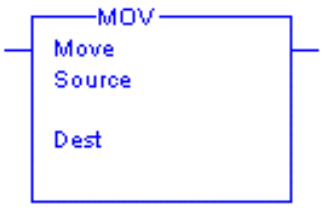
Posuvné inštrukcie

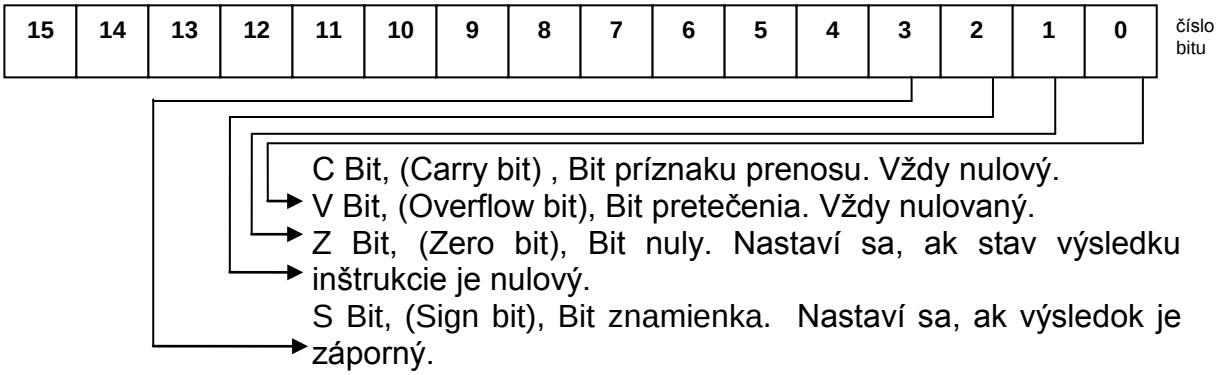


MOV (Posunúť do)

MOV je inštrukcia, ktorá kopíruje obsah Source slova do Dest slova. Obsah ostáva nezmenený. Táto inštrukcia sa vykoná, ak stav rungu je priechodný.

Aritmetické stavové bity nájdeme v slove 0 (S0) a od prvého až po tretí bit. Po vykonaní inštrukcie sa tieto bity v stavovom súbore aktualizujú.

Slovo0:



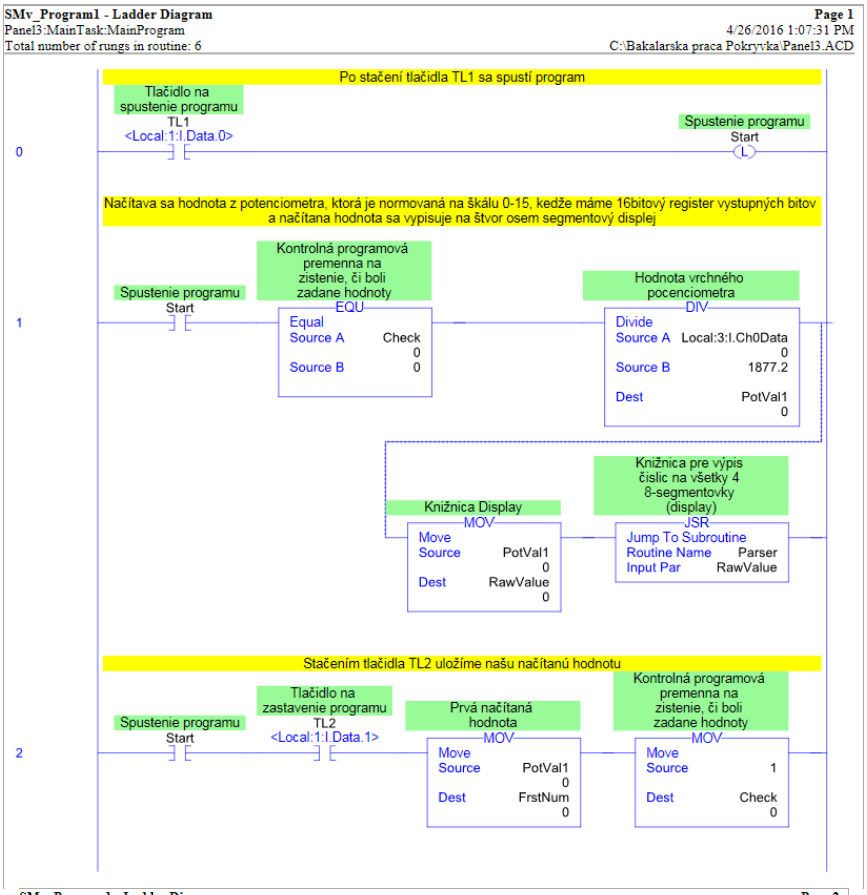
C Bit, (Carry bit) , Bit príznaku prenosu. Vždy nulový.
V Bit, (Overflow bit), Bit pretečenia. Vždy nulovaný.
Z Bit, (Zero bit), Bit nuly. Nastaví sa, ak stav výsledku inštrukcie je nulový.
S Bit, (Sign bit), Bit znamienka. Nastaví sa, ak výsledok je záporný.

