za izdelavo sheme



Brisanje povezav, kontaktov, relejev ali praznih vej v vezalni shemi.

Preklapljanje med delovnim in mirovnim kontaktom; vstavljanje povezav med kontakti in releji, vstavljanje praznih vej

↕ Zamenjava vrednosti

↕ Premikanje kurzorja navpično

< > Premikanje kurzorja vodoravno

Uporaba tipk P za ročne komande:

< Vhod P1, ↕ Vhod P2

> Vhod P3, ↕ Vhod P4

Preklic zadnje potrditve s tipko OK. Zapustimo trenutni prikaz.

Urejanje in vstavljanje kontaktov/relejev; Shranjevanje nastavitvev

Izbiranje načina delovanja

Obstajata dva načina delovanja: RUN in STOP.

RUN: “easy” obdeluje vezalno shemo (program).

STOP: Rišemo vezalno shemo (program).

V menuju se pod točko RUN/STOP vedno prikazuje naslednja možnost, ki jo imamo na razpolago.

Način delovanja STOP: prikazan je RUN

Način delovanja RUN: prikazan je STOP

```
PROGRAM...
RUN
PARAMETER
SET CLOCK..
```

Simboli vezalne sheme

Kontakti

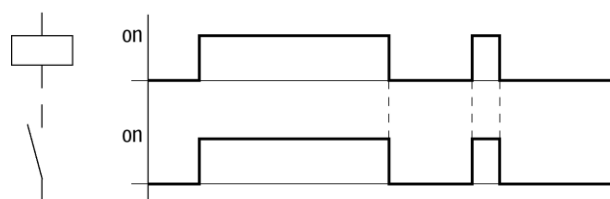
Tip kontakta	Delovni kontakt	Mirovni kontakt	EASY 412	EASY 6...
Kontakti na vhod. sponkah	I	I̅	I1...I8	I1...I12
Kontakti kurzorskih tipk	P	P̅	P1...P4	P1...P4
Kontakti izhodnih sponk	Q	Q̅	Q1...Q4	Q1...Q8
Kontakti pomožnih relejev	M	M̅	M1...M16	M1...M16
Kontakti števecv	C	C̅	C1...C8	C1...C8
Kontakti časovnikov	T	T̅	T1...T8	T1...T8
Kontakti preklapne ure	Ø	Ø̅	Ø1...Ø4	Ø1...Ø4
Kontakti analog. komparatorja	A	A̅	A1...A8	A1...A8
Kontakti tekstovnega releja	D	D̅	—	D1...D8
Kontakti skokov	:	—	—	:1...:8
Kontakti pomožnih relejev	S	S̅	—	S1...S8
Rezervirano	R	R̅	—	R1...R16
Kontakti kratek stik / preobremenitev	I	I̅	I16	I15, I16

Releji

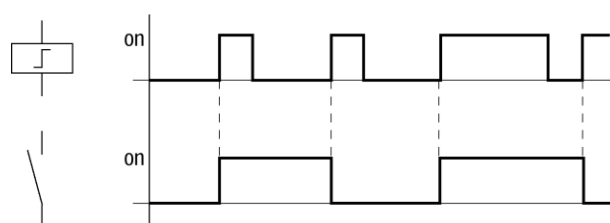
Tip releja	Oznaka	EASY 412	EASY 6...	Funkcija tuljave ?	Parametri ?
Kontakti na vhod. sponkah	I	I1...I8	I1...I12	—	—
Kontakti kurzorskih tipk	P	P1...P4	P1...P4	—	—
Kontakti izhodnih sponk	Q	Q1...Q4	Q1...Q8	X	—
Kontakti pomožnih relejev	M	M1...M16	M1...M16	X	—
Kontakti števecv	C	C1...C8	C1...C8	X	X
Kontakti časovnikov	T	T1...T8	T1...T8	X	X
Kontakti preklapne ure	Ø	Ø1...Ø4	Ø1...Ø4	—	X
Kontakti analog. komparatorja	A	A1...A8	A1...A8	—	X
Kontakti tekstovnega releja	D	—	D1...D8	X	X
Kontakti skokov	:	—	:1...:8	X	—
Kontakti pomožnih relejev	S	—	S1...S8	X	—
Rezervirani	R	—	R1...R16	—	—
Kontakti kratek stik / preobremenitev EASY...DC-T...	I	I16	I15, I16	—	—



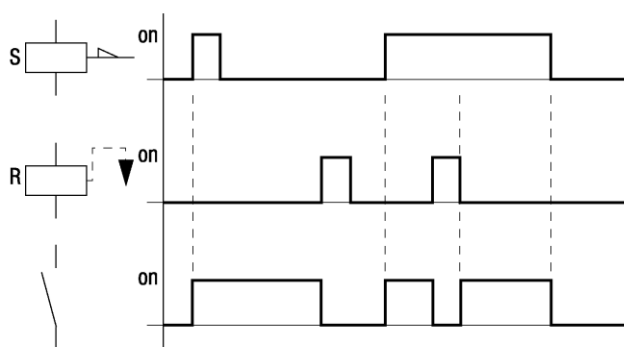
Osnovni rele



Impulzno preklopni rele



Samodržni rele



Funkcija releja

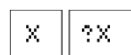
Časovna enota (s, min, h)

