



DK – STEP 7

Kombinacijske funkcije

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"

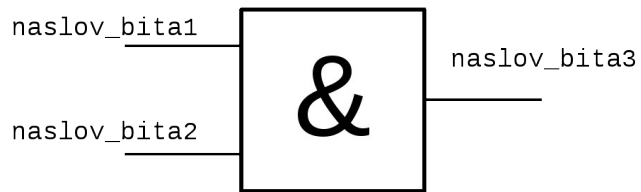
Statusna beseda (STW)

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

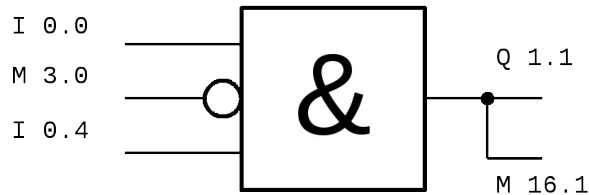


DK – STEP 7

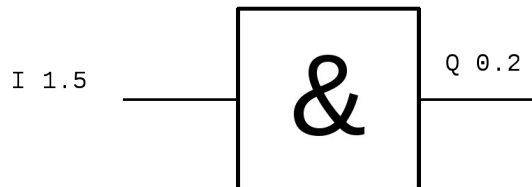
Kombinacijske funkcije - AND



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

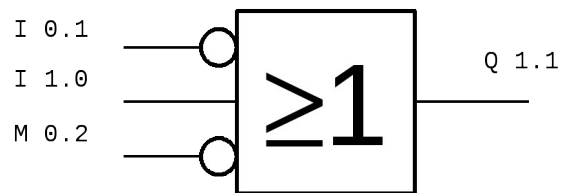


DK – STEP 7

◆ Kombinacijske funkcije - OR

Privoščimo si lahko celo takšen zapis:

```
A  naslov_bita1  
O  naslov_bita2  
=  naslov_bita3
```

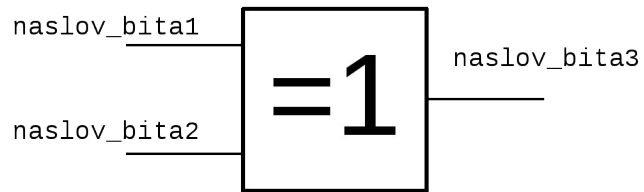


```
ON I 0.1  
O I 1.0  
ON M 0.2  
= Q 1.1
```

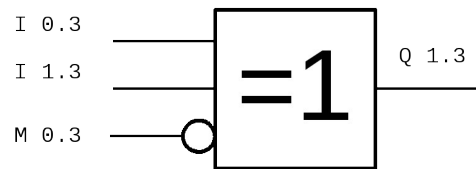


DK – STEP 7

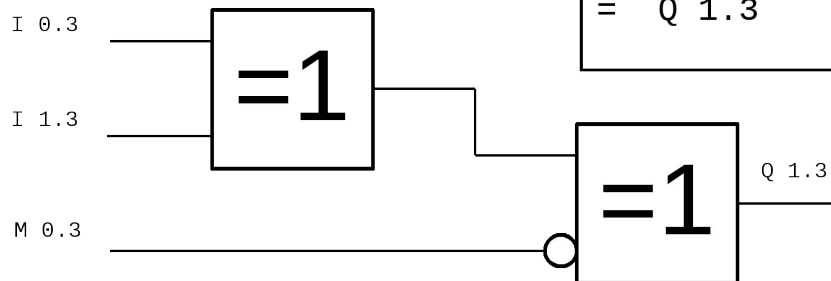
Kombinacijske funkcije - XOR



X naslov_bita1
 X naslov_bita2
 = naslov_bita3



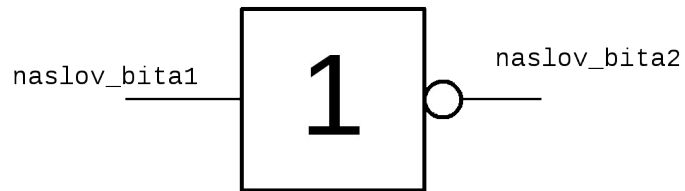
	STA	RLO	/FC
X	X	0	0
1	1	1	1
1	0	1	1
1	1	1	1
1	1	0	0



