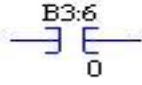


Bitové inštrukcie

XIC (Zistiť, či je bit zatvorený)



XIC inštrukcia slúži na zistenie, či je bitové slovo danej adresy (na ukážke sa jedná o adresu B3:6 a ku nej prislúchajúcemu bitu) nastavený na logickú jednotku, respektíve zapnutý . Ak podmienka je splnená, tak stav rungu sa nastaví na priechodný a vykonajú sa nasledujúce podmienky a ak podmienka nie je splnená, tak stav rungu sa nastaví na nepriechodný a nevykonajú sa nasledujúce inštrukcie.

Použitie: Tlačidlá; prepínače; koncové a proxy spínače; svetlá a led žiarovky; vnútorné bity; vstupné a výstupné bity

XIO (Zistiť, či je bit otvorený)



XIO inštrukcia slúži na zistenie, či je bitové slovo danej adresy nastavený na logickú nulu. Ak je daná podmienka splnená, tak stav rungu sa nastaví na priechodný a ak nie je splnená, tak stav rungu sa nastaví na nepriechodný a nasledujúce inštrukcie sa nevykonajú.

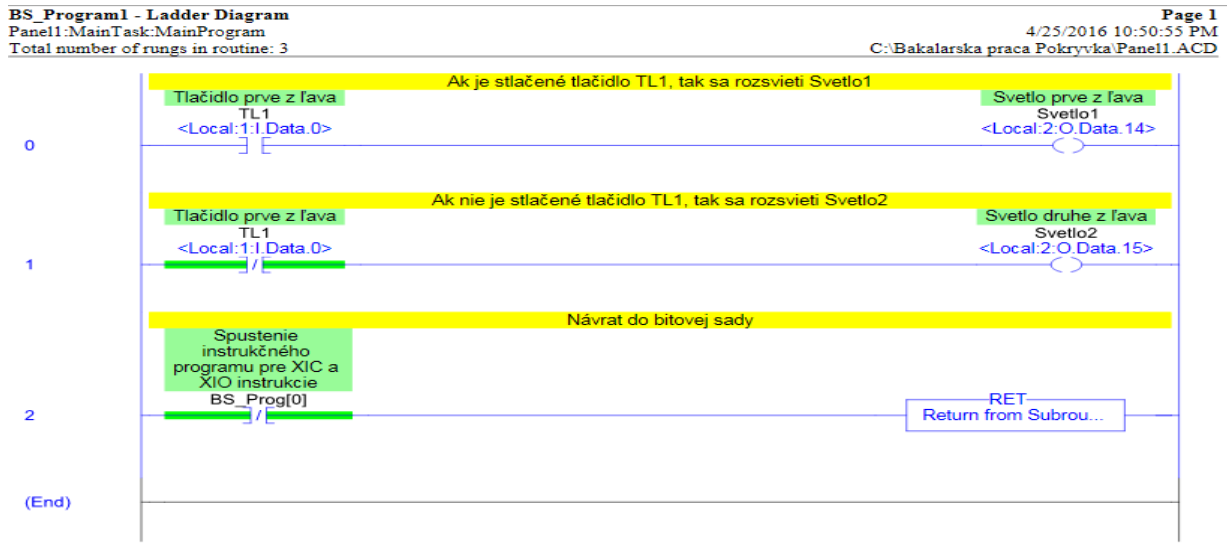
Použitie: Tlačidlá; prepínače; koncové a proxy spínače; svetlá a led žiarovky; vnútorné bity; vstupné a výstupné bity

BS_Program1

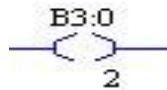
Úloha: Úlohou programu je jednoznačne poukázať na rozdiel funkcionality XIC a XIO inštrukcií.

Riešenie: Tento rozdiel bude riešený pomocou dvoch svetiel a tlačidla. Program sa zapína pomocou programovej premennej BS_Prog[0] a vykonáva sa dokým tato premenná nebude vynulovaná. Vykonáva jednoduchú svetelnú ukážku v prepínaní stavu výstupných bitov dvoch svetiel pomocou základných podmienených inštrukcií XIC a XIO.

Rebríková schéma podprogramu BS_Program1:



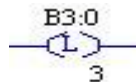
OTE (Dodat' energiu výstupu)



OTE inštrukcia slúži na nastavenie bitového slova na ktoré ukazuje zadaná adresa. Bit sa nastaví na logickú 1 ak stav rungu je priechodný a ak stav rungu je nepriechodný tak bit sa nastaví na logickú 0.

Použitie: svetlá a led žiarovky; vnútorné bity; výstupné bity

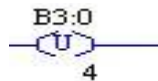
OTL (Výstup uzavrieť)



OTL inštrukcia slúži na nastavenie logickej 1 na bitovom slove na ktoré ukazuje zadaná adresa. Ak stav rungu je priechodný, tak sa inštrukcia vykoná a bit sa preklopí na logickú 1 a ak je stav rungu nepriechodný, tak inštrukcia sa nevykoná a bit zachová pôvodný stav. Ak bit je nastavený na logickú 1, tak stav bitu sa zachová.

Použitie: svetlá a led žiarovky; vnútorné bity; výstupné bity

OTU (Výstup otvoriť)



OTU inštrukcia slúži na nastavenie logickej 0 na bite na ktorý ukazuje adresa. Ak stav rungu je priechodný, tak sa inštrukcie vykoná a bit sa nastaví na logickú 0, a ak je stav rungu nepriechodný, tak sa inštrukcia nevykoná a stav bitu ostane zachovaný.