Proženje

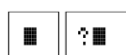
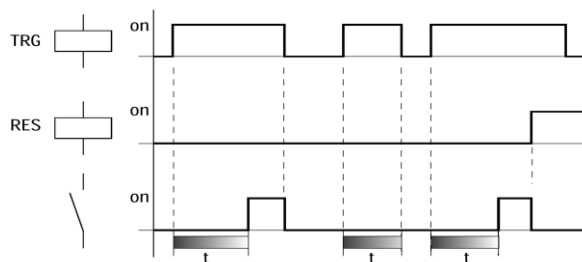
Reset

Λ	00.00	Trenutni čas
S	30.00	Nast. čas
C	TRG	Št. releja
	T1	
	RES	Dostop
	+	možnost
		spreminjanja
		uporabnika
		+ Da
		- Ne

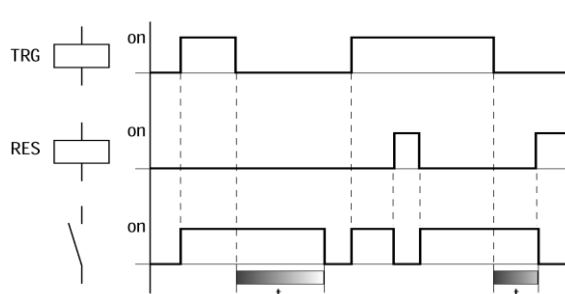
in za



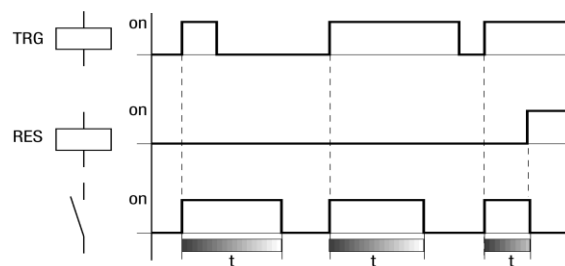
Časovni rele z zakasnjениm vklopom z ali brez definiranega časa



Timing relay, off-delayed, with and without random switching

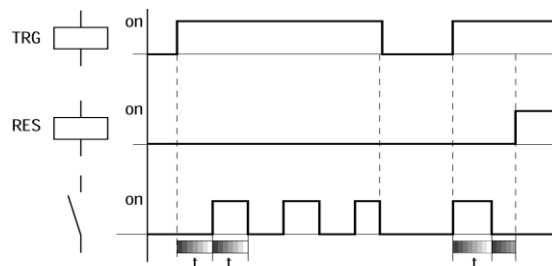


Časovni rele, enojni impulz

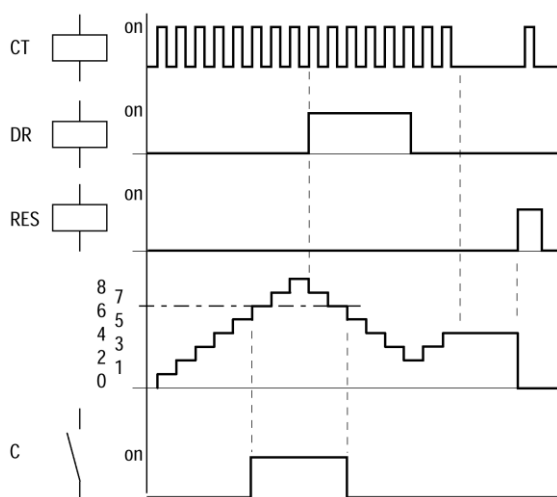


Utripalni rele (zaporedni impulzi)

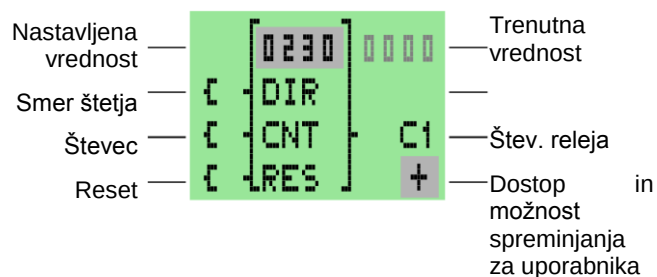
$$\text{Frekv. utripanja} = \frac{1}{2 \times \text{nastav. čas}}$$



Števec

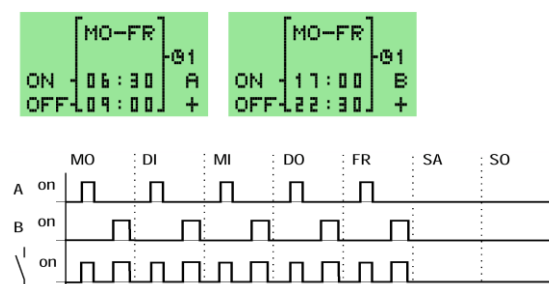


Prikaz parametrov za števec

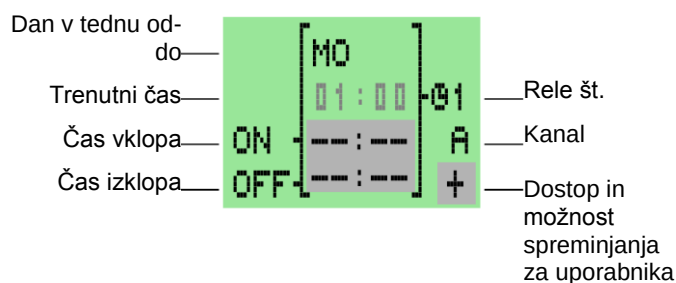


Preklopna ura

Primer: Preklopna ura naj vklopi rele "01" od ponedeljka do petka med 6:30 in 9:00 ter med 17:00 in 22:30.