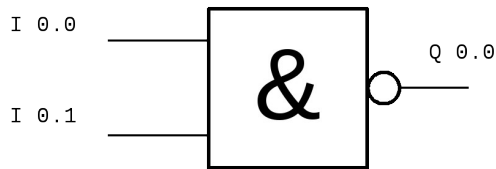


DK – STEP 7

◆ Kombinacijske funkcije - NOT



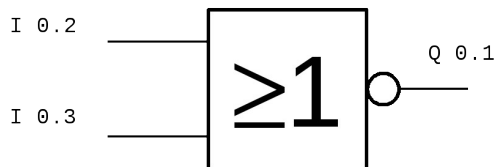
AN naslov_bita1
= naslov_bita2



AN(
A I 0.0
A I 0.1
)
= Q 0.0

A I 0.0
A I 0.1
= M 10.0
AN M 10.0
= Q 0.0

A I 0.0
A I 0.1
NOT
= Q 0.0

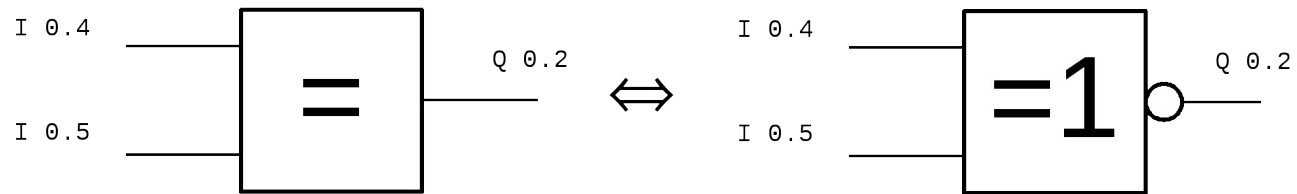


O I 0.2
O I 0.3
NOT
= Q 0.1



DK – STEP 7

◆ Kombinacijske funkcije - ekvivalenca



X I 0.4
X I 0.5
NOT
= Q 0.2

A I 0.4
A I 0.5
0
AN I 0.4
AN I 0.5
= Q 0.2



DK – STEP 7

◆ Kombinacijske funkcije - SET, CLR, SAVE

SET

= Q 0.0

= Q 0.1

= M 4.3

CLR

= Q 0.2

= Q 0.3

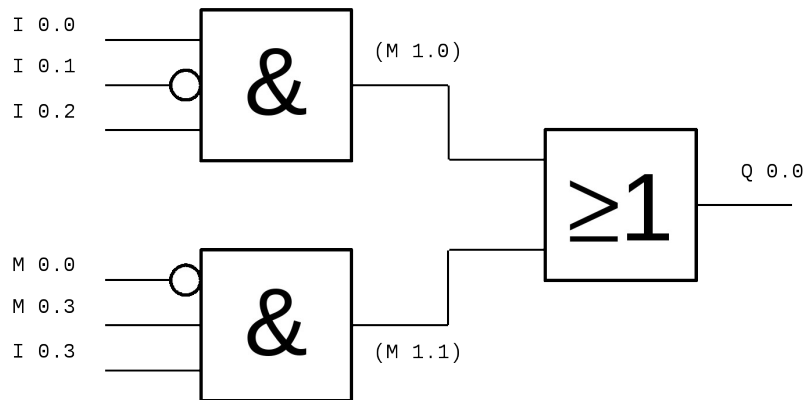
= M 4.4

SAVE



DK – STEP 7

◆ Kombinaijske funkcije - sestavljene



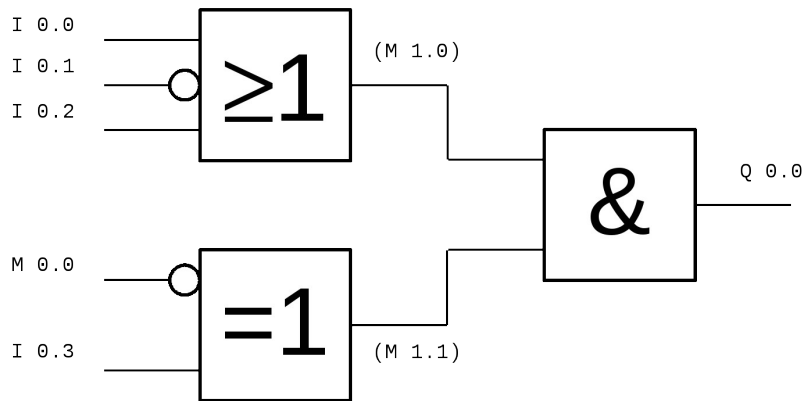
```
A I 0.0
AN I 0.1
A I 0.2
= M 1.0
AN M 0.0
A M 0.3
A I 0.3
= M 1.1
0 M 1.0
0 M 1.1
= Q 0.0
```

```
A I 0.0
AN I 0.1
A I 0.2
0
AN M 0.0
A M 0.3
A I 0.3
= Q 0.0
```




