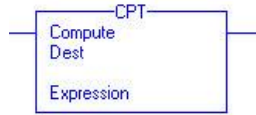


Matematické inštrukcie

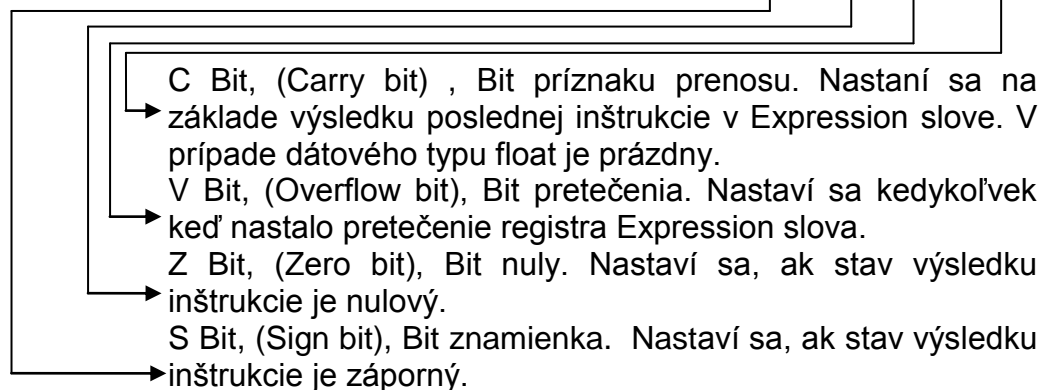
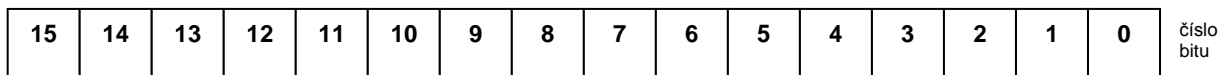
CPT (Vypočítat')



Inštrukcia sa vykoná ak stav rungu je priechodný. Táto inštrukcia slúži na výpočet zložitejších výpočtov, ktoré pomocou základných matematických inštrukcií boli náročne a prácne. Do adresy expression sa zadá matematická rovnica, ktorej výsledok sa po výpočte uloží do destination (dest) slova. Slovo môže nadobúdať adresu bitu alebo programovú adresu. Čas potrebný na vykonanie tejto inštrukcie je oveľa dlhší oproti bežným matematickým inštrukciám. Adresy môžu nadobúdať tieto dátové typy: SINT,INT,DINT,REAL.

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

Slovo0:



ADD (Súčet)

ADD	
Addr	
Source A	22406
	22406
Source B	N7:3
	0
Dest	N7:12
	0

Inštrukcia sa vykoná, ak stav rungu je priechodný. Sčítava Source A adresu a Source B adresu, výsledok ukladá do destination (dest) adresy, ktorú reprezentuje vnútorný register alebo výstupný bit. Obidva zdrojové (sources) adresy môžu byť adresy registrov s hodnotami alebo konštantné hodnoty, ale aspoň jeden zdroj musí byť adresa registra. Všetky adresy registrov môžu nadobúdať tieto dátové typy: SINT, INT, DINT, REAL, ale výpočet prebieha v DINT. Aritmetické stavové bity nájdeme v slove 0 na od prvého až po tretí bit. Po vykonaní inštrukcie sa tieto bity v stavovom súbore aktualizujú.

Slovo0:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	číslo bitu
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------------

- C Bit, (Carry bit), Bit príznaku prenosu. Nastaní sa, ak príznak prenosu bol generovaný. V prípade dátového typu float je prázdny.
- V Bit, (Overflow bit), Bit pretečenia. Nastaví sa kedykoľvek keď nastalo pretečenie registra Dest slova. Pretečená hodnota pri dátovom type float sa uloží do Dest slova.
- Z Bit, (Zero bit), Bit nuly. Nastaví sa, ak stav výsledku inštrukcie je nulový.
- S Bit, (Sign bit), Bit znamienka. Nastaví sa, ak stav výsledku inštrukcie je záporný.