Prikaz parametrov za preklopno uro



Analogni komparator (analogni komparator lahko primerja napetost od 0 V do 10 V)

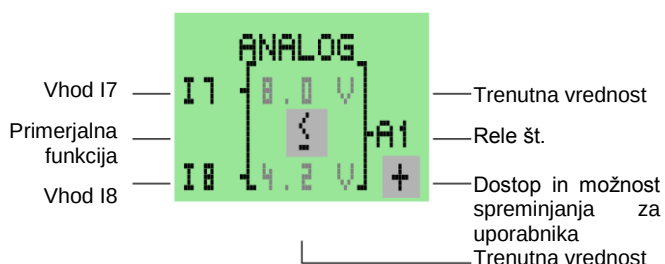
Možne funkcije: $I7 \geq I8$, $I7 \leq I8$

$I7 \geq$ nastavl. vrednost., $I7 \leq$ nastavl. vrednost

$I8 \geq$ nastavl. vrednost, $I8 \leq$ nastavl. vrednost

Prikaz parametrov analognega komparatorja

Primerjava vhodov I7 in I8



Priloga 1: Aplikacijske zanke in skoki

Concept of IL Instruction and JMP instruction

Description

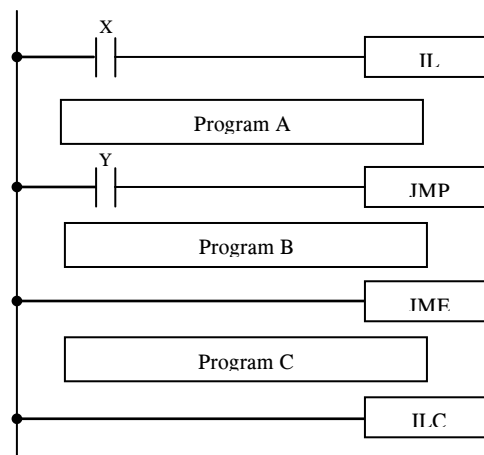
The IL instruction must always be used in conjunction with an ILC instruction, e.g., IL-ILC or IL-IL-ILC. When the IL condition is logical 1, the programs between the IL and IC instructions are executed according to the ladder diagram. When the IL condition is logical 0, all the output relays, internal auxiliary relays and timers in the programs between the IL and ILC instructions are turned OFF, and any counters, shift registers, holding relays and data memory relays in the same programs are held in their present status.

The JMP instruction must always be used in conjunction with a JME instruction, e.g., JMP-JME or JMP-JMP-JME. When the JMP condition is logical 1, the programs between the JMP and JME instructions are executed according to the ladder diagram. When the JMP condition is logical 0, all the output relays, internal auxiliary relays, timers, counters, shift register, holding relays and data memory relays in the same programs are held in their present status.

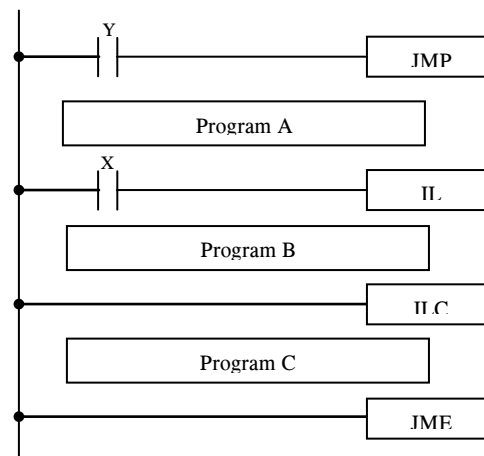
Combination of IL and JMP instructions with combinations such as IL-JMP-ILC-JME and JMP-IL-ILC-JME, the CPU cannot execute programs properly and must therefore be avoided. However, with combinations such as IL-JMP-JME-ILC and JMP-IL-ILC-JME, the CPU performs program execution without problem.

Circuit Example

IL-JMP-JME-ILC



JMP-IL-ILC-JME



Legend

X: IL condition

Y: JMP condition

A: The programs are executed according to the ladder diagram.

B: Output relays, internal auxiliary relays and timers are turned off; counters, shift registers, holding relays and data memory relays are held in their present status.

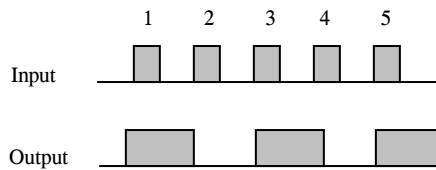
C: Output relays, internal auxiliary relays, timers, relays are held in their present status.