DK – STEP 7

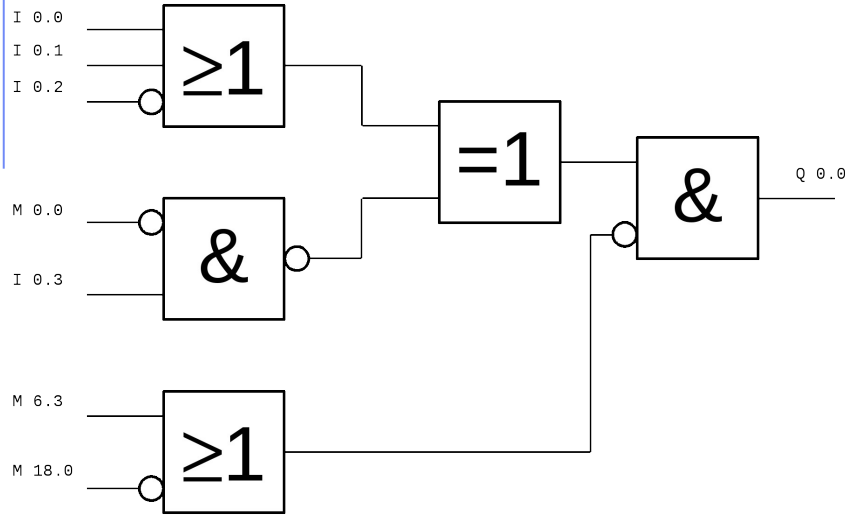
◆ Kombinacijske funkcije - sestavljene





DK – STEP 7

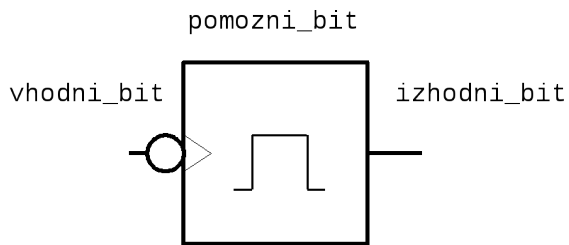
Kombinacijske funkcije - sestavljene



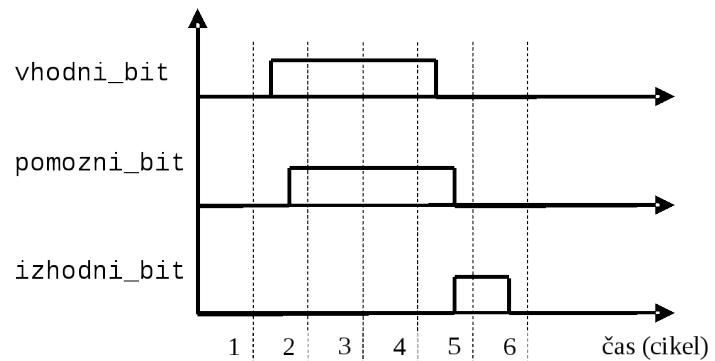


DK – STEP 7

◆ Dinamični člen (DLF)



A vhodni_bit
FN pomozni_bit
= izhodni bit





DK – STEP 7

◆ Dinamični člen (DLF) - zгледа

```
A I 0.0
FP M 1.3
JC skok
:
skok: AN M 2.1
O Q 0.0
= Q 1.1
```

```
A I 0.0
FP M 1.3
= M 19.0
:
A M 19.0 //povpr. po spremembi I 0.0 z "0" na "1"
AN I 1.4
= Q 3.0
:
A M 19.0 //povpr. po spremembi I 0.0 z "0" na "1"
A Q 2.3
S M 4.0
```



DK – STEP 7

◆ Pomnilne funkcije - RS

Ukaz S torej pogojno postavi operand na "1":

```
S  naslov_bita
```

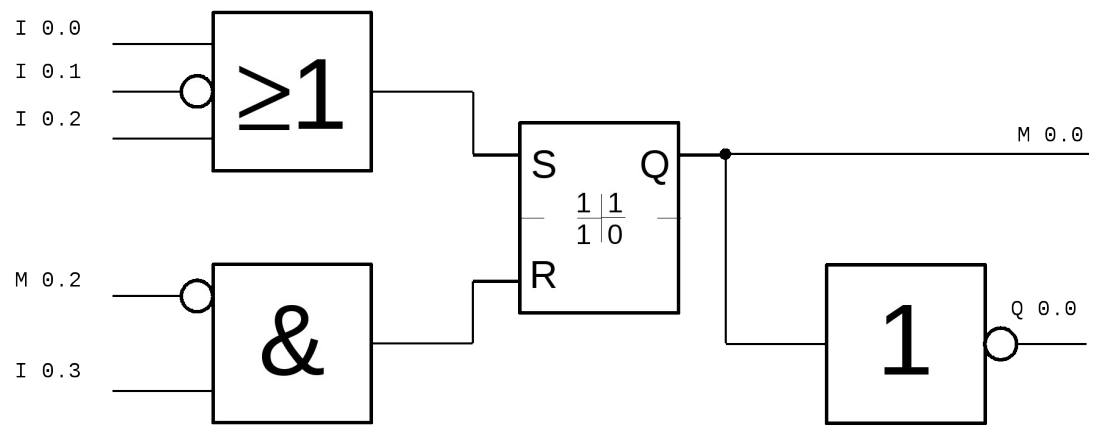
Ukaz R pa pogojno postavi operand na "0":

```
R  naslov_bita
```