SUB (Rozdiel)

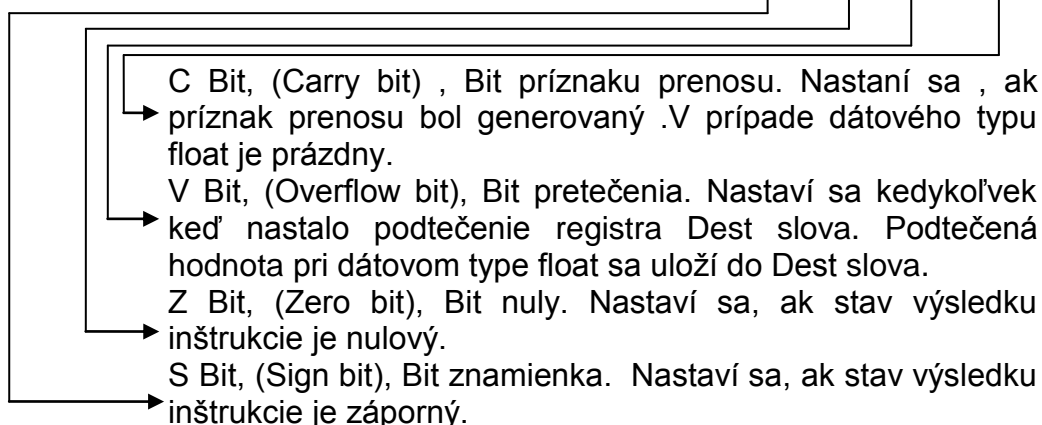
SUB	
Subtract	
Source A	50000.0 50000.0
Source B	R7:33 0
Dest	R7:34 0

Inštrukcia sa vykoná, ak stav rungu je priechodný. Odčíta hodnotu uloženú v adrese Source B od hodnoty uloženej v adrese Source A a výsledok uloží do destination adresy. V zdrojových adresách môžu byť adresy bitových registrov alebo konštanty, ale obidve zdrojové adresy nemôžu byť konštantné hodnoty. Všetky adresy registrov môžu nadobúdať tieto dátové typy: SINT, INT, DINT, REAL, ale výpočet prebieha v DINT.

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

Slovo0:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	číslo bitu
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------------



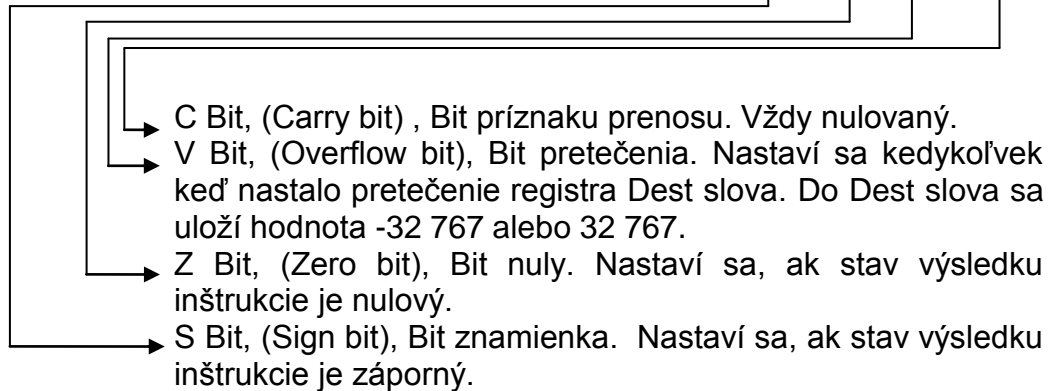
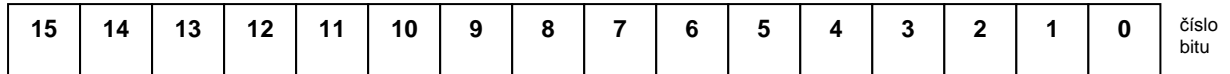
MUL (Súčin)

MUL	
Multiply	
Source A	500 500
Source B	R30:0 0
Dest	R30:10 0

Inštrukcia sa vykoná, ak stav rungu je priechodný. Inštrukcia MUL násobí hodnotu uloženú v zdrojovej (sources) A adrese s hodnotou uloženou v zdrojovej (source) B adrese a výsledok uloží do destination (dest) adresy. Zdroj A a zdroj B môžu byť konštantne hodnoty alebo adresy registrov s hodnotami. Adresy môžu nadobúdať tieto dátové typy: SINT, INT, DINT a REAL.

