**Skoky**

**nepodmíněné**

V této kapitole se naučíme skoky, tedy jak donutit program, aby pokračoval někde jinde, ne na instrukci, která následuje

Procesor při vykonávání programu pracuje následovně:

1. přečte instrukci z paměti programu EEPROM. Adresa instrukce je uložena v registru PC
2. instrukci dekóduje
3. instrukci provede
4. zvýší obsah registru PC
5. pokračuje bodem 1

Registr, který určuje, ze které adresy se má instrukce číst, je PC. Po RESETu procesoru je samozřejmě PC == 0 , takže čteme instrukci z adresy 0x0000 .

**Změna obsahu registru PC znamená skok v programu** .

Pokud tedy chceme procesor donutit, aby nepokračoval na následující instrukci, ale někde jinde, prostě změníme obsah registru PC

**Nepodmíněný absolutní skok**

nepodmíněný – provede se vždycky

absolutní – je přímo udána adresa, na kterou má program skočit

**goto adresa**

Takže například

goto 0x1234 provede skok na adresu 0x1234

zapsat adresu v instrukci je možné, bude to fungovat. Ale pro programátora je to výrazně nepohodlné a nepřehledné, protože si on sám musí spočítat, na jakou adresu chce skočit. Naštěstí nám přichází na pomoc překladač s mechanismem, který se jmenuje návěští. V simulátoru si pustíme program nepod.s

**Návěští znamená adresu instrukce, u které stojí** ( u které stojí to návěští)

Překladač pracuje tak, že kdykoli narazí v programu na navesti , nahradí ho adresou

Dále si pusťte Window – pic Memory views – Program memory , okno si zaktivujte cvaknutím myši a mačkejte F7

Ve sloupci opcode vidíme instrukční kód. Pro goto je poměrně jasný – 04 , a za ním je adresa, kam má program skočit. Projděte si skoky několikrát, až pochopíte, co se přesně děje

**Nepodmíněný relativní skok**

nepodmíněný – provede se vždycky

relativní – je udáno, o kolik buněk se má skočit

**bra okolik**

Opět by pro programátora bylo smrtící, kdyby měl počítat, o kolik buněk má skočit. Napíše prostě návěští a překladač sám spočítá