DK – STEP 7

Pomnilne funkcije - RS

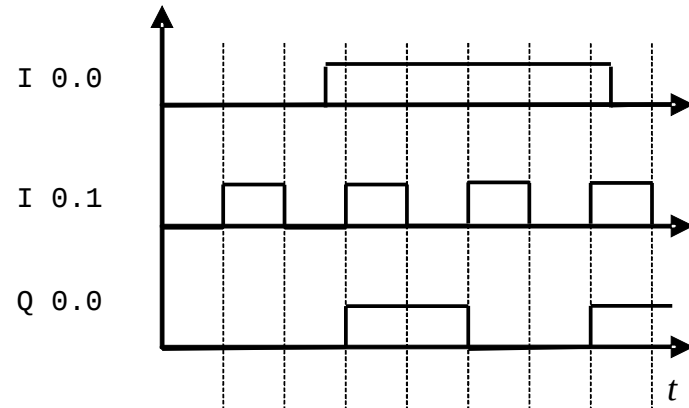
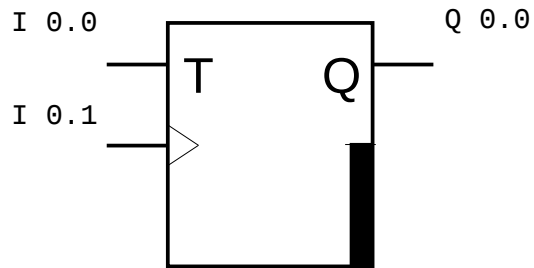


AN M 0.2
A I 0.3
R M 0.0
O I 0.0
ON I 0.1
O I 0.2
S M 0.0
AN M 0.0
= Q 0.0



DK – STEP 7

Pomnilne funkcije - T



```

A I 0.1
FP M 0.0 // dinamični člen za vhod I 0.1
= M 0.1 // spremembo priredimo M 0.1 za nadaljnjo uporabo

```

```

A I 0.0 // če je vhod T
A M 0.1 // in če je na I 0.1 sprememba z "0" na "1"
AN Q 0.0 // in če je izhod Q na "0"
S Q 0.0 // setiraj izhod Q

```

```

A I 0.0 // če je vhod T
A M 0.1 // in če je na I 0.1 sprememba z "0" na "1"
A Q 0.0 // in če je izhod Q na "1"
R Q 0.0 // potem resetiraj izhod Q