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

Slovo0:



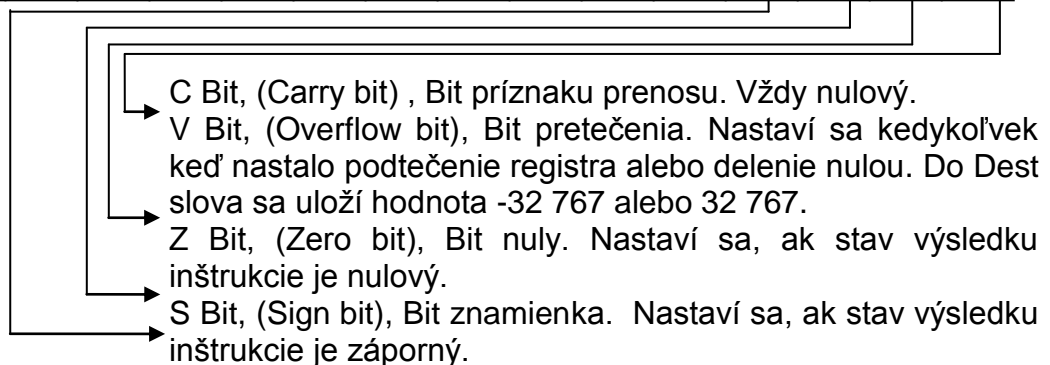
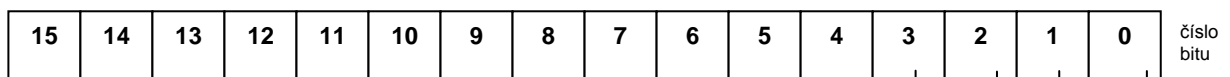
DIV (Podiel)

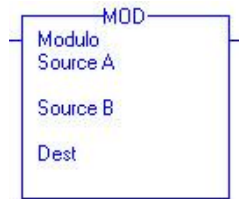
DIV	
Divide	
Source A	N7:20
	0
Source B	44
	44
Dest	N7:5
	0

Inštrukcia sa vykoná, ak stav rungu je priechodný. Div inštrukcia delí hodnotu uloženom v zdrojovej (sources) A adrese hodnotou uloženou v zdrojovej (sources) B adrese, a výsledok ukladá do cieľovej (dest) adresy, ktorá nadobúda jeden z týchto dátových typov: SINT, INT, DINT, REAL. Cieľová adresa by mala nadobúdať REAL dátový typ ináč dôjde ku zaokrúhľovaniu. Pozor pri delení nulovou hodnotou nastane minoritná chyba.

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

Slovo0:





MOD (Modulo)

Inštrukcia sa vykoná, ak stav rungu je priechodný. Táto inštrukcia delí, rozdeľuje hodnotu zdrojovej (sources) A adresy hodnotou zdrojovej (sources) B adresy pomocou vzťahu:

$$\text{DEST} = \text{SOURCES A} - (\text{TRN}(\text{SOURCES A}/\text{SOURCE B}) * \text{SOURCES B})$$

a ukladá výsledok do cieľovej (dest) adresy. Vo vzťahu TRN je zaokrúhlenie delenia na celočíselný výsledok. Adresy môžu mať tieto dátové typy: SINT, INT, DINT, REAL. Minoritná chyba môže nastať pri delení nulou.

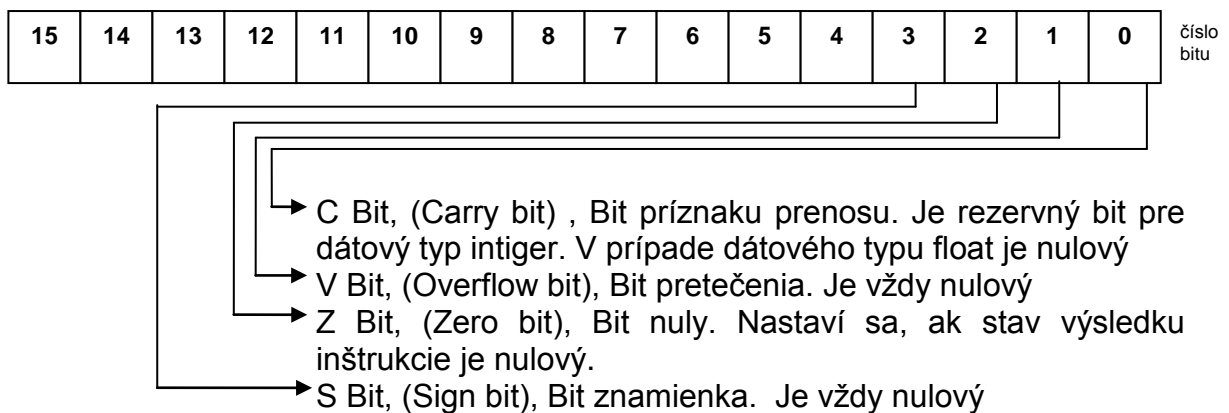


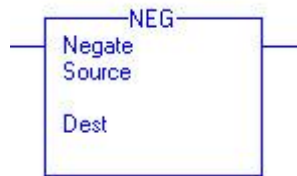
SQR (Odmocnina)