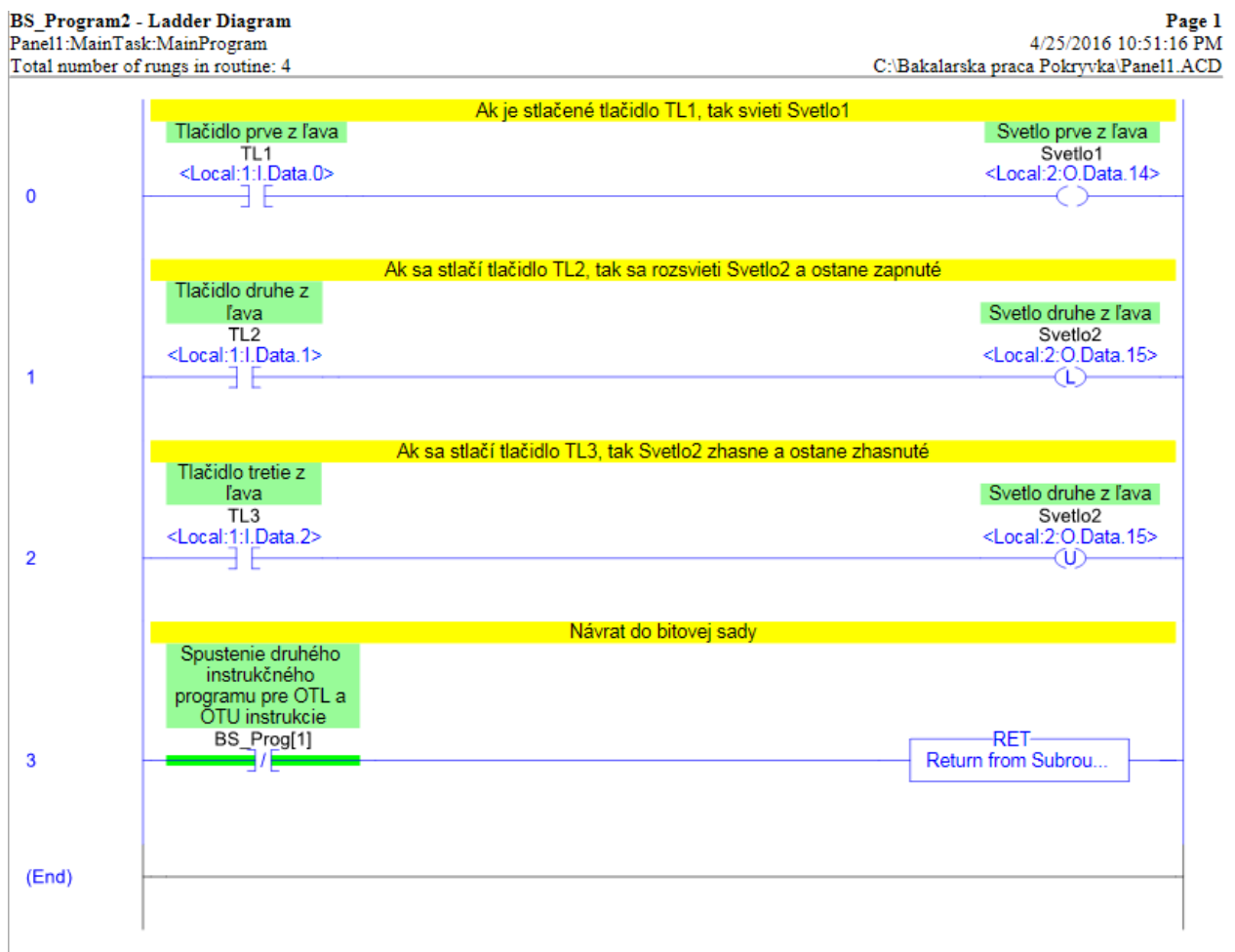
Použitie: svetlá a led žiarovky; vnútorné bity; výstupné bity

BS_Program2

Úloha: Úlohou programu je jednoznačne poukázať na rozdiel funkcionality OTE inštrukcie oproti párovým OTL/OTU inštrukciám.

Riešenie: Tento rozdiel bude riešený pomocou dvoch svetiel a troch tlačidiel. Program sa zapína pomocou programovej premennej BS_Prog[1] a vykonáva sa dokým tato premenná nebude vynulovaná. Vykonáva jednoduchú svetelnú ukážku v prepínaní stavu výstupných bitov dvoch svetiel pomocou rozličných výstupných inštrukcií.

Rebríková schéma podprogramu BS_Program2 :