```

```

A I 0.1
FP M 0.0
= M 0.1

```

```

A I 0.0
A M 0.1
AN Q 0.0
S Q 0.0
BEC

```

```

A I 0.0
A M 0.1
A Q 0.0
R Q 0.0

```



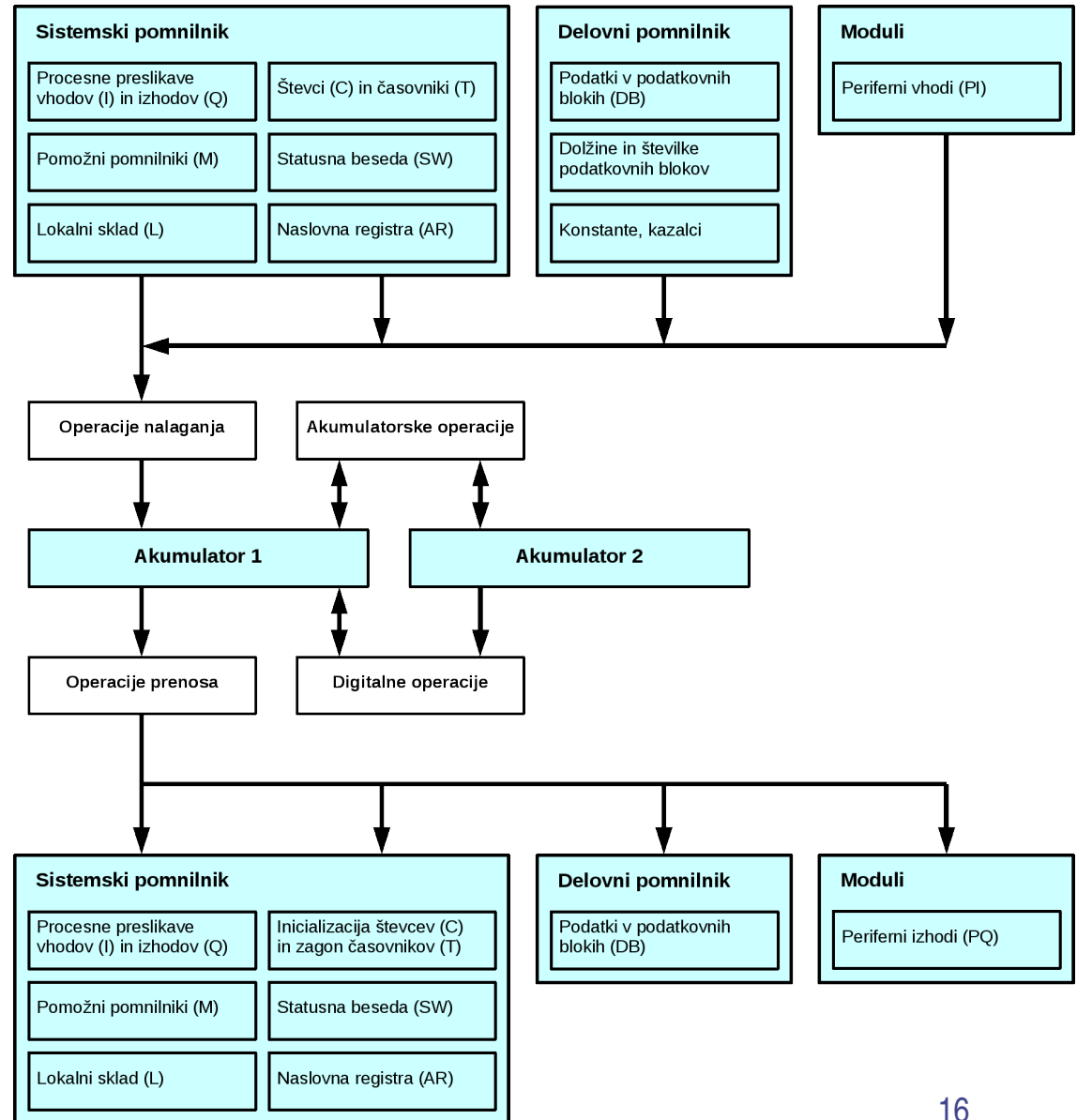
DK – STEP 7

AKUMULATOR

- AKU1
- AKU2

Naslavljanje:

- **takojšnje**
L konstanta
- **neposredno**
L naslov



DK – STEP 7

AKUMULATOR - 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 prekopodročni 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



DK – STEP 7

◆ AKUMULATOR – LOAD

- Kako do AKU2 ???
- Absolutna operacija (neodvisna os RLO)!!!

◆ NAROBE:

```
A I 1.0 // če je vkl. stikalo
L IB 0 // naloži v AKU1 vsebino vhodnega zloga 0
```

◆ PRAVILNO:

```
A I 1.0 // če je vkl. stikalo
JC sk1 // skoči na nalaganje
JU sk2 // sicer brezpogojno skoči naprej!!!
sk1: L IB 0 // naloži v AKU1 vsebino vhodnega zloga 0
...
sk2: ... // nadaljnji ukazi
```