Concition		IL-JMP-JME-ILC			JMP-IL-ILC-JMP		
		Program A	Program B	Program C	Program A	Program B	Program C
X	"1"	A	A	A	A	A	A
Y	"1"						
X	1 ► 1	A	C	A	C	C	C
Y	1 ► 0						
X	1 ► 0	B	B	B	A	B	A
Y	1 ► 1						
X	1 ► 0	B	B	B	C	C	C
Y	0 ► 0						
X	0 ► 0	B	B	B	C	B	C
Y	1 ► 0						
X	0	B	B	B	C	C	C
Y	0						
X	0 ► 1	A	C	A	C	C	C
Y	0 ► 0						
X	0 ► 0	B	B	B	A	B	A
Y	0 ► 1						
X	1 ► 1	A	A	A	A	A	A
Y	0 ► 1						
X	0 ► 1	A	A	A	A	A	A
Y	0 ► 1						

Push-On/Push-Off circuit (Binary Circuit)

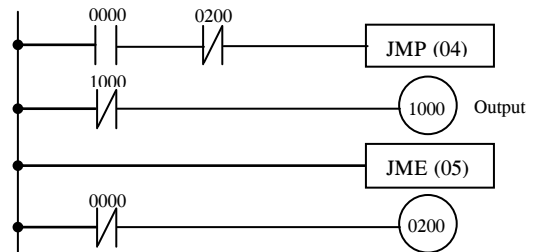
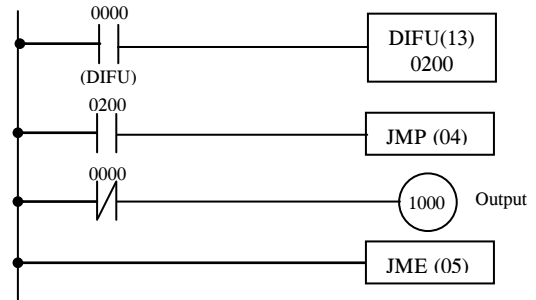
Description

This circuit repeat outputs at specified ON/OFF intervals when a signal (input signal) is applied.



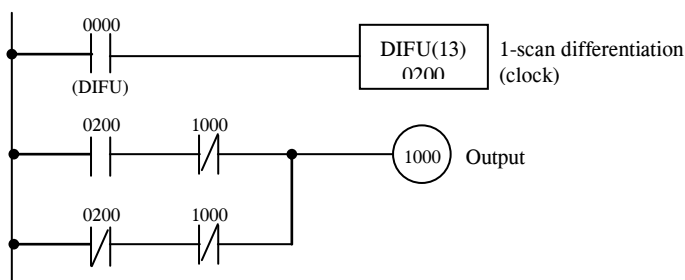
I/O Assignment

Input	0000
Output	1000
Work bit	0200
	0201
	0202

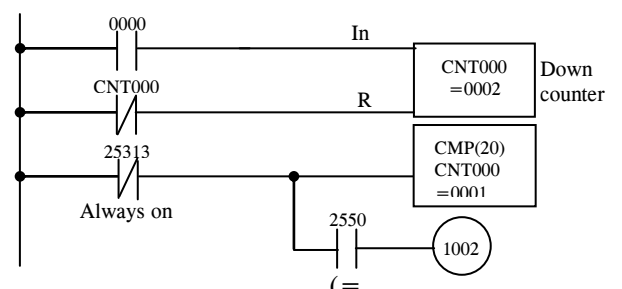
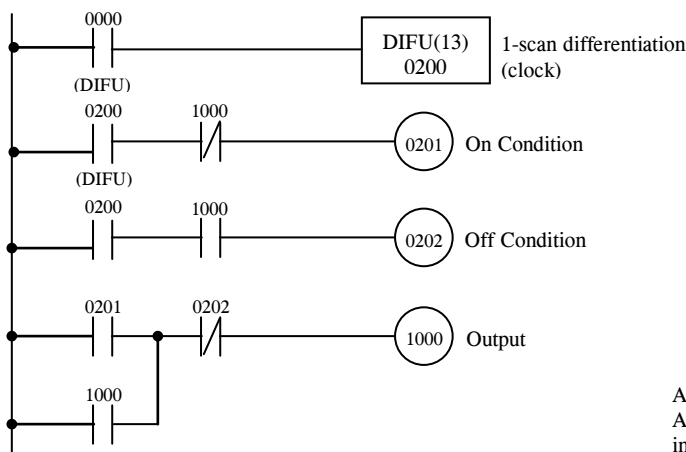
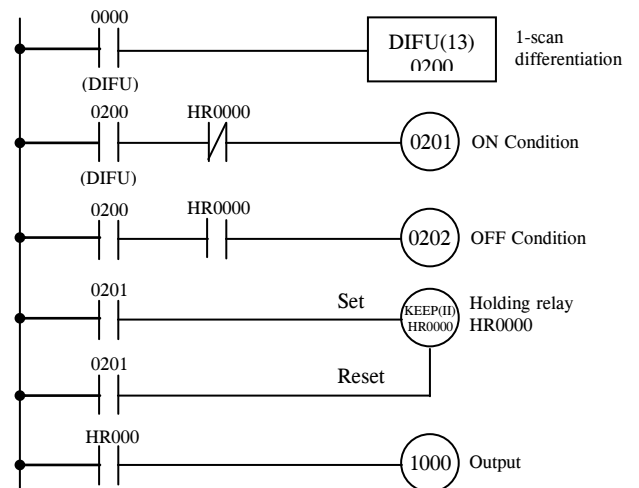


Circuit Example

When data retention during power failure is unnecessary



When data retention during power failure is unnecessary



Auxiliary relay 25313 is normally ON.
Auxiliary relay 25506 turns ON if the result when the Compare (CMP) instruction is executed is equal.