SMv_Program1

Úloha: Úlohou programu je znázornenie funkcionality inštrukcie MOV, pričom sa bude načítavať hodnota z potenciometra, ktorá sa presunie na adresu výstupných bitov.

Riešenie: Program sa spúšťa tlačidlom TL1 a obnovuje sa tlačidlom TL4. Po spustení program sa načíta hodnotu z potenciometra a po stlačení tlačidla TL2 sa odošle hodnota v rozsahu od 0 po 15 do knižnice SetBit, kde sa prevedie na adresu 16bitového registra z nastaveným bitom a pomocou inštrukcie MOV sa odošle na adresu výstupných bitov. Do podprogramu SMv_Program1 vstúpime, ak stav premennej SMv_Prog[0] nadobúda logickú jednotku.

Rebríková schéma podprogramu SMv_Program1:

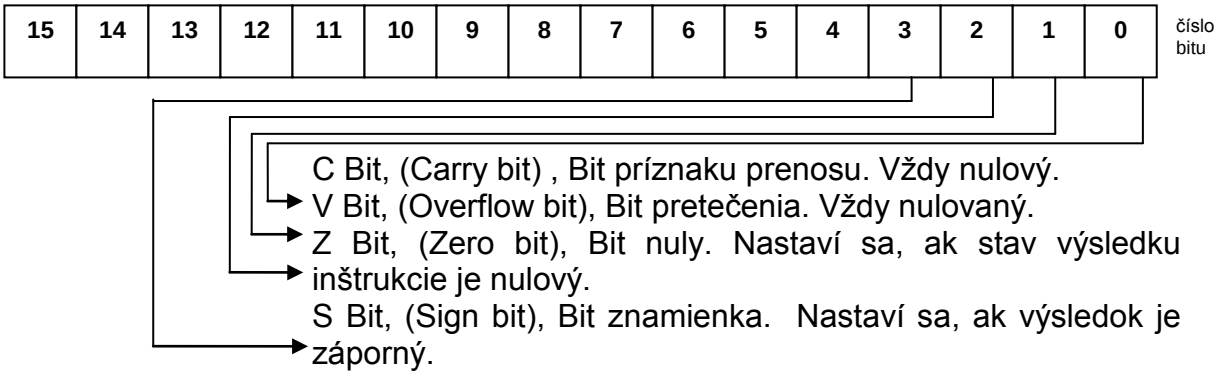


MVM (Posunúť cez masku do)

Táto inštrukcia kopíruje obsah Source slova cez masku do Dest slova. Obsah Source slova sa môže pri prechode cez masku zmeniť a uložiť do Dest slova Maskovanie je vysvetlené pri MEQ inštrukcii. Táto inštrukcia sa vykoná ak stav rungu je priechodný.

Aritmetické stavové bity nájdeme v slove 0 (S0) a od prvého až po tretí bit. Po vykonaní inštrukcie sa tieto bity v stavovom súbore aktualizujú.