Inštrukcia sa vykoná, ak stav rungu je priechodný. SQR inštrukcia vykonáva druhú odmocninu hodnoty zdrojovej (sources) adresy a výsledok ukladá do cieľovej adresy. Vhodné dátové typy pre SQR inštrukciu sú DINT a REAL ale môžu sa použiť aj SINT a INT. Funkčný blok SQR inštrukcie nemá žiadne chybové podmienky.

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

Slovo0:





NEG (Negácia)

Inštrukcia sa vykoná, ak stav rungu je priechodný. Neg inštrukcia mení znamienko hodnoty zdrojovej (sources) adresy a výsledok uloží do cieľovej (dest) adresy. Vhodné dátové typy pre túto inštrukciu sú SINT, INT, DINT, REAL. Funkčný blok neg inštrukcie nemá žiadne chybové podmienky.

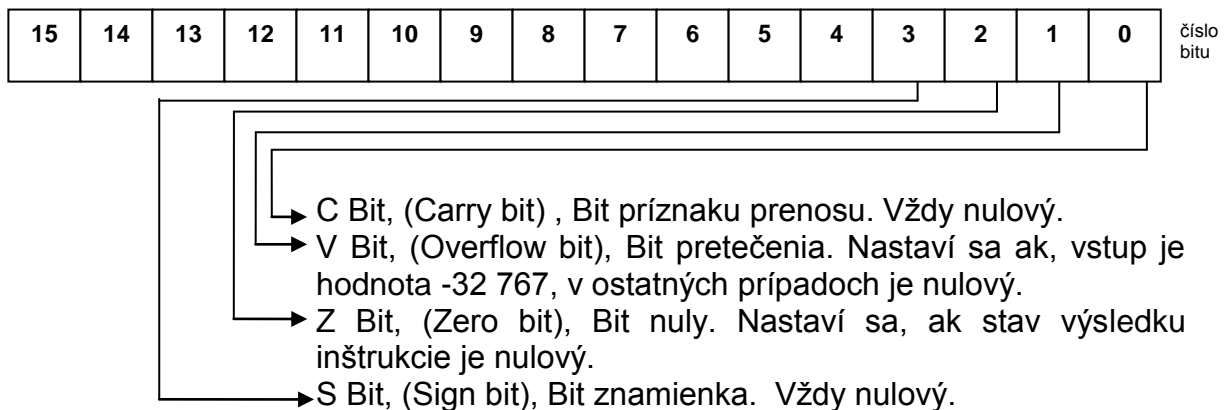


ABS (Absolútna hodnota)

Inštrukcia sa vykoná, ak stav rungu je priechodný. ABS inštrukcia ukladá do cieľovej adresy absolútnu hodnotu zdrojovej adresy. Vhodné dátové typy sú SINT, INT, DINT, REAL. Funkčný blok abs inštrukcie nemá žiadne chybové podmienky.

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:



Porovnávacie inštrukcie



CMP (Porovnaj)

Inštrukcia sa vykoná, ak stav rungu je priechodný. CMP inštrukcia porovnáva aritmetické adresy (tágy), alebo aritmetickú adresu so zadanou hodnotou, poprípade zadaným výpočtom. Ak do expression zadáme iba aritmetickú adresu a ak je daný tág rovný nule, tak stav rungu za inštrukcia sa nastaví na nepriechodný. Ak daný tág nie je rovný nule, tak stav rungu ostáva priechodný.