DK – STEP 7

AKUMULATOR – 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

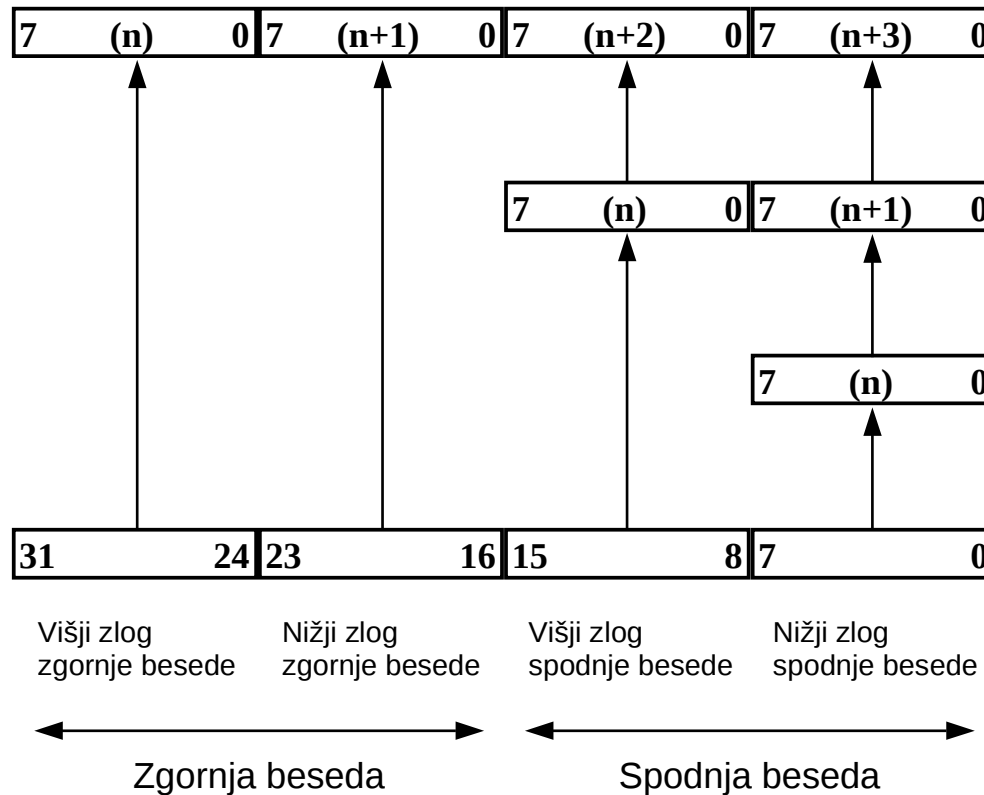
Tabela 7.2: Zgledi nalaganja v akumulator za operande, ki imajo različno podatkovno širino



DK – STEP 7

AKUMULATOR – TRANSFER

T naslov



Dvojna beseda n (D)

Beseda n (W)

Zlog n (B)

Akumulator 1

DK – STEP 7

AKUMULATOR – TRANSFER

- Prav tako *absolutna* operacija (neodvisna os RLO)!!!
- Nanaša se na AKU1, ki se ne spremeni; na AKU2 ni vpliva

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



DK – STEP 7

◆ AKUMULATOR – ostale operacije

```
... // prej          AKU1: 16#89AB_CDEF   AKU2: 16#0123_4567
PUSH
... // potom        AKU1: 16#89AB_CDEF   AKU2: 16#89AB_CDEF

... // prej          AKU1: 16#89AB_CDEF   AKU2: 16#0123_4567
POP
... // potom        AKU1: 16#0123_4567   AKU2: 16#0123_4567

... // prej          AKU1: 16#89AB_CDEF   AKU2: 16#0123_4567
TAK
... // potom        AKU1: 16#0123_4567   AKU2: 16#89AB_CDEF

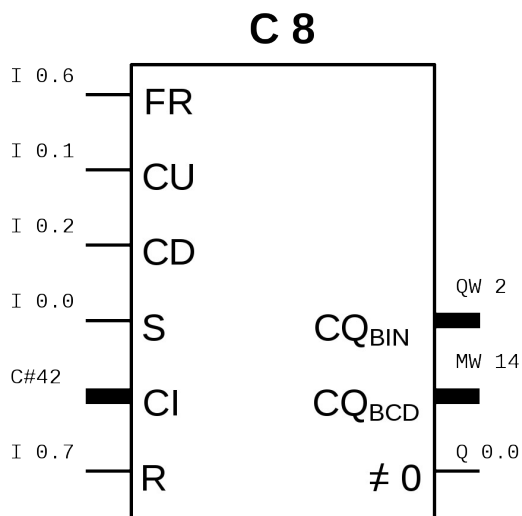
... // prej          AKU1: 16#89AB_CDEF   AKU2: 16#0123_4567
CAW
... // potom        AKU1: 16#89AB_EFCD   AKU2: 16#0123_4567

... // prej          AKU1: 16#89AB_CDEF   AKU2: 16#0123_4567
CAD
... // potom        AKU1: 16#EFCD_AB89   AKU2: 16#0123_4567
```

DK – STEP 7

ŠTEVEC

- naslovi C0 – C63
- dogodki: 0-999



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			

```

A I 0.6
FR C 8 // sprostitvev

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
    
```




DK – STEP 7

◆ ŠTEVEC

- pozor na oznake C, C#, LC....
- ukaza L in LC (!!!)

npr. števec C 4 : 498 dogodkov



DK – STEP 7

◆ Mimogrede: PRIREDITVENA TABELA

- **OBVEZNA!!!**
- **POZOR: to ni kombinacijska (tudi pravilnostna, izjavnostna) tabela !!!**

Št.	Naslov	Enota	Teh nol. 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	odštevava pretočene litre
7	M 0.0	pomn.	–	pomožni pomnilnik za DLF	pri pogoju za odprtje EV