Priloga2: Specifikacija in komponente krmilnika CPM1

Osnovna specifikacija

Parametri		10-point I/O	20-point I/O	30-point I/O	40-point I/O
Supply voltage	AC type	100 to 240 VAC, 50/60 H			
	DC type	24 VDC			
Operating voltage range	AC type	85 to 264 VAC			
	DC type	20.4 to 26.4VDC			
Power consumption	AC type	30 VA max.		60 VA max.	
	DC type	6 W max.		20 W max.	
Inrush current		30 A max.		60 A max.	
External power supply (AC type only)	Power supply Voltage	24 VDC			
	Power supply output capacity	200 mA		300 mA	
Insulation resistance		20 MΩ min. (at 500 VDC) between the external AC terminals and protective earth terminals.			
Dielectric strength		2,300 VAC 50/60 Hz for 1 min between the external AC and protective earth terminals, leakage current: 10 mA max.			
Noise immunity		1,500 Vp–p, pulse width: 0.1 to 1 μs, rise time: 1 ns (via noise simulation)			
Vibration resistance		10 to 57 Hz, 0.075–mm amplitude, 57 to 150 Hz, acceleration: 9.8 m/s ² (1G) in X, Y and Z directions for 80 minutes each (i.e. swept for 8 minutes, 10 times)			
Shock resistance		147 m/s ² (20G) three times each in X, Y and Z directions			
Ambient temperature		Operating: 0 ° to 55 °C Storage: –20 ° to 75 °C			
Ambient Humidity (operating)		10% to 90% (with no condensation)			
Ambient environment (operating)		With no corrosive gas			
Terminal screw size		M3			
Power supply holding time		AC type: 10 ms min; DC type: 2 ms min. (A power interruption occurs if power falls below 85% of the rated voltage for longer than the power interruption time.)			
CPU weight	AC type	400 g max.	500 g max.	600 g max.	700 g max.
	DC type	300 g max.	400 g max.	500 g max.	600 g max.

Karakteristike

Parametri		10-point I/O	20-point I/O	30-point I/O	40-point I/O
Control method		Stored program method			
I/O control method		Combination of the cyclic scan and immediate refresh processing methods.			
Programming language		Ladder diagram			
Instruction length		1 step per instruction, 1 to 5 words per instruction			
Types of instructions		Basic instruction: 14 Special instructions: 77 types, 135 instructions			
Execution time		Basic instructions: 0.72 to 16.2 μs Special instructions: 12.375 μs (MOV instruction)			
Program capacity		2,048 words			
Maximum I/O points	CPU only	10 point (6 input/4 output)	20 point (12 input/8 input)	30 point (18 input/12 output)	40 point (24 input/16 output)
	With Expansion I/O unit	- - -	- - -	99 point (54 input/36 output)	100 point (60 input/40 output)
Input bits		00000 to 00915 (Words 0 to 9)			
Output bits		01000 to 01915 (Words 10 to 19)			
Work bits (IR area)		512 bits: IR 20000 to 23115 (Words IR 200 to IR 231)			
Special bits (SR area)		384 bits: SR 23200 to 25515 (Words SR 232 to IR 255)			
Temporary bits (TR area)		8 bits (TR0 to TR7)			
Holding bits (HR area)		320 bits: HR 0000 to HR 1915 (Words HR 00 to HR 19)			
Auxiliary bits (AR area)		256 bits: AR 0000 to AR 1515 (Words AR 00 to AR 15)			
Link bits (LR area)		256 bits: LR 0000 to LR 1515 (Words LR 00 to LR 15)			
Timers/Counters		128 timers/counters (TIM/CNT 000 to TIM/CNT 127) 100—ms timers: TIM000 to TIM 127 10—ms timers : TIM 00 to TIM 127 Decrementing counters and reversible counters			
Data memory		Read/write: 1,024 words (DM 0000 to DM 1023) Read-only: 512 words (DM 6144 to DM 6655)			
Interrupt processing		2 points (Response time: 0.3 ms max.)	4 points (Response time: 0.3 ms max.)		
Memory protection		HR, AR, Data Memory area contents and counter values maintained during power interruptions.			
Memory backup		Flash memory: User program, data memory (Read only) (Non-battery powered storage) Capacitor backup: Data Memory (Read/Write), holding bits, auxiliary memory bits, counter (20 days at ambient temperature of 25 °C).			
Self-diagnostic functions		CPU failure (watchdog timer), I/O bus error, and memory failure			
Program checks		No END instruction, programming errors (continuously checked during operation)			
High-speed counter		1 point: 5 kHz single-phase or 2.5 kHz two-phase (linear count method) Increment mode: 0 to 65,535 (16 bits) Up/Down mode: -32,767 to 32,767 (16 bits)			
Quick-response inputs		Together with the external interrupts inputs. (Min. pulse width: 0.2 ms)			
Input time constant		Can be set to 1 ms, 2 ms, 4 ms, 8 ms, 16 ms, 32 ms, 64 ms, or 128 ms.			
Analog volume settings		2 controls (0 to 200 BCD)			