Slovo0:



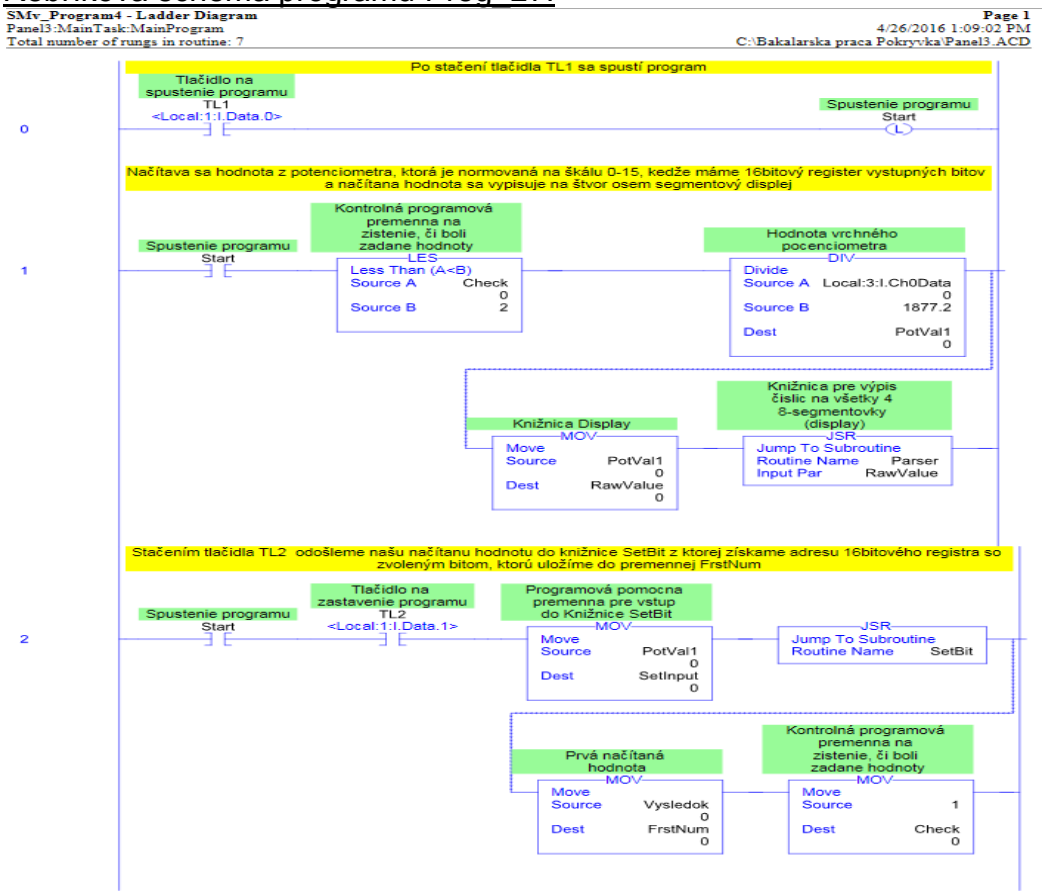
C Bit, (Carry bit) , Bit príznaku prenosu. Vždy nulový.
V Bit, (Overflow bit), Bit pretečenia. Vždy nulovaný.
Z Bit, (Zero bit), Bit nuly. Nastaví sa, ak stav výsledku inštrukcie je nulový.
S Bit, (Sign bit), Bit znamienka. Nastaví sa, ak výsledok je záporný.