Poradie	Možné matematické a logické operácie
1	()
2	ABS, ACS, ASN, ATN, COS, DEG, FRD, LN, LOG, RAD, SIN, SQR, TAN, TOD, TRN
3	**
4	- (negate), NOT
5	*, /, MOD
6	<, <=, >, >=, =, <>
7	- (subtract), +
8	AND
9	XOR
10	OR

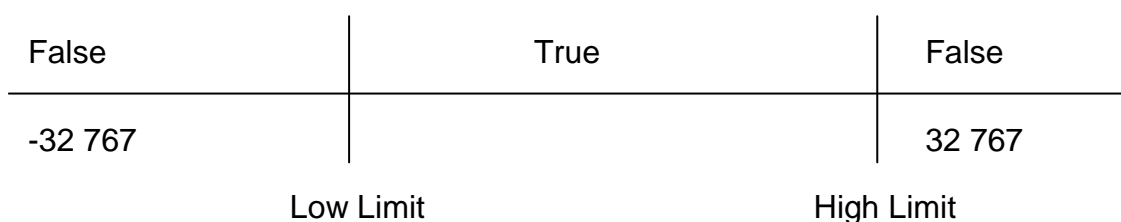
Tabuľka 1. Poradie vykonania jednotlivých operácií

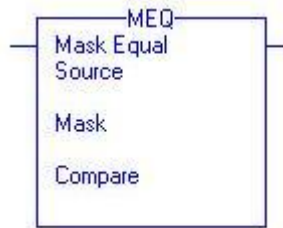


LIM (Intervalový test)

LIM je inštrukcia, ktorá nastaví stav rungu na priechodný, ak testovacia hodnota je väčšia ako low limit hodnota a menšia ako high limit hodnota. Test, low limit a high limit slova môžu byť adresy bitov alebo konštantné parametre. Ak test slovo nadobúda konštantný parameter, tak low limit slovo a high limit slovo musia nadobúdať adresy bitov. Ak test slovo nadobúda adresu bitu, tak low limit slovo a high limit slovo môžu byť konštantné parametre alebo adresy bitov.

Grafické zobrazenie True/False stavu inštrukcie LIM:





MEQ (Porovnanie cez masku)

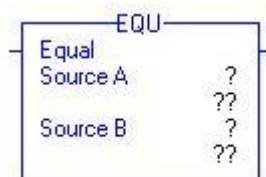
MEQ je inštrukcia, ktorá porovná hodnoty v source a compare slovách cez masku. Ak sa source hodnota, ktorá prešla cez masku rovná compare hodnote, ktorá tiež prešla cez masku, tak MEQ inštrukcia nastaví stav rungu na priechodný. Ak sa tieto hodnoty nerovnajú, tak MEQ inštrukcia nastaví stav rungu na nepriechodný. Hodnoty môžu byť zapísane v desiatkovej, dvojkovej a šestnástkovej sústave, zvičajne sa udávajú hodnoty v rovnakej číselnej sústave. Ak hodnota masky sa rovná 1, tak sa porovnajú iba hodnoty v source a compare slovách. Ak hodnota masky je rovná 0, tak MEQ inštrukcia nastaví nepriechodný stav rungu.

Maskovanie $S \text{ AND } M = Ms$

Source:	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0
Mask:	1	1	0	0	1	0	1	1	0	0	1	1	1	0	1	0
Masked source:	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0

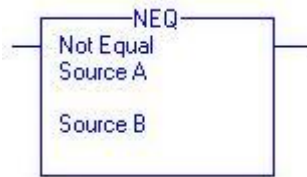
S	M	SM
0	0	0
1	0	0
0	1	0
1	1	1

Maskovacia
tabuľka



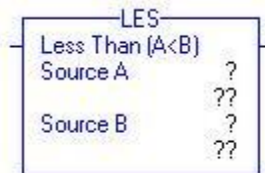
EQU (Rovnajú sa)

EQU je inštrukcia, ktorá overuje, či sa hodnota source A slova rovná hodnote source B slova. Ak sa hodnoty rovnajú, tak inštrukcia nastaví stav rungu na priechodný. Ak sa hodnoty nerovnajú, tak inštrukcia nastaví stav rungu na nepriechodný. Source A a source B slova môžu nadobúdať konštantne hodnoty alebo adresy bitov, ale aspoň jedno musí obsahovať adresu bitu.



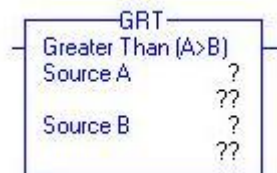
NEQ (Nerovnajú sa)

NEQ je inštrukcia, ktorá overuje, či sa hodnota source A slova nerovná hodnote source B slova. Ak sa tieto hodnoty rovnajú, tak inštrukcia NEQ nastaví stav rungu na nepriechodný. Ak sa tieto hodnoty nerovnejú, tak inštrukcia NEQ nastaví stav rungu na priechodný. Source A a source B slova môžu nadobúdať konštantnú hodnotu alebo adresu bitu, ale aspoň jedno slovo musí obsahovať adresu bitu.