Struktura pomnilniških lokacij

Podatkovno okolje		Words	Bits	Function
IR area ¹	Input area	IR 000 to IR 009 (10 words)	IR 00000 to IR 00915 (160 bits)	These bits can be allocated to the external I/O terminals.
	Output area	IR 010 to IR 019 (10 words)	IR 01000 to IR 01915 (160 bits)	
	Work area	IR 200 to IR 231 (32 words)	IR 20000 to IR 23115 (512 bits)	Work bits can be freely used within the program.
SR area		SR 232 to SR 255 (24 words)	SR 23200 to SR 25515 (384 bits)	These bits serve specific functions such as flags and control bits.
TR area		- - -	TR 0 to TR 7 (8 bits)	These bits are used to temporarily store ON/OFF status at program branches.
HR area ²		HR 00 to HR 19 (20 words)	HR 0000 to HR 1915 (320 bits)	These bits store data and retain their ON/OFF status when power is turned off.
AR area ²		AR 00 to HR 15 (16 words)	AR 0000 to HR 1515 (256 bits)	These bits serve specific functions such as flags and control bits.
LR area ¹		LR 00 to LR 15 (16 words)	LR 00000 to LR1515 (256 bits)	Used for a 1:1 data link with another PC.
Timer/Counter area ²		TC 000 to TC 127 (timer/counter numbers) ³		The same numbers are used for both timers and counters.
DM area	Read/write ²	DM 0000 to DM 0999 DM 1022 to DM 1023 (1,002 words)	- - -	DM area data can be accessed in word units only. Word values are required when the power is turned off.
	Error log ⁴	DM 1000 to DM 1021 (22 words)	- - -	Used to store the timer of occurrence and error code of errors that occur. These words can be used as ordinary read/write DM when the error log function isn't being used.
	Read-only ⁴	DM 6144 to DM 6599 (456 words)	- - -	Cannot be overwritten from program.
	PC Setup ⁴	DM 6600 to DM 6655 (56 words)	- - -	Used to store various parameters that control PC operation.

Priloga 3: Programski ukazi, funkcije in instrukcije (CPM1, CQM1)

1. Sekvenčne instrukcije

Sekvenčne vhodne instrukcije

Instruction	Mnemonic	Code	Function
LOAD	LD	0	Connects an NO condition to the left bus bar.
LOAD NOT	LD NOT	0	Connects an NC condition to the left bus bar.
AND	AND	0	Connects an NO condition in series with the previous condition
AND NOT	AND NOT	0	Connects an NC condition in series with the previous condition
OR	OR	0	Connects an NO condition in parallel with the previous condition.
OR NOT	OR NOT	0	Connects an NC condition in parallel with the previous condition.
AND LOAD	AND LD	0	Connects two instruction blocks in series.
OR LOAD	OR LD	0	Connects two instruction blocks in parallel.

Note: 0: Instruction keys allocated to the Programming Console.

Sekvenčne izhodne instrukcije

Instruction	Mnemonic	Code	Function
OUTPUT	OUT	0	Outputs the result of logic to a bit.
OUT NOT	OUT NOT	0	Reverses and outputs the result of logic to a bit.
SET	SET	0	Force sets (ON) a bit.
RESET	RESET	0	Force resets (OFF) a bit.
KEEP	KEEP	11	Maintains the status of the designated bit.
DIFFERENTIATE UP	DIFU	13	Turns ON a bit for one cycle when the execution condition goes from OFF to ON.
DIFFERENTIATE DOWN	DIFD	14	Turns ON a bit for one cycle when the execution condition goes from ON to OFF.

Note: 0: Instruction keys allocated to the Programming Console.

Sekvenčne krmilne funkcije

Instruction	Mnemonic	Code	Function
NO OPERATION	NOP	00	---
END	END	01	Required at the end of the program.
INTERLOCK	IL	02	If the execution condition for IL(02) is OFF, all outputs are turned OFF and all timer PVs reset between IL(02) and the next ILC(03).
INTERLOCK CLEAR	ILC	03	ILC(03) indicates the end of an interlock (beginning at IL(02)).
JUMP	JMP	04	If the execution condition for JMP(04) is ON, all instructions between JMP(04) and JME(05) are treated as NOP(00).
JUMP END	JME	05	JME(05) indicates the end of a jump (beginning at JMP(04)).

Instrukcije za časovnike in števce (Timer/Counter Instructions)

Instruction	Mnemonic	Code	Function
TIMER	TIM	0	An ON-delay (decrementing) timer.
COUNTER	CNT	0	A decrementing counter.
REVERSIBLE COUNTER	CNTR	12	Increases or decreases PV by one.
HIGH-SPEED TIMER	TIMH	15	A high-speed, ON-delay (decrementing) timer.

Note: 0: Instruction keys allocated to the Programming Console.