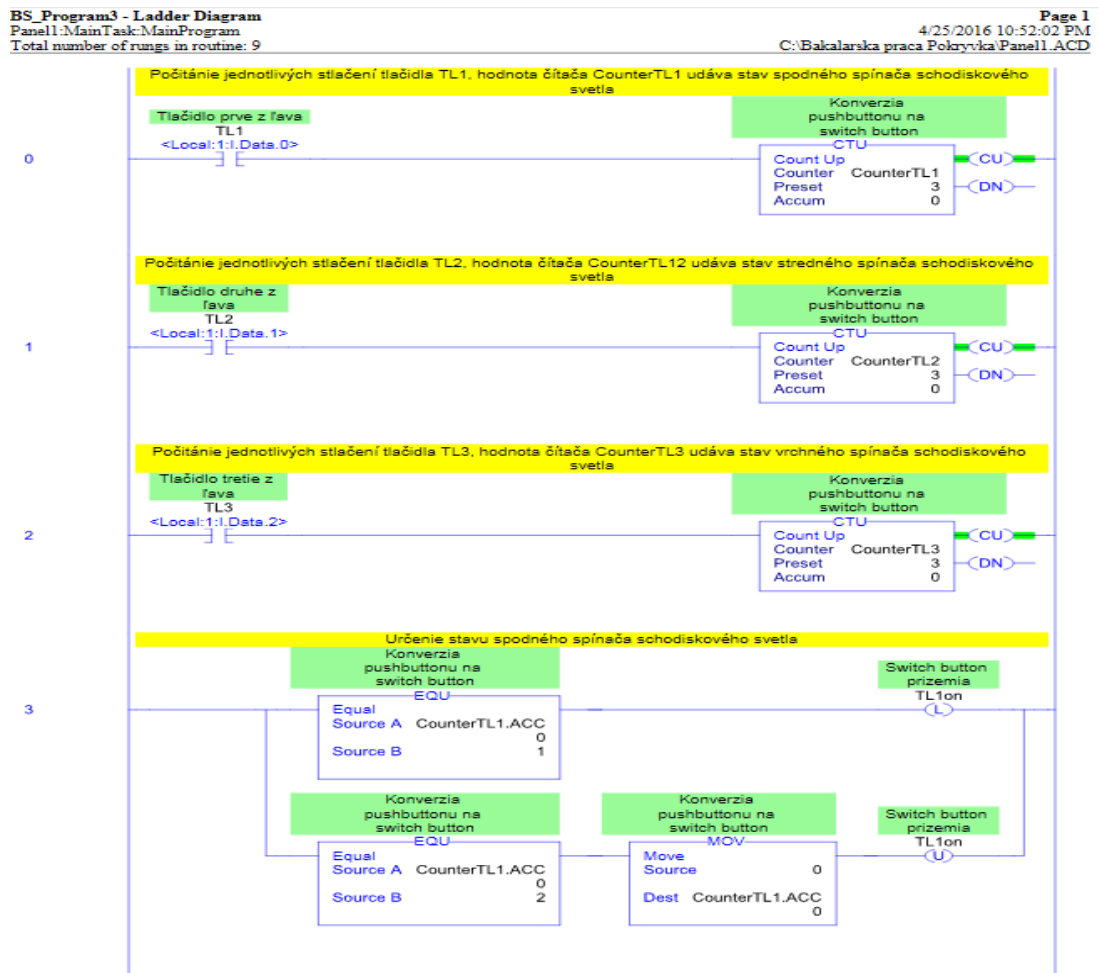
BS_Program3

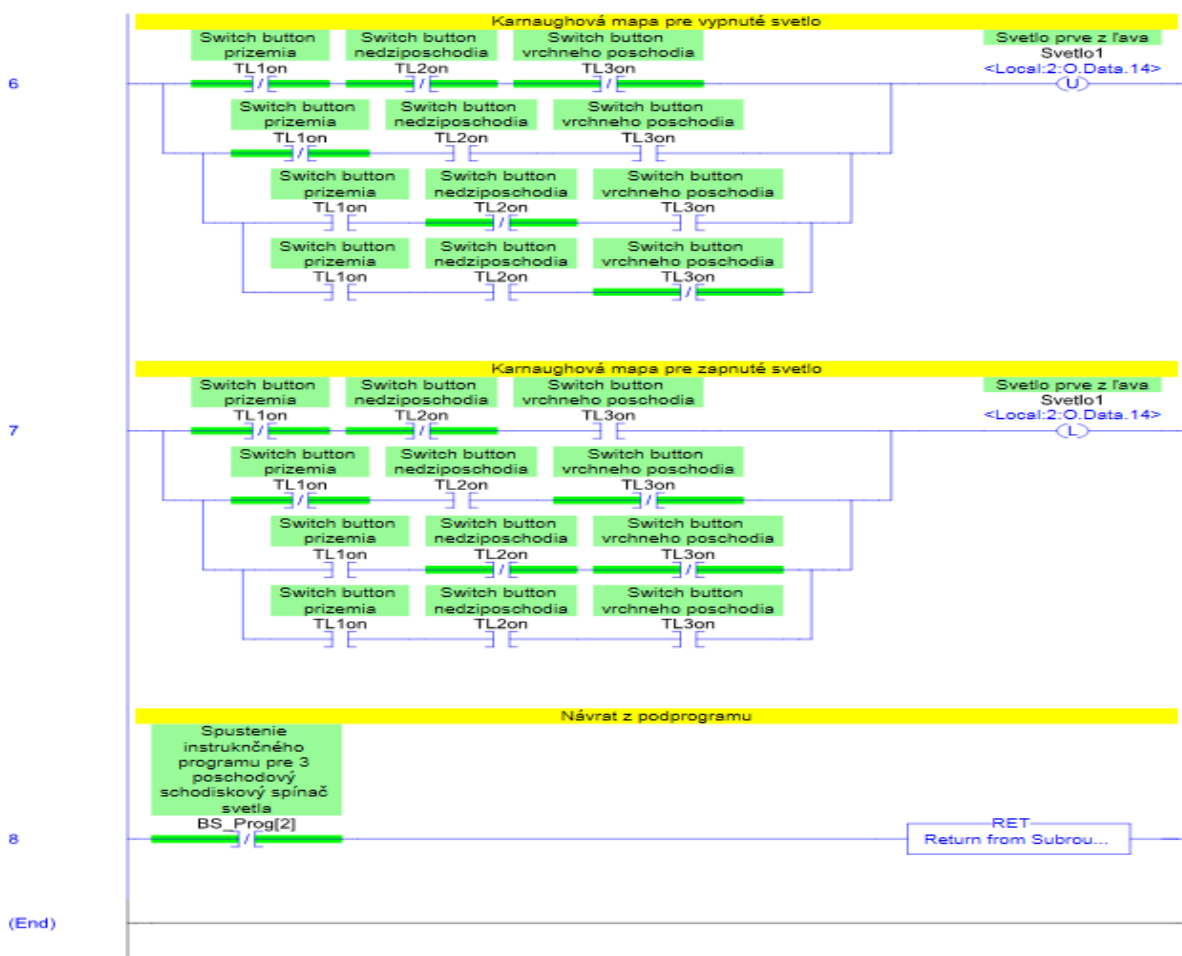
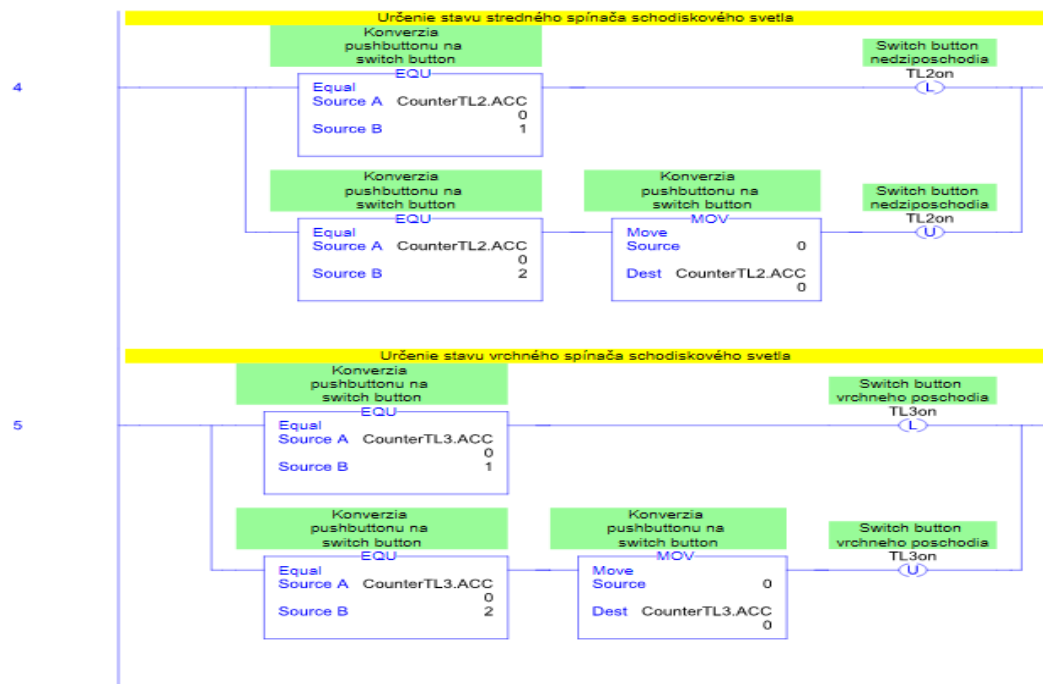
Úloha: Úlohou programu je vytvorenie schodiskového vypínača pre 3 poschodia.

Riešenie: Táto úloha bude riešená pomocou jedného svetla a troch tlačidiel. Program sa zapína pomocou programovej premennej BS_Prog[2] a vykonáva sa dokým tato premenná nebude vynulovaná. Pri tejto úlohe je nutné programovo zmeniť tlačidlá z push tlačidiel na switch tlačidlá, aby sme zachovávali stav, ako pri bežnom vypínači svetla. Problém troch vypínačov riešime pomocou Karnaughovej mapy, ktorá ma v našom prípade tvar:

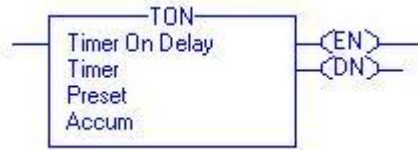
TLon1	TLon2	TLon3	SVETLO1
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Rebríková schéma podprogramu BS_Program3:





Časovače a čítače



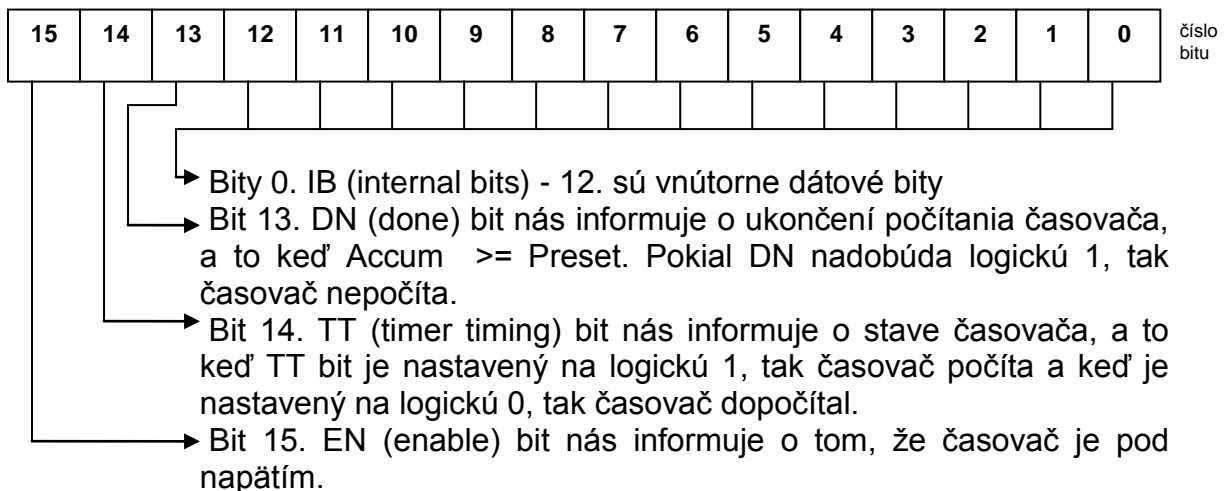
TON (časovač na oneskorenie)

Inštrukcia počíta časové intervaly, ak stav rungu nadobúda priechodný stav. Pokiaľ stav rungu ostáva priechodný, tak časovač zväčšuje hodnotu v akumulátore (ACC bite), až pokiaľ nenadobudne rovnakú hodnotu akú má v preset bite časovača. Ak časovač dopočíta, tak hodnota v DN (done) bite nadobudne logickú 1 a časovač ďalej nebude počítať. Časovač začne znova počítať, až keď DN bit sa nastaví na logickú 0. DN bit sa dá nastaviť do logickej 0 inštrukciou reset, ktorou reštartujeme časovač, alebo pomocou inštrukcie MOV, kde priamo prepisujeme hodnotu zadanej adresy bitu. Ak počas počítania časovača sa zmení stav rungu z priechodného na nepriechodný, tak akumulovaná hodnota sa resetuje. Časovač sa používa na nastavovanie výstupných alebo vnútorných bitov po uplynutí prednastaveného času, pokiaľ stav rungu ostáva v priechodnom stave.

Každý časovač na oneskorenie sa skladá z troch vnútorných slov.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	číslo bitu
Slovo 0	E N	TT	D N	IB	IB	IB	IB	IB	IB	IB	IB	IB	IB	IB	IB	IB	
Slovo 1	Prednastavená hodnota (Preset)																
Slovo 2	Akumulátor (Accum)																

1.slovo - Riadiace slovo:



2.slovo - ukladá nastavenú hodnotu (preset bit):

Programátor zadáva túto hodnotu. Ak akumulovaná hodnota dosiahne zadanú hodnotu alebo je väčšia ako zadaná hodnota, tak časovač prestane počítat' a nastaví sa DN bit. Táto hodnota môže byť zadaná v rozsahu: 0 - 32 767, a ak je mimo tohto rozsahu dôjde ku chybe.

3.slovo - ukladá akumulovanú hodnotu (Accum bit):

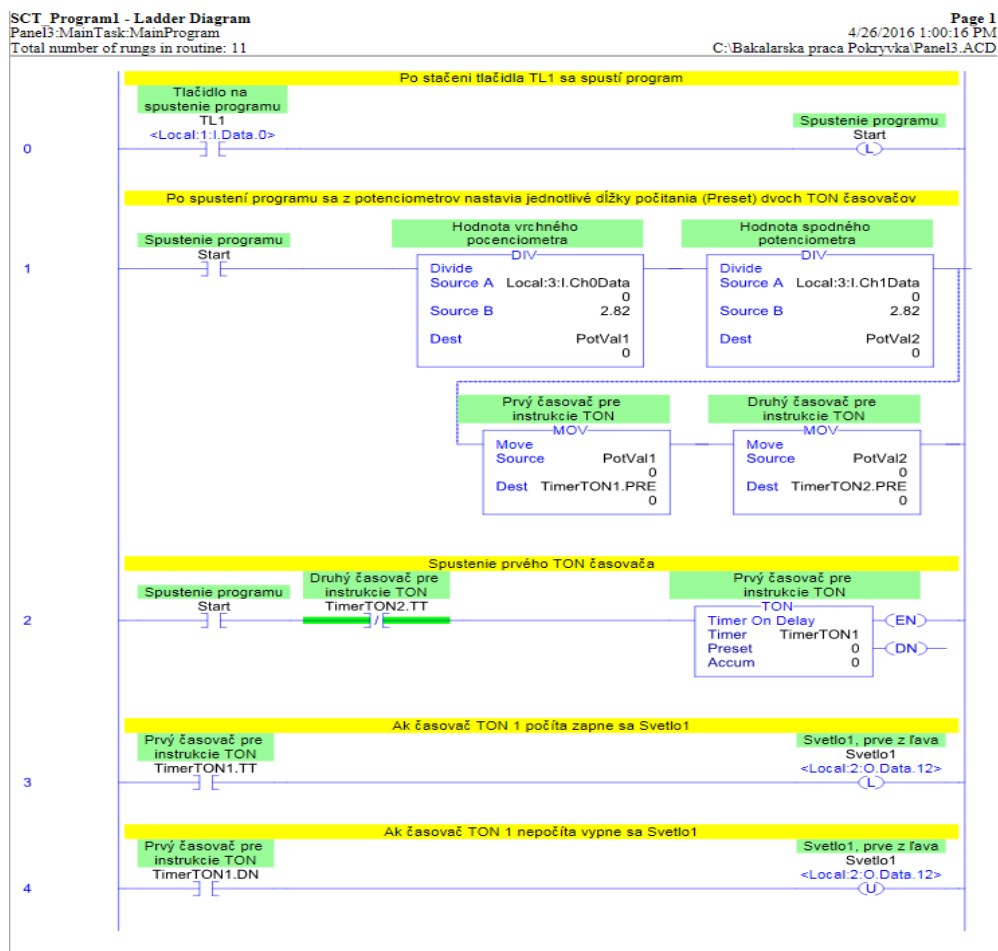
Je to hodnota ktorá narastá od posledného reštartovania a spustenia časovača. Každá jej hodnota predstavuje 1milisekundu.