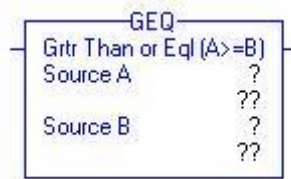
LES (Menší ako)

LES je inštrukcia, ktorá overuje, či hodnota source A slova je menšia ako hodnota source B slova a ak je menšia, tak inštrukcia nastaví stav rungu na priechodný. Ak hodnota source A slova nie je menšia ako hodnota source B slova, tak inštrukcia nastaví stav rungu na nepriechodný. Source A a source B slova môžu byť konštantné hodnoty alebo adresy bitov, ale aspoň jedno slovo musí obsahovať adresu bitu.



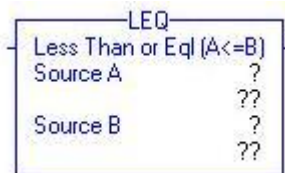
GRT (Väčší ako)

GRT je inštrukcia, ktorá overuje či hodnota source A slova je väčšia ako hodnota source B slova a ak je väčšia, tak inštrukcia nastaví stav rungu na priechodný. Ak hodnota source A slova je menšia ako hodnota source B slova, tak inštrukcia nastaví stav rungu na nepriechodný. Source A a source B slova môžu byť konštantné hodnoty alebo adresy bitov, ale aspoň jedno slovo musí obsahovať adresu bitu.



GEQ (Väčší alebo rovný ako)

GEQ je inštrukcia, ktorá overuje či hodnota source A slova je väčšia alebo rovná ako hodnota source B slova a ak je väčšia alebo rovná, tak inštrukcia nastaví stav rungu na priechodný. Ak hodnota source A slova je menšia ako hodnota source B slova, tak inštrukcia nastaví stav rungu na nepriechodný. Source A a source B slová môžu byť konštantné hodnoty alebo adresy bitov, ale aspoň jedno slovo musí obsahovať adresu bitu.



LEQ (Menší alebo rovný ako)

LEQ je inštrukcia, ktorá overuje či hodnota source A slova je menšia alebo rovná ako hodnota source B slova a ak je menšia alebo rovná, tak inštrukcia nastaví stav rungu na priechodný. Ak hodnota source A slova je väčšia ako hodnota source B slova, tak inštrukcia nastaví stav rungu na nepriechodný. Source A a source B slová môžu byť konštantné hodnoty alebo adresy bitov, ale aspoň jedno slovo musí obsahovať adresu bitu.

Logické inštrukcie



AND (Bitový súčin)

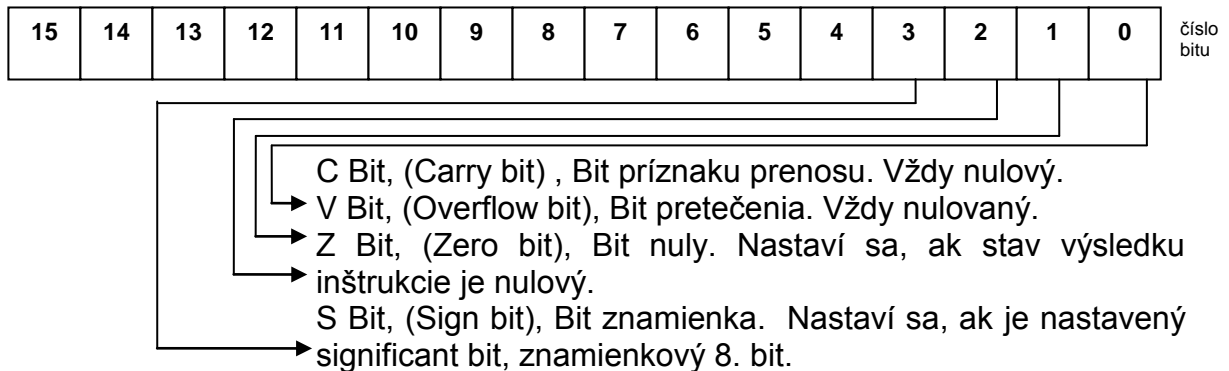
AND je inštrukcia, ktorá predstavuje bitový súčin Source A slova a Source B slova. Source A a Source B môžu byť adresy bitov, alebo jedno slovo môže byť konštanta. Obidve slova nemôžu byť konštanty. Ak bit v Source A aj B slove nie sú nastavené logickú nulu, tak hodnota v Dest slove sa vypočíta násobením Source A slova s Source B slovom, v prípade že stav rangu je priechodný. Ostatne možnosti nastavujú hodnotu v Dest slove na logickú nulu alebo, ak stav rangu je nepriechodný.

