**Skoky**

**podmíněné**

V této kapitole se naučíme podmíněné skoky. Očekáváme znalost z kapitoly „Nepodmíněné skoky“

Podmíněný skok je takový, který se někdy provede a jindy neprovede, v závislosti na nějaké podmínce.

První takový skok je

**btsc registr , #bit**

nebo

**btsc bunka , #bit**

např: btsc W5, #4 btsc 0x1000, #15

Anglicky: bit test and skip if clear

Instrukce otestuje zadaný bit, a když je tento bit v 0 , přeskočí **jednu** následující instrukci. Důležité je, že přeskočí jednu instrukci. A buď přeskočí, nebo nepřeskočí.

Pustíme si v simulátoru file podmin01.s . Ten je tentokrát pojat jako hra. Musíte nastavit číslo v registru W5 tak, abyste se dostali až na konec programu.

**Obsah registru změníte v simulátoru jednoduše. Ve Variables cvakneme na řádek, ve kterém máme WREG5 , cvakneme do sloupce value a hodnotu přepíšeme.**

Druhý skok je

**btss registr , #bit**

nebo

**btss bunka , #bit**

např: btss W5, #4 btss 0x1000, #15

Anglicky: bit test and skip if set

Instrukce otestuje zadaný bit, a když je tento bit v 1 , přeskočí **jednu** následující instrukci. Důležité je, že přeskočí právě jednu instrukci. A buď přeskočí, nebo nepřeskočí.

Dalším typem podmíněných skoků jsou skoky, které skáčou podle stavu bitů status registru (SR , viz aritmetické operace)

bra podmínka , okolik

bra Z , kam pokud je bit Z status registru v 1 , skočí na návěští kam

bra NZ , kam pokud je bit Z status registru v 0 , skočí na návěští kam

bra C , kam pokud je bit C status registru v 1 , skočí na návěští kam

bra NC , kam pokud je bit C status registru v 0 , skočí na návěští kam

bra N , kam pokud je bit N status registru v 1 , skočí na návěští kam

bra NN , kam pokud je bit N status registru v 0 , skočí na návěští kam

Logika je jednoduchá: pokud je v podmínce název bitu, skočí, když je příslušný bit v 1. Pokud je u názvu bitu N, skočí, když bit je v 0 ( N - not ).

Skoky bra očekávají, že mají nastavené bity Status registru SR. To dělají aritmetické instrukce a mnohé další. Pro porovnání dvou čísel je určena instrukce cp

**cp neco , necojineho**

Instrukce cp provede rozdíl ( neco – necojineho) , **výsledek nikam neuloží**, a podle výsledku nastaví bity SR. Tím je dáno, že oba operandy instrukce cp zůstanou nezměněny. Z nastavení bitů SR lze pak usoudit, v jakém vzájemném stavu byly oba operandy

Na straně 330 datasheetu máte ještě mnoho dalších podmínek pro BRA. Vždy se jedná o skok podle nějaké kombinace bitů SR, a předpokládá se, že před bra bylo provedeno cp.

Dále si pustíme file podmin02.s . V něm máme postupně dosazovat do registrů v simulátoru (VARIABLES) takové hodnoty, abyste se v programu dostali až na jeho konec. **A především, ujasnit si, co dělají podmíněné skoky !**

Na následujícím obrázku máte příklad realizace konstrukce if ( ) else ,

zde konkrétně if ( 3.bitW0 == 1 ) KUS1 ; else KUS2 ;

|  |  |
| --- | --- |
| **3. bit W0 má hodnotu 0** | **3. bit W0 má hodnotu 1** |
| skip if set – není nastaven, proto pokračujeme následující instrukcí | skip if set – je nastaven, proto přeskočíme jednu následující instrukci |
|  |  |