Podatkovne primerjalne instrukcije

Instruction	Mnemonic	Code	Function
COMPARE	CMP	20	Compares two four-digit hexadecimal values.
DOUBLE COMPARE	CMPL	60	Compares two eight-digit hexadecimal values.
BLOCK COMPARE	(@)BCMP	68	Judges whether the value of a word is within 16 ranges (defined by lower and upper limits).
TABLE COMPARE	(@)TCMP	85	Compares the value of a word to 16 consecutive words.

2. Instrukcije pomikanja podatkov (Data Movement Instructions)

Instruction	Mnemonic	Code	Function
MOVE	(@)MOV	21	Copies a constant or the content of a word to a word.
MOVE NOT	(@)MVN	22	Copies the complement of a constant or the content of a word to a word.
BLOCK TRANSFER	(@)XFER	70	Copies the content of a block of up to 1,000 consecutive words to a block of consecutive words.
BLOCK SET	(@)BSET	71	Copies the content of a word to a block of consecutive words.
DATA EXCHANGE	(@)XCHG	73	Exchanges the content of two words.
SINGLE WORD DISTRIBUTE	(@)DIST	80	Copies the content of a word to a word (whose address is determined by adding an offset to a word address).
DATA COLLECT	(@)COLL	81	Copies the content of a word (whose address is determined by adding an offset to a word address) to a word.
MOVE BIT	(@)MOVB	82	Copies the specified bit from one word to the specified bit of a word.
MOVE DIGIT	(@)MOVD	83	Copies the specified digits (4-bit units) from a word to the specified digits of a word.

3. Instrukcije za pomikanje podatkov (Shift Instructions)

Instruction	Mnemonic	Code	Function
SHIFT REGISTER	SFT	0/10	Copies the specified bit (0 or 1) into the rightmost bit of a shift register and shifts the other bits one bit to the left.
WORD SHIFT	(@)WSFT	16	Creates a multiple-word shift register that shifts data to the left in one-word units.
ASYNCHRONOUS SHIFT REGISTER	(@)ASFT	17	Creates a shift register that exchanges the contents of adjacent words when one is zero and the other is not.
ARITHMETIC SHIFT LEFT	(@)ASL	25	Shifts a 0 into bit 00 of the specified word and shifts the other bits one bit to the left.
ARITHMETIC SHIFT RIGHT	(@)ASR	26	Shifts a 0 into bit 15 of the specified word and shifts the other bits one bit to the right.
ROTATE LEFT	(@)ROL	27	Moves the content of CY into bit 00 of the specified word, shifts the other bits one bit to the left, and moves bit 15 to CY.
ROTATE RIGHT	(@)ROR	28	Moves the content of CY into bit 15 of the specified word, shifts the other bits one bit to the left, and moves bit 00 to CY.
ONE DIGIT SHIFT LEFT	(@)SLD	74	Shifts a 0 into the rightmost digit (4-bit unit) of the shift register and shifts the other digits one digit to the left.
ONE DIGIT SHIFT RIGHT	(@)SRD	75	Shifts a 0 into the rightmost digit (4-bit unit) of the shift register and shifts the other digits one digit to the right.
REVERSIBLE SHIFT REGISTER	(@)SFTR	84	Creates a single or multiple-word shift register that can shift data to the left or right.

4. Instrukcije povečevanja / zmanjševanja (Increment/Decrement Instructions)

Instruction	Mnemonic	Code	Function
INCREMENT	(@)INC	38	Increments the BCD content of the specified word by 1.
DECREMENT	(@)DEC	39	Decrements the BCD content of the specified word by 1.

5. Matematične instrukcije (BCD/Binary Calculation Instructions)

Instruction	Mnemonic	Code	Function
BCD ADD	(@)ADD	30	Adds the content of a word (or a constant).
BCD SUBTRACT	(@)SUB	31	Subtracts the contents of a word (or constant) and CY from the content of a word (or constant).
BCD MULTIPLY	(@)MUL	32	Multiplies the content of two words (or contents).
BCD DIVIDE	(@)DIV	33	Divides the contents of a word (or constant) by the content of a word (or constant).
BINARY ADD	(@)ADB	50	Adds the contents of two words (or constants) and CY.
BINARY SUBTRACT	(@)SBB	51	Subtracts the content of a word (or constant) and CY from the content of the word (or constant).
BINARY MULTIPLY	(@)MLB	52	Multiplies the contents of two words (or constants).
BINARY DIVIDE	(@)DVB	53	Divides the content of a word (or constant) by the content of a word and obtains the result and remainder.
DOUBLE BCD ADD	(@)ADDL	54	Add the 8-digit BCD contents of two pairs of words (or constants) and CY.

Instruction	Mnemonic	Code	Function
DOUBLE BCD SUBTRACT	(@)SUBL	55	Subtracts the 8-digit BCD contents of a pair of words (or constants) and CY from the 80digit BCD contents of a pair of words (or constants).
DOUBLE BCD MULITPLY	(@)MULL	56	Multiplies the 8-digit BCD contents of two pairs of words (or constants).
DOUBLE BCD DIVIDE	(@)DIVL	57	Divides the 8-digit BCD contents of a pair of words (or constants) by the 8 –digits BCD contents of a pair of words (or constants)

6. Instrukcije prekodiranja (Data Conversion Instructions)

Instruction	Mnemonic	Code	Function
BCD TO BINARY	(@)BIN	23	Converts 4-digit BCD data to 4-digit binary data.
BINARY TO BCD	(@)BCD	24	Converts 4-digit binary data to 4 digit BCD data.
4 to 16 DECODER	(@)MLPX	76	Takes the hexadecimal value of the specified digit(s) in a word and turn ON the corresponding bit in a word(s).
16 to 4 DECODER	(@)DPMX	77	Identifies the highest ON bit in the specified word(s) and moves the hexadecimal value(s) corresponding to its location to the specified digit(s) in a word.
ASCII CODE CONVERT	(@)ASC	86	Converts the designated digit(s) of a word into the equivalent 8-bit ASCII code.