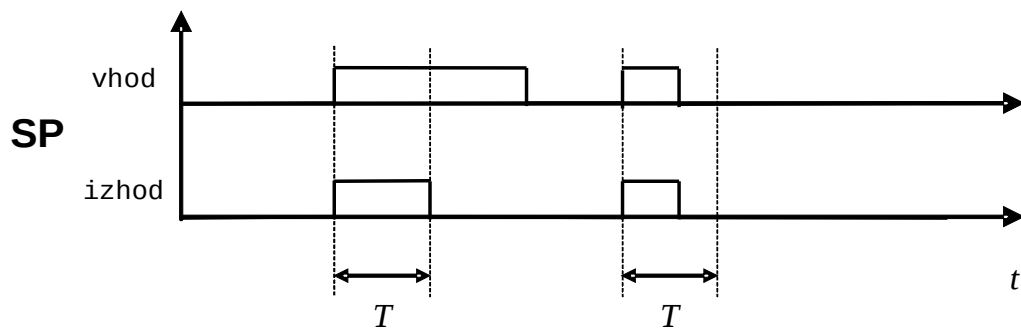
Tabela 7.4: Prireditvena tabela za krmilje pri procesu doziranja tekočine

DK – STEP 7

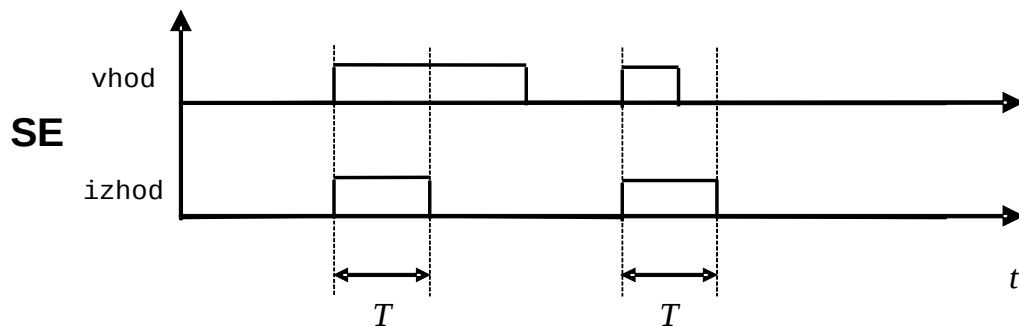
ČASOVNIK

- naslovi: T0 – T127
- čas: 10 ms – 9990 s (2H46M30s)
- različne časovne funkcije:

Skrajšanje pulza (SP):



Podaljšanje pulza (SE):



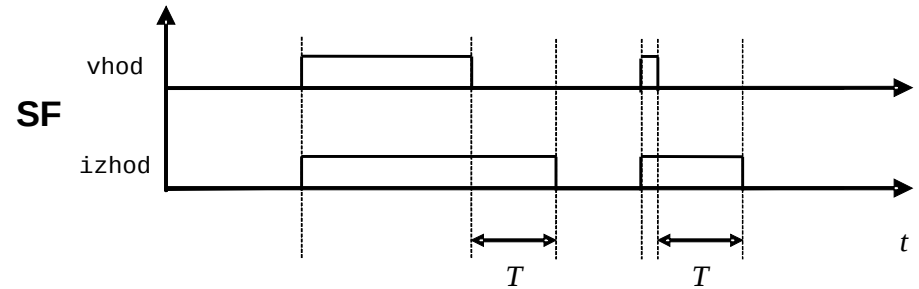


DK – STEP 7

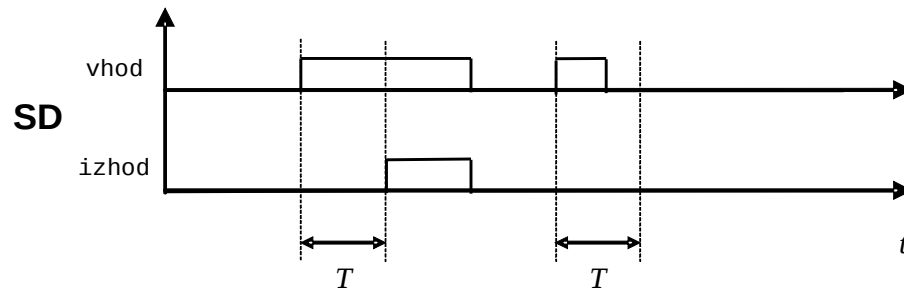
ČASOVNIK

- časovne funkcije:

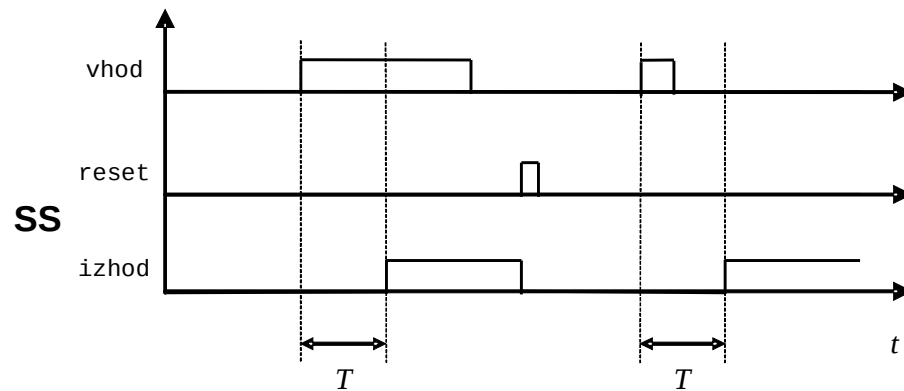
Zakasnitev izklopa (SF):



Zakasnitev vklopa (SD):



Zakasnitev vklopa s spominom (SS):

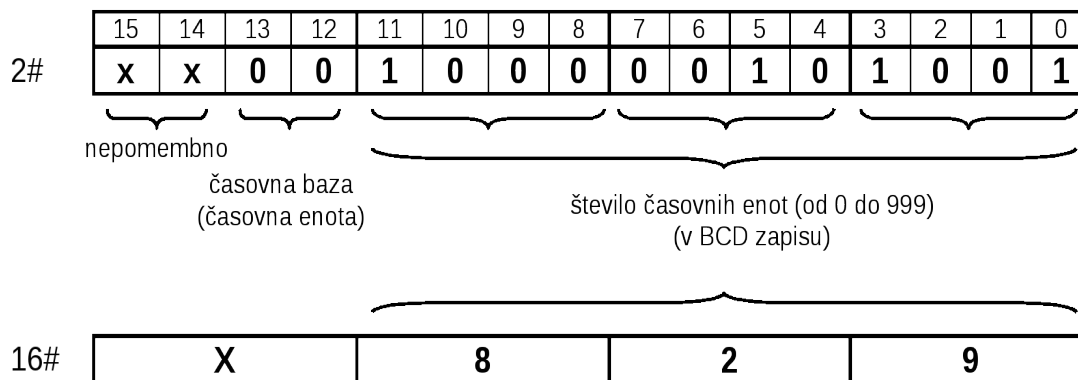


DK – STEP 7

ČASOVNIK - format

```
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"
```

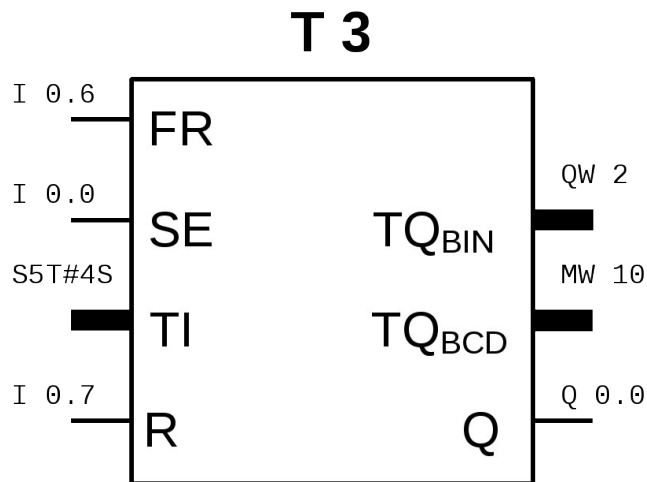
S5T# časovniški format



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

DK – STEP 7

ČASOVNIK - zgled



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

```
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
```



DK – STEP 7

STATUSNA BESEDA

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

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

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



DK – STEP 7

SKOČNI UKAZI

- znotraj FB !!!
- na labelo!
- pomni:
 - ◆ JC
 - ◆ JU
 - ◆ JCN
- zgled str. 177 !!!

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***	–	–	–	–	–	–	–	–	–	–	–



DK – STEP 7

◆ BLOKOVNI UKAZI

- BE (na koncu)
- BEU - brezpogojni zaključek bloka (vmes)
- BEC - pogojni zaključek bloka (vmes)
 - ◆ JC
 - ◆ JU
 - ◆ JCN
- zgled str. 177 !!!

◆ KLICI (po zaključku sledi povratek na ukaz, ki sledi klicu):

- pogojni (RLO): CC FB3
- brezpogojni: UC FB5

◆ "Prazna ukaza"

- NOP 0
- NOP1

(a zasedata pomnilnik in potrebujeta čas za izvajanje!!!)



DK – STEP 7

BLOKOVNI UKAZI – ZGLED

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   sko1      // 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!
```

```
sko1: 0   I 0.1      // če je I 0.1 vklopljeno
        0   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
```