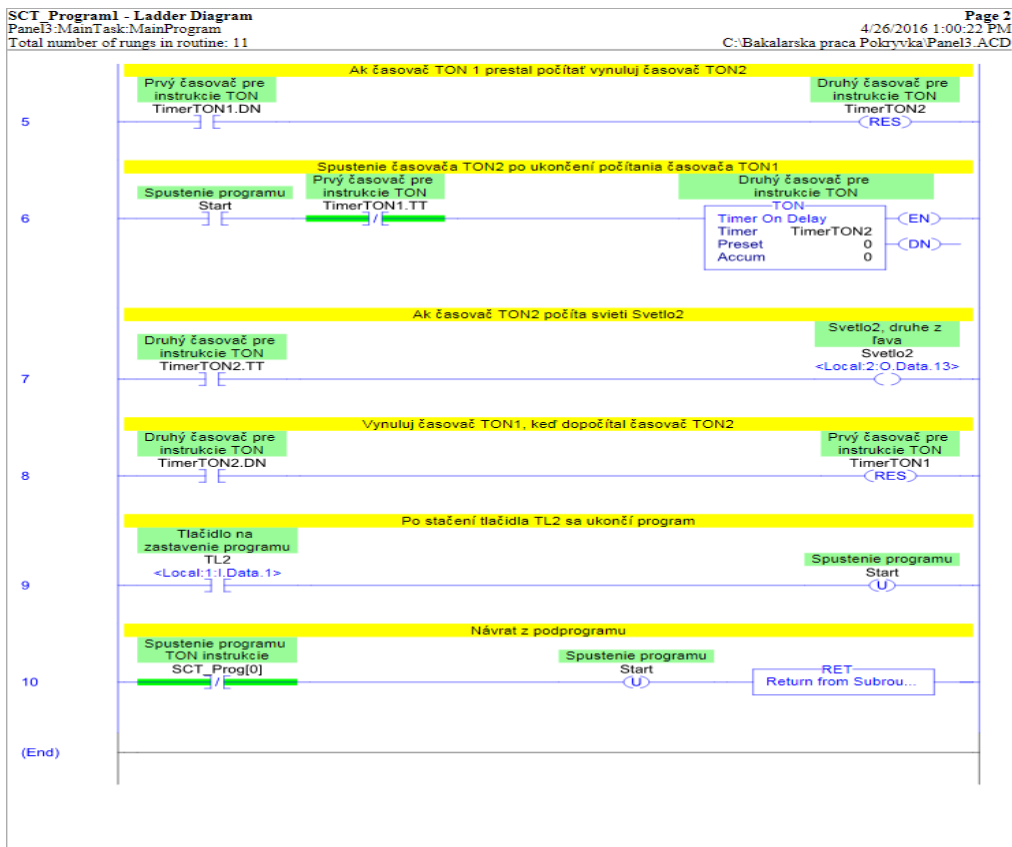
SCT_Program1

Úloha: Úlohou programu je znázornenie funkcionality inštrukcie TON, ktorá predstavuje softvérový časovač, ktorý sa inkrementuje až po zadanú hodnotu.

Riešenie: Úlohu znázorňujeme na príklade preklápacieho obvodu, ktorý zapína jedno z dvoch svetiel, pričom rýchlosť prepínania svetiel sa nastavuje podľa potenciometra. Program zapíname stlačením tlačidla TL1 a zastavíme ho stlačením tlačidla TL2. Do podprogramu SCT_Program1 vstúpime, ak stav premennej SCT_Prog[0] nadobúda logickú jednotku.

Rebríková schéma podprogramu STC Program1:



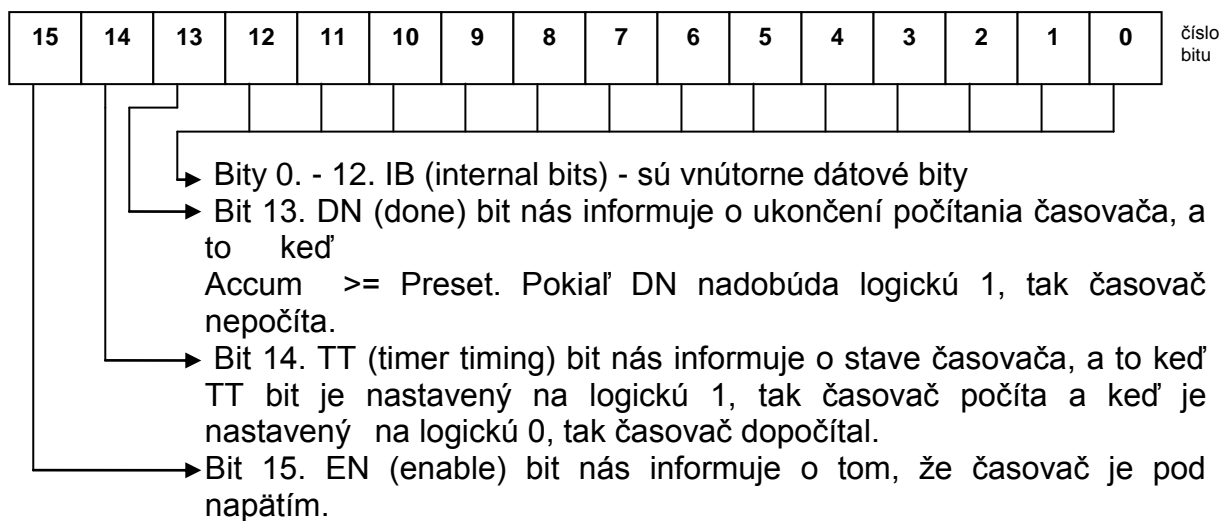


TOF (časovač oneskorenia vypnutia)

Inštrukcia počíta časové intervaly, ak stav rungu nadobúda nepriechodný stav. TOF časovač začne počítat' ak sa stav rungu zmení z priechodného na nepriechodný. Pokiaľ stav rungu ostáva nepriechodný, tak časovač zväčšuje hodnotu v akumulátore (Accum bite), až pokiaľ nenadobudne rovnakú hodnotu akú ma v preset bite časovača. Ak časovač dopočíta, tak hodnota v DN (done) bite nadobudne logickú 1 a časovač ďalej nebude počítat'. Časovač začne znova počítat', až keď DN bit sa nastaví na logickú 0. Ak počas počítania časovača sa zmení stav rungu z nepriechodného na priechodný, tak akumulovaná hodnota sa resetuje. Časovač sa používa na nastavovanie výstupných alebo vnútorných bitov po uplynutí prednastaveného času, pokiaľ stav rungu ostáva v nepriechodnom stave. Každý časovač oneskorenia vypnutia sa skladá z troch vnútorných slov.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	číslo bitu
Slovo 0	E N	TT	D N	IB	IB	IB	IB	IB	IB	IB	IB	IB	IB	IB	IB	IB	
Slovo 1	Prednastavená hodnota (Preset)																
Slovo 2	Akumulátor (Accum)																

1.slovo - Riadiace slovo:



2.slovo - ukladá nastavenú hodnotu (preset bit):

Programátor zadáva túto hodnotu. Ak akumulovaná hodnota dosiahne zadanú hodnotu alebo je väčšia ako zadaná hodnota, tak časovač prestane počítať a nastaví sa DN bit. Táto hodnota môže byť zadaná v rozsahu: 0 - 32 767, a ak je mimo tohto rozsahu dôjde ku chybe.