7. Logične instrukcije (Logic Instructions)

Instruction	Mnemonic	Code	Function
COMPLEMENT	(@)COM	29	Turns OFF all ON bits and turns ON all OFF bits in the specified word
LOGICAL AND	(@)ANDW	34	Logically ANDs the corresponding bits of two word (or constants)
LOGICAL OR	(@)ORW	35	Logically ORs the corresponding bits of two word (or constants)
EXCLUSIVE OR	(@)XORW	36	Exclusively ORs the corresponding bits of two words (or constants)
EXCLUSIVE NOR	(@)XNRW	37	Exclusively NORs the corresponding bits of two words (or constants).

8. Posebne matematične instrukcije (Special Calculation Instructions)

Instruction	Mnemonic	Code	Function
BIT COUNTER	(@)BCNT	67	Counts the total number of bits that are ON in the specified block

9. Podprogramske instrukcije (Subroutine Instructions)

Instruction	Mnemonic	Code	Function
SUBROUTINE ENTER	(@)SBS	91	Executes a subroutine in the main program.
SUBROUTINE ENTRY	SBN	92	Marks the beginning of a subroutine program.
SUBROUTINE RETURN	RET	93	Marks the end of a subroutine program.
MACRO	MACRO	99	Calls and executes the specified subroutine, substituting the specified input and output words for the input and output words in the subroutine.

10. Prekinitvene instrukcije (Interrupt Control Instructions)

Instruction	Mnemonic	Code	Function
INTERVAL TIMER	(@)STIM	69	Controls interval timers used to perform scheduled interrupts.
INTERRUPT CONTROL	(@)INT	89	Performs interrupts control, such as masking and unmasking the interrupt bits for I/O interrupts.

11. Koračne instrukcije (Step Instructions)

Instruction	Mnemonic	Code	Function
STEP DEFINE	STEP	08	Defines the start of a new step and resets the previous step when used with a control bit. Defines the end of step execution when used without a control bit.
STEP START	SNXT	09	Starts the execution of the step when used with a control bit.

12. Periferne krmilne instrukcije (Peripheral Device Control Instructions)

Instruction	Mnemonic	Code	Function
BCD TO BINARY	(@)BIN	23	Converts 4-digit BCD data to 4-digit binary data.
BINARY TO BCD	(@)BCD	24	Converts 4-digit binary data to 4-digit BCD data.
4 to 16 DECODER	(@)MLPX	76	Takes the hexadecimal value of the specified digit(s) in a word and turn ON the corresponding bit in a word(s).
16 to 4 DECODER	(@)DPMX	77	Identifies the highest ON bit in the specified word(s) and moves the hexadecimal value(s) corresponding to its location to the specified digit(s) in a word.
ASCII CODE CONVERT	(@)ASC	86	Converts the designated digit(s) of a word into the equivalent 8-bit ASCII code.

13. Instrukcije vhodno – izhodnih enot (I/O Units Instructions)

Instruction	Mnemonic	Code	Function
7-SEGMENT DECODER	(@)SDEC	78	Converts the designated digit(s) of a word into an 8-bit, 7-segment display code.
I/O REFRESH	(@)IORF	97	Refreshes the specified I/O word.

14. Display Instructions

Instruction	Mnemonic	Code	Function
MEASSAGE	(@)MSG	46	Reads up to 8 words of ASCII code (16 characters) from memory and displays the message on the Programming Console or other Peripheral

15. Instrukcije hitrega števca (High Speed Counter Control Instructions)

Instruction	Mnemonic	Code	Function
MODE CONTROL	(@)INI	61	Starts and stops counter operation, compares and changes counter PVs, and stops pulse output.
PV READ	(@)PRV	62	Reads counter PVs and status data.
COMPARE TABLE LOAD	(@)CTBL	63	Compares counter PVs and generates a direct table or starts operation.

16. Diagnostične instrukcije (Damage Diagnosis Instructions)

Instruction	Mnemonic	Code	Function
FAILURE ALARM	(@)FAL	06	Generates a non-fatal error when executed. The Error/Alarm indicator flashes and the CPU continues operating.
SEVERE FAILURE ALARM	FAL	07	Generates a fatal error when executed. The Error/Alarm indicator lights and the CPU stops operating.

17. Sistemske instrukcije (Special System Instructions)

Instruction	Mnemonic	Code	Function
SET CARRY	(@)STC	40	Sets Carry Flag 25504 to 1.
CLEAR CARRY	(@)CLC	41	Sets Carry Flag 25504 to 0.

Priloga 4:
 Algoritem delovanja
 naprave in aplikacija