Source A	Source B	Dest
0	0	0
1	0	0
0	1	0
1	1	1

Tabuľka 2. Bitový súčin

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.

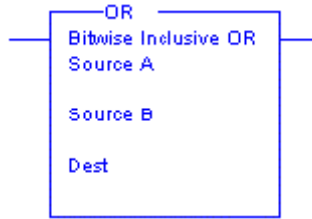
Slovo0:



Názorná ukážka bitovej operácie AND:

```

Source A:  0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
Source B:  1 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0
Dest:      0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
  
```



OR (Bitový súčet)

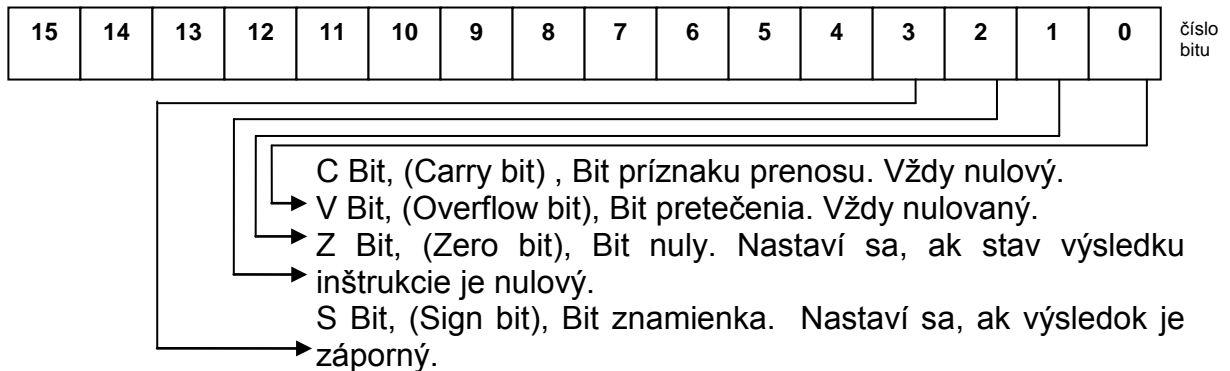
OR je inštrukcia, ktorá predstavuje bitový súčet Source A slova a Source B slova, kde výsledok sa uloží do Dest slova, ak stav rungu je priechodný. Ak stav rungu bude nepriechodný, tak inštrukcia sa nevykoná. Source A a Source B slova môžu byť adresy bitov alebo konštanty, ale aspoň jedno slovo musí byť adresa bitu. Ak Source A slovo a Source B slovo sú nastavené na logickú nulu, tak aj výsledok v Dest slove bude logická nula.

Source A	Source B	Dest
0	0	0
1	0	1
0	1	1
1	1	1

Tabuľka 3. Bitový súčet

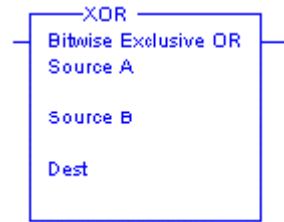
Aritmetické stavové bity nájdeme v slove 0 (S0) a od prvého až po tretí bit. Po vykonaní inštrukcie sa tieto bity v stavovom súbore aktualizujú.

Slovo0:



Názorná ukážka bitovej operácie OR:

Source A:	0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
Source B:	1 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0
Dest:	1 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0



XOR (Exkluzívny bitový logický súčet)