3.slovo - ukladá akumulovanú hodnotu (Accum bit):

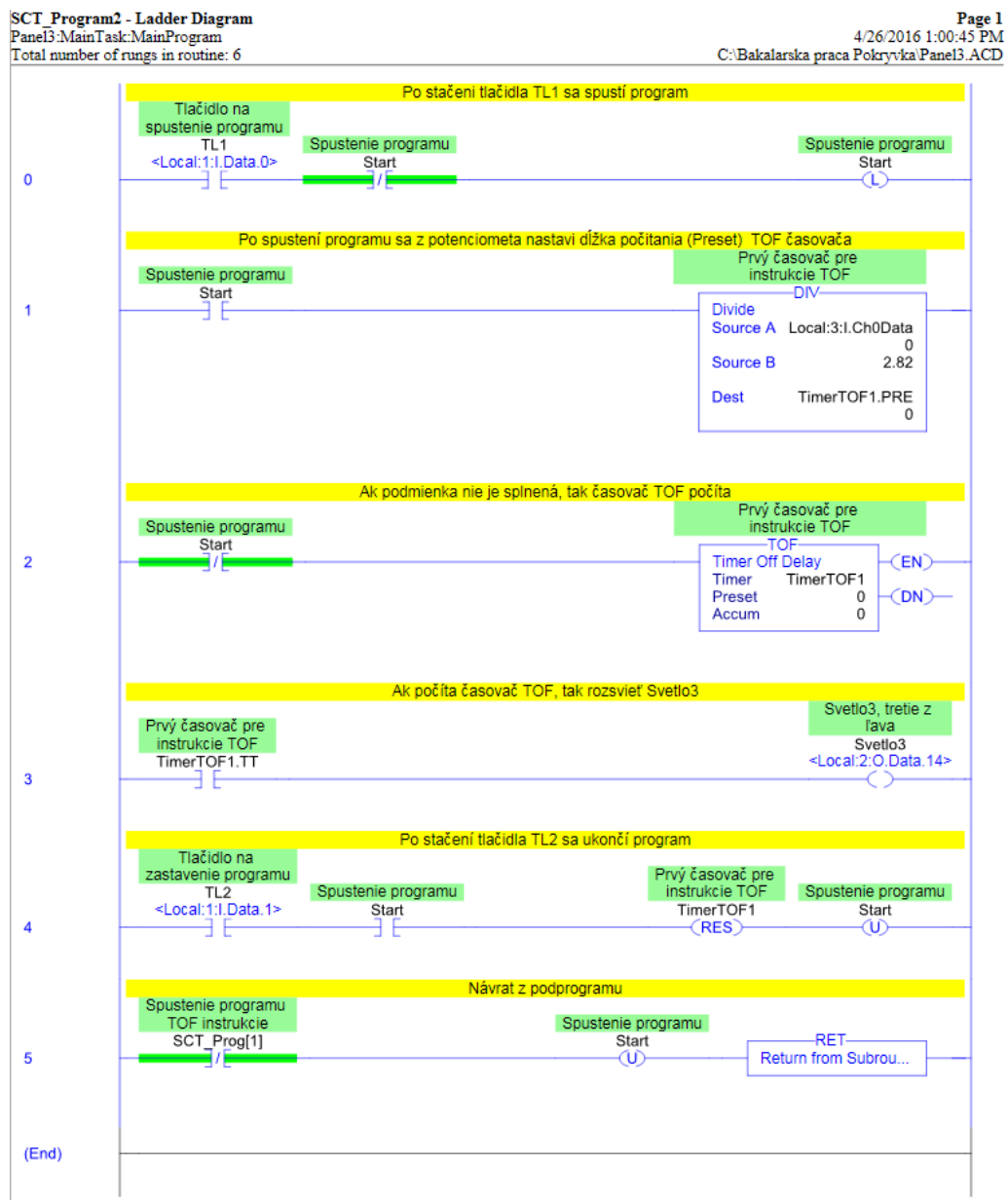
Je to hodnota ktorá narastá od posledného reštartovania a spustenia časovača. Každá jej hodnota predstavuje 1 milisekundu.

SCT_Program2

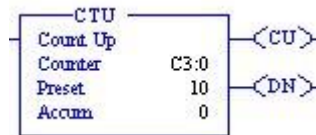
Úloha: Úlohou programu je znázornenie funkcionality inštrukcie TOF, ktorá predstavuje softvérový časovač, ktorý dekrementuje zadanú hodnotu až po nulu.

Riešenie: Úlohu znázorňujeme na príklade jednoduchého zapínania svetla po dobu počítania časovača, pričom dĺžka počítania sa dá zadať pomocou potenciometra. Program zapíname stlačením tlačidla TL1 a zastavíme ho stlačením tlačidla TL2. Do podprogramu SCT_Program2 vstúpime, ak stav premennej SCT_Prog[1] nadobúda logickú jednotku.

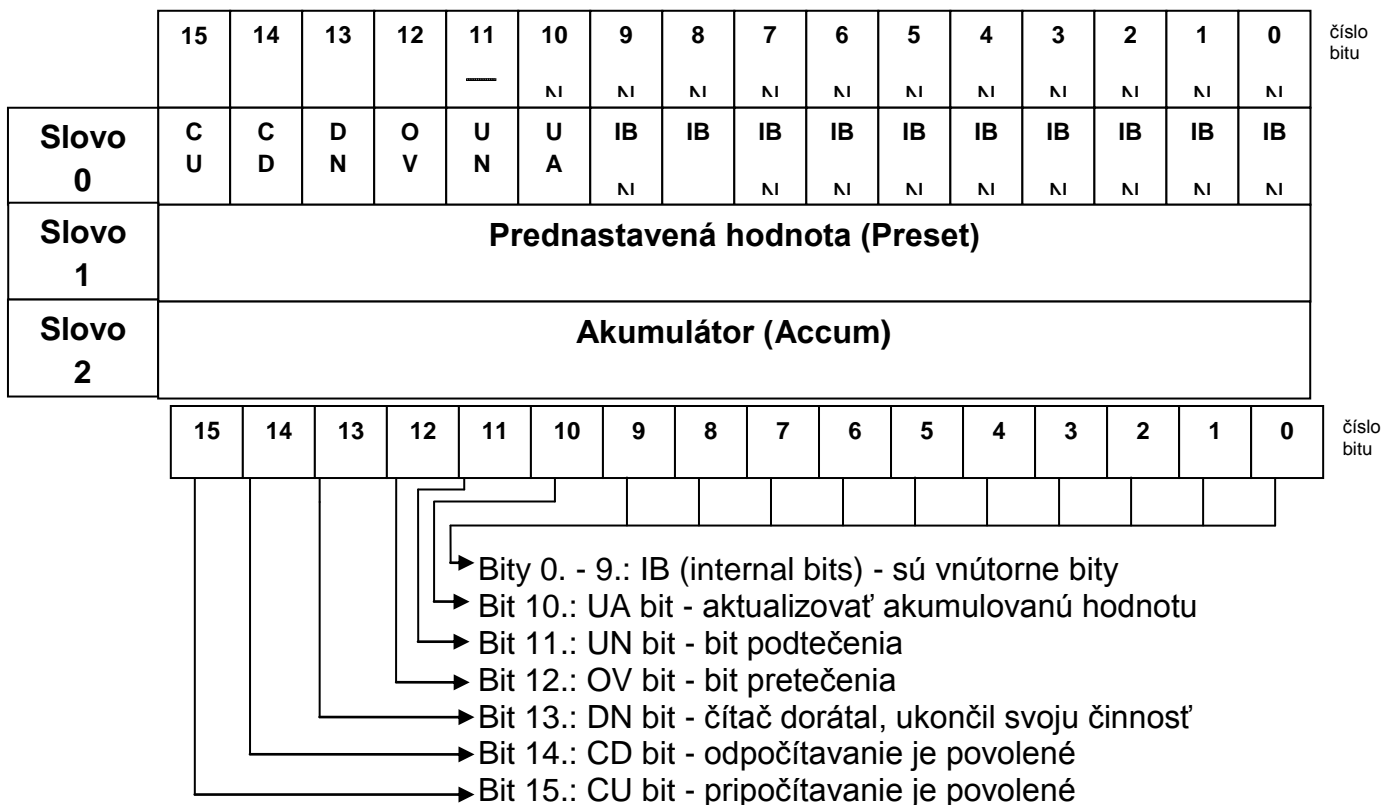
Rebríková schéma podprogramu STC_Program2:



CTU (čítač nahor)



Inštrukcia počíta zmeny stavu rungu z nepriechodného do priechodného a uchováva túto hodnotu. Ak nastane daná zmena stavu rungu tak hodnota v akumulátore narastie o 1. Ak akumulovaná hodnota dosiahne zadanú hodnotu alebo ju prerastie, tak sa nastaví DN bit čítača a čítač by mal prestať počítať. Akumulovaná hodnota je vynulovaná pomocou inštrukcie RES alebo priamym prepísaním hodnoty pomocou inštrukcie MOV.



2.slovo - ukladá prednastavenú hodnotu (preset bit)

Špecifikuje hodnotu, ktorú hodnota v akumulátore musí dosiahnuť aby bol nastavený DN bit. Túto hodnotu zadáva programátor a môže byť zadaná od -32 767 do 32 767. Ak zadaná hodnota je mimo tohto rozsahu dôjde ku chybe.

3.slovo - ukladá akumulovanú hodnotu (Accum bit)

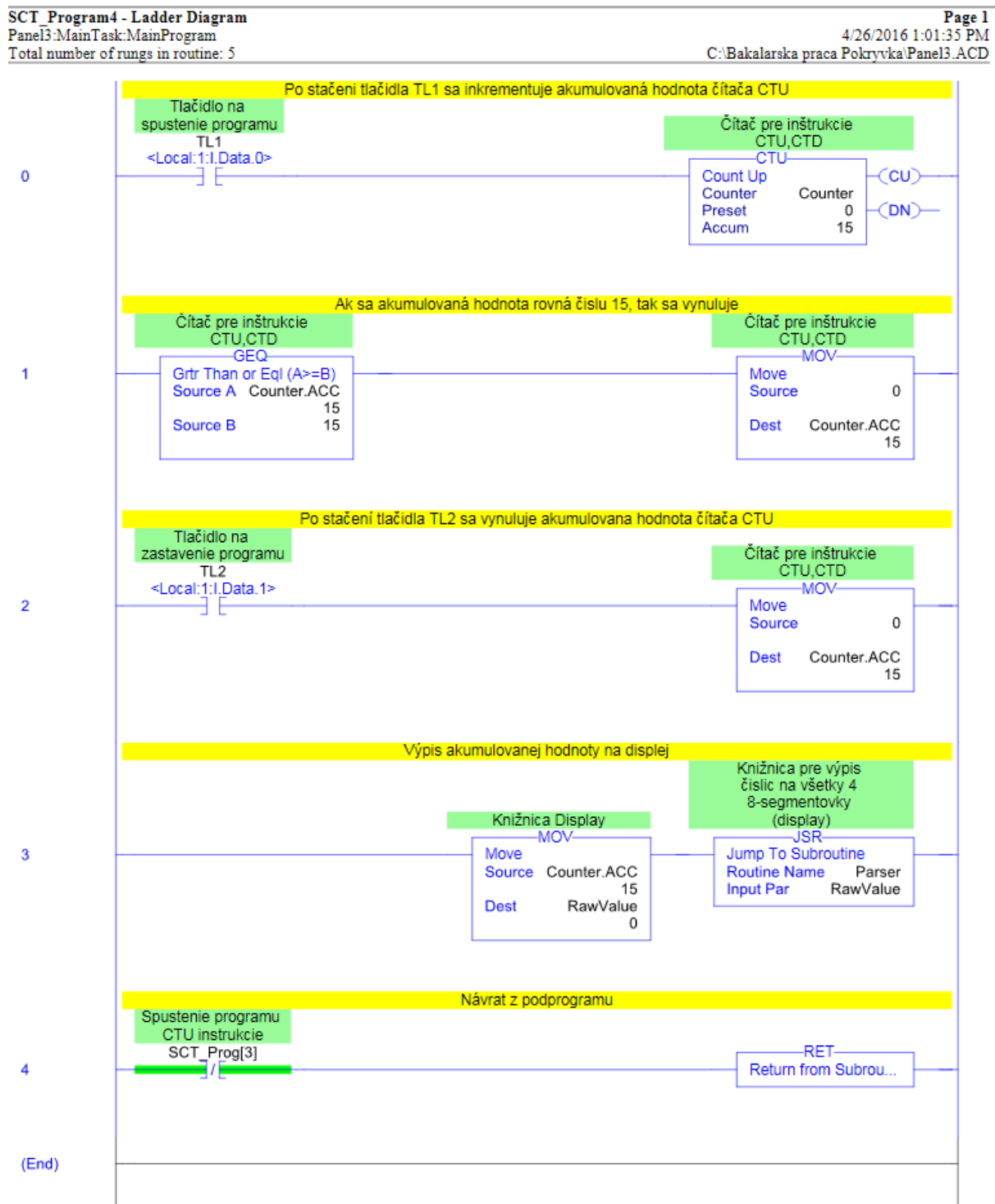
Je to počet zmien stavov rungu z nepriechodného na priechodný, na ktorom sa nachádza inštrukcia CTU.

SCT_Program4

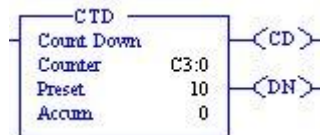
Úloha: Úlohou programu je znázornenie funkcionality inštrukcie CTU, ktorá predstavuje softvérový, ktorý inkrementuje akumulovanú hodnotu až po zadanú hodnotu .

Riešenie: Úlohu znázorňujeme na príklade počítadla stlačení tlačidla TL1, kde sa akumulovaná hodnota inkrementuje až po hodnotu 15, ak sa prekročí, tak sa vynuluje. Pri stlačení tlačidla TL2 sa akumulovaná hodnota vynuluje. Do podprogramu SCT_Program4 vstúpime, ak stav premennej SCT_Prog[3] nadobúda logickú jednotku.

Rebríková schéma podprogramu SCT_Program4:



CTD (čítač nadol)

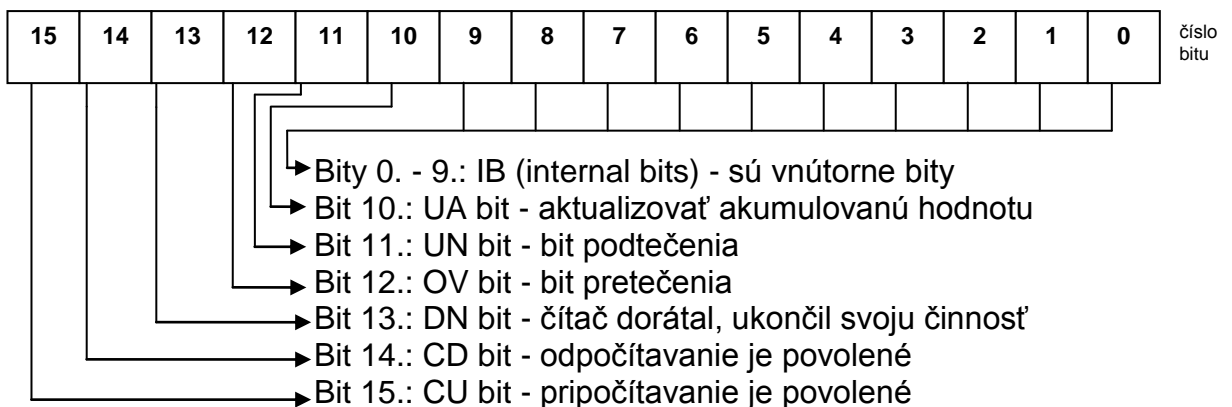


Inštrukcia počíta zmeny stavu rungu z nepriechodného do priechodného a uchováva túto hodnotu. Ak nastane daná zmena stavu rungu tak hodnota v akumulátore sa zmenší o 1. Ak akumulovaná hodnota je menšia ako minimálny rozsah, tak sa nastaví UN bit na logickú 1. Akumulovaná hodnota je vynulovaná pomocou inštrukcie RES alebo priamym prepísaním hodnoty pomocou inštrukcie MOV.

Každá adresa čítača nadol sa skladá z troch vnútorných slov:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	číslo bitu
					—	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	
Slovo 0	C	C	D	O	U	U	IB	IB	IB	IB	IB	IB	IB	IB	IB	IB	
	U	D	N	V	N	A											
Slovo 1	Prednastavená hodnota (Preset)																
Slovo 2	Akumulátor (Accum)																

1.slovo - riadiace slovo



2.slovo - ukladá prednastavenú hodnotu (preset bit)

Špecifikuje hodnotu, ktorú hodnota v akumulátore musí dosiahnuť aby bol nastavený DN bit. Túto hodnotu zadáva programátor a môže byť zadaná od - 32 767 do 32 767. Ak zadaná hodnota je mimo tohto rozsahu dôjde ku chybe.

3.slovo - ukladá akumulovanú hodnotu (Accum bit)

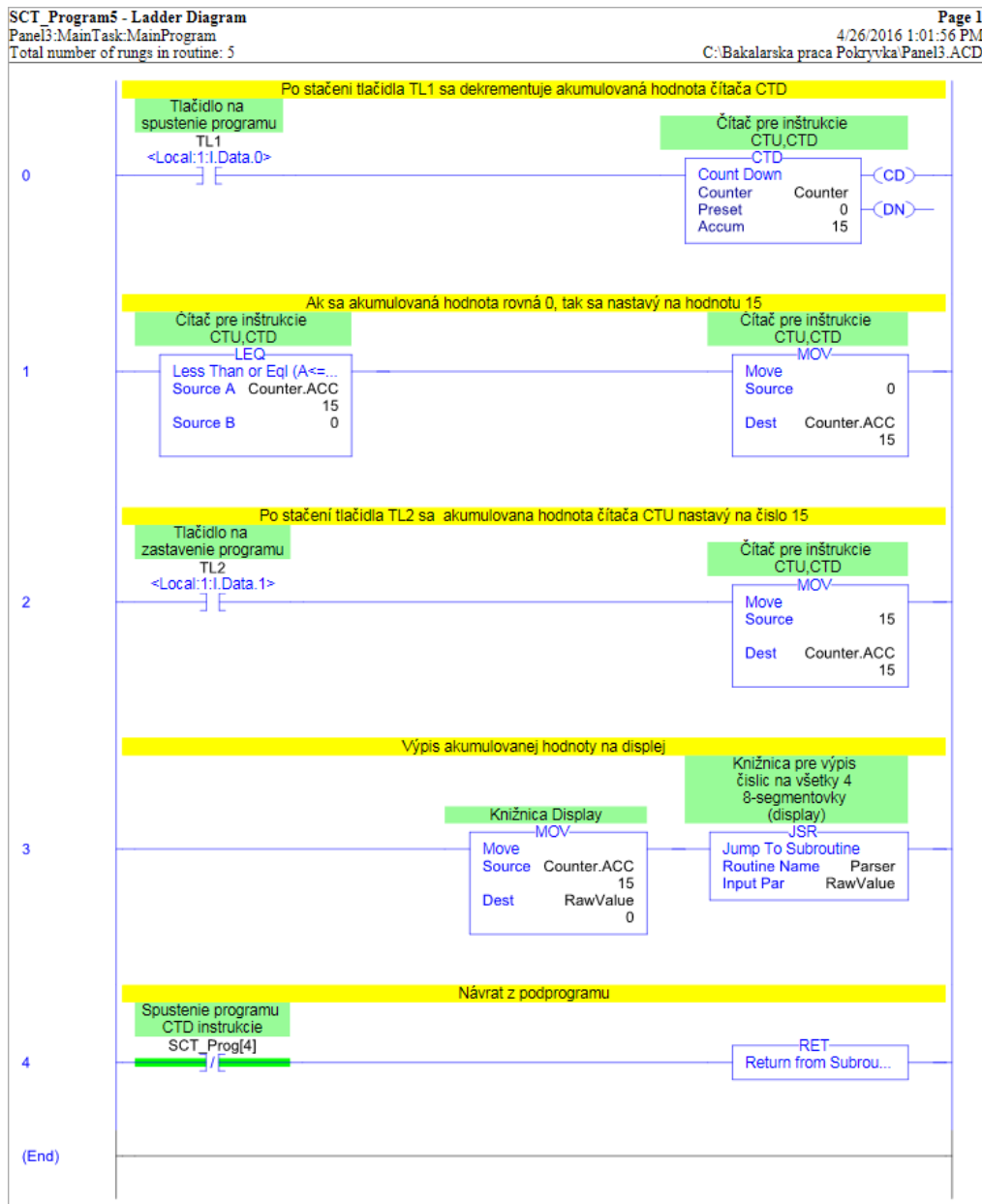
Je to počet zmien stavov rungu z nepriechodného na priechodný, na ktorom sa nachádza inštrukcia CTD.

SCT_Program5

Úloha: Úlohou programu je znázornenie funkcionality inštrukcie CTD, ktorá predstavuje softvérový, ktorý dekrementuje akumulovanú hodnotu až po 0 .

Riešenie: Úlohu znázorňujeme na príklade počítadla stlačení tlačidla TL1, kde sa zadaná hodnota dekrementuje od hodnoty 15 až po 0, ak hodnota klesne na nulu, tak už nižšie neklesne. Pri stlačení tlačidla TL2 sa zadaná hodnota nastaví na hodnotu 15. Do podprogramu SCT_Program5 vstúpime, ak stav premennej SCT_Prog[4] nadobúda logickú jednotku.

Rebríková schéma podprogramu SCT_Program5:



RES (Obnoviť)

Inštrukcia obnovuje hodnotu akumulátora a stavového bitu čítača a časovača, podľa zadanej adresy. Táto inštrukcia dokáže obnoviť stav CTU, CTD, TON, RST, HSC ale aj TOF ak má rovnakú adresu.