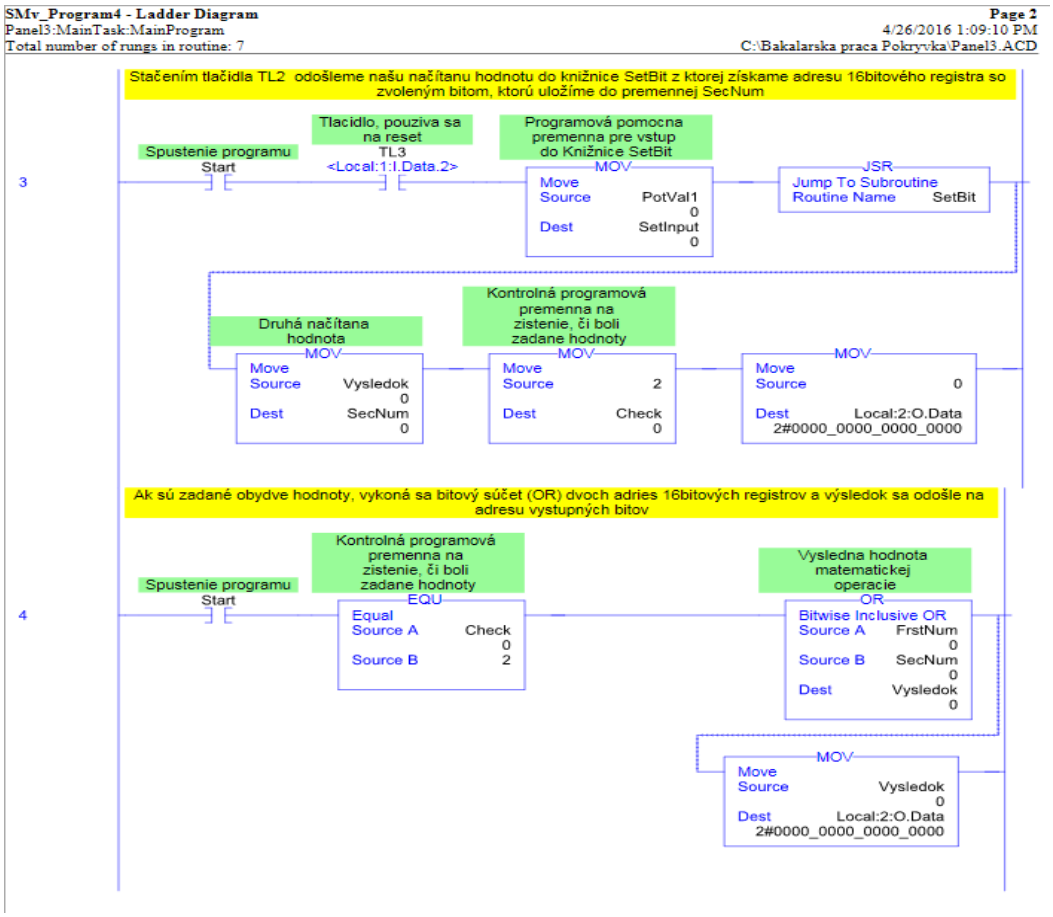
SMv_Program2

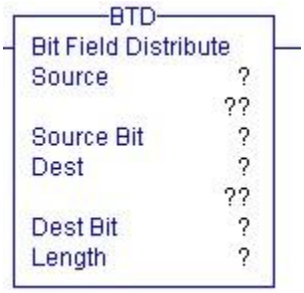
Úloha: Úlohou programu je znázornenie funkcionality inštrukcie MVM, pričom sa bude načítavať hodnota z potenciometra, ktorá sa presunie cez zadanu masku na adresu výstupných bitov.

Riešenie: Program sa spúšťa tlačidlom TL1 a obnovuje sa tlačidlom TL4. Po spustení program sa načíta hodnotu z potenciometra a po stlačení tlačidla TL2 sa odošle hodnota v rozsahu od 0 po 15 do knižnice SetBit, kde sa prevedie na adresu 16bitového registra z nastaveným bitom a pomocou inštrukcie MVM sa odošle cez Masku na adresu výstupných bitov. Do podprogramu SMv_Program2 vstúpime, ak stav premennej SMv_Prog[2] nadobúda logickú jednotku.

Rebríková schéma programu Prog_27:







BTD (Rozdelenie bitového poľa)

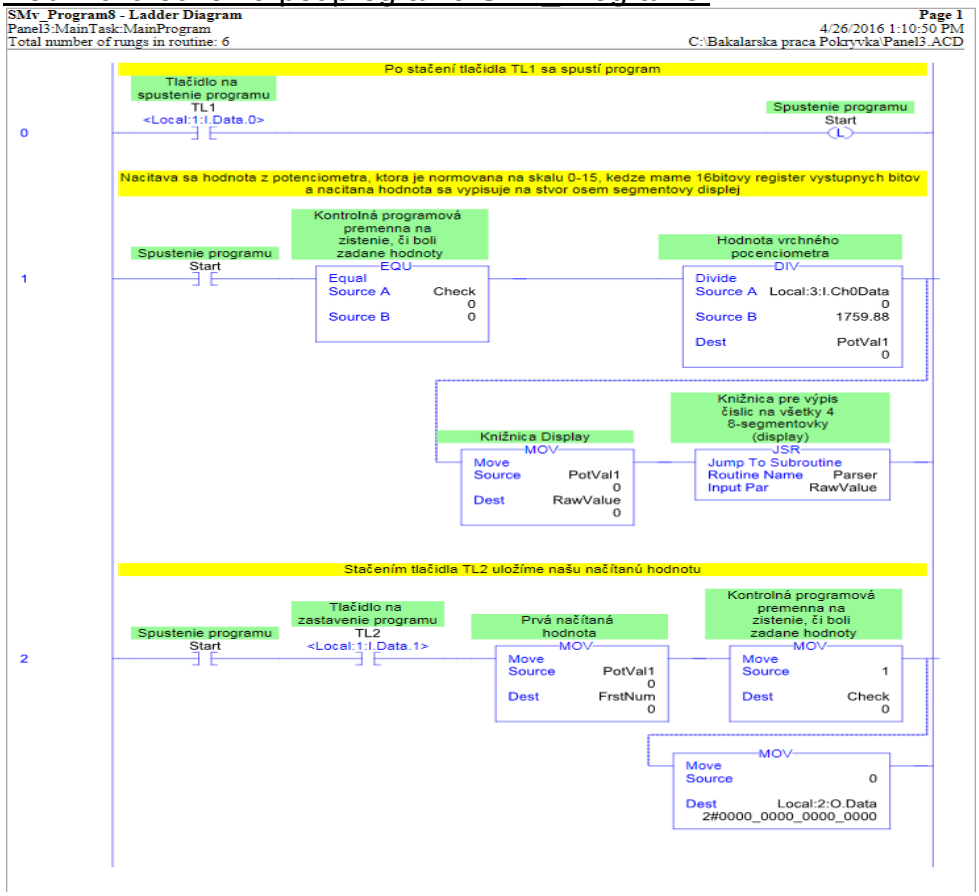
BTD je inštrukcia, ktorá sa vykoná, ak stav rungu je priechodný. Táto inštrukcia kopíruje skupiny bitov z registra uloženého v Source slove, do registra uloženého v Dest slove. Skupina bitov sa identifikuje pomocou registra Source Bit slova, kde je zadaný najnižší bit skupiny bitov. Kopírujú sa bity od najnižšieho zadaného až po zadaný počet, ktorý je uložený v Length slove. V slove Dest Bit je uložený najnižší bit od ktorého ma zapisovať bity do registra uloženého v Dest slove. Register uložený v Source slove ostáva nezmenený. Ak dĺžka kopírovaných bitov presiahne dĺžku registra Dest slova, tak ostatne bity sa neprekopírujú. Pri miešaní dátových typov, sa menší dátový typ doplní nulami aby sa dĺžkou dorovnal väčšiemu dátovému typu, napr. SINT a INT.

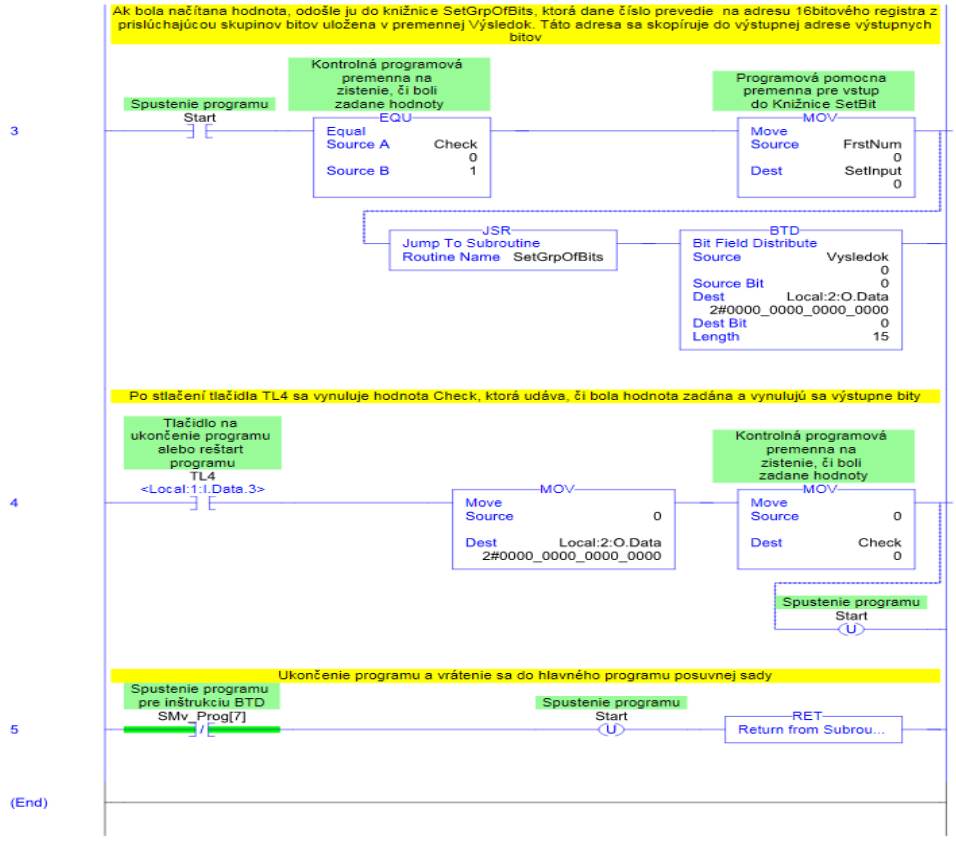
SMv_Program8

Úloha: Úlohou programu je znázornenie funkcionality inštrukcie BTD, ktorá prekopíruje počet bitov z zdrojovej adresy registra bitov do cielovej adresy registra bitov, pričom sa bude načítavať hodnota z potenciometra, ktorá sa prevedie na adresu registra bitov a vykoná sa prekopírovanie všetkých bitov získanej adresy do adresu výstupných bitov.

Riešenie: Program sa spúšťa tlačidlom TL1 a obnovuje sa tlačidlom TL4. Po spustení program sa načíta hodnotu z potenciometra a po stlačení tlačidla TL2 sa odošle hodnota v rozsahu od 0 po 15 do knižnice SetGrpOfBits, kde sa prevedie na adresu 16bitového registra z nastavenou skupinou bitov a pomocou inštrukcie BTD sa prekopírujú všetkých 15 bitov získanej adresy do adresu výstupných bitov. Do podprogramu SMv_Program8 vstúpime, ak stav premennej SMv_Prog[7] nadobúda logickú jednotku.

Rebríková schéma podprogramu SMv_Program8:




Ak bola načítana hodnota, odošle ju do knižnice SetGrpOfBits, ktorá dane číslo prevedie na adresu 16bitového registra z prislúchajúcou skupinov bitov uložena v premennej Výsledok. Táto adresa sa skopíruje do výstupnej adrese výstupnych bitov
Kontrolná programová premenna na zistenie, či boli zadane hodnoty
Programová pomocna premenna pre vstup do Knižnice SetBit
Spustenie programu
Start
3
EQU
Equal
Source A Check
0
Source B
1
MOV
Move
Source FrstNum
0
Dest SetInput
0
JSR
Jump To Subroutine
Routine Name SetGrpOfBits
BTD
Bit Field Distribute
Source Vysledok
0
Source Bit
0
Dest Local:2:O.Data
2#0000_0000_0000_0000
Dest Bit
0
Length
15
Po stlačení tlačidla TL4 sa vynuluje hodnota Check, ktorá udáva, či bola hodnota zadána a vynulujú sa výstupne bity
Tlačidlo na ukončenie programu alebo reštart programu
TL4
<Local:1:I.Data.3>
4
MOV
Move
Source
0
Dest Local:2:O.Data
2#0000_0000_0000_0000
Kontrolná programová premenna na zistenie, či boli zadane hodnoty
MOV
Move
Source
0
Dest Check
0
Spustenie programu
Start
U
Ukončenie programu a vrátenie sa do hlavného programu posuvnej sady
Spustenie programu pre inštrukciu BTD
SMv_Prog[7]
5
/
Spustenie programu
Start
U
RET
Return from Subrou...
(End)