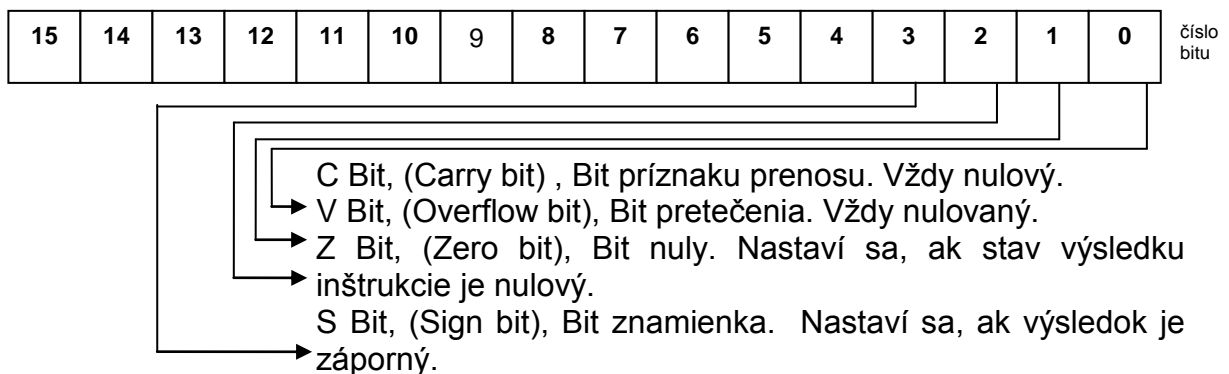
Táto inštrukcia vykonáva exkluzívny logický súčet Source A slova s Source B slovom a výsledok uloží do Dest slova, ak stav rungu je priechodný. V

Source A	Source B	Dest
0	0	0
1	0	1
0	1	1
1	1	0

Tabuľka 4. Exkluzívny bitový súčet

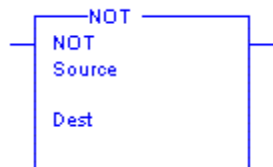
Aritmetické stavové bity nájdeme v slove 0 (S0) a od prvého až po tretí bit. Po vykonaní inštrukcie sa tieto bity v stavovom súbore aktualizujú.

Slovo0:



Názorná ukážka bitovej operácie XOR:

Source A:	0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0	0
Source B:	1 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0	0
Dest:	1 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0	



NOT (negácia bitov)

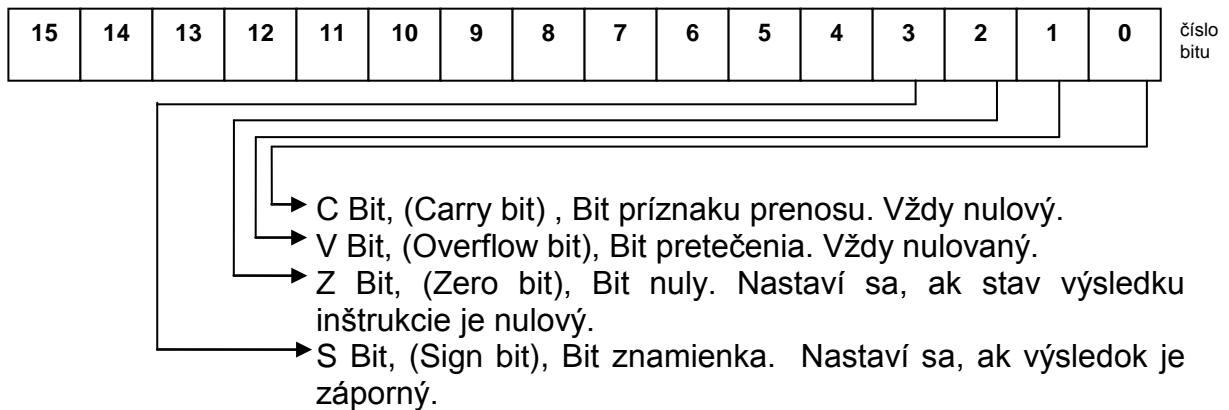
NOT je inštrukcia, ktorá vykonáva bitovú negáciu registra uloženého v Source slove a ukladá výsledok do Dest slova. Táto inštrukcia sa vykoná, ak stav rungu je priechodný. Pri miešaní dátových typov, sa menší dátový typ doplní nulami aby sa dĺžkou dorovnal väčšiemu dátovému typu, napr. SINT a INT.

Source	Dest
0	1
1	0

Tabuľka 4. Bitová negácia

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.

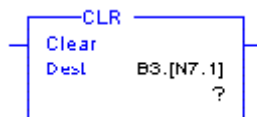
Slovo0:



Názorná ukážka bitovej operácie NOT:

Source A: 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
 Dest: 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1

CLR (Nulovať)



CLR je inštrukcia, ktorá vynuluje všetky bity v registri, ktorý je uložený v Dest slove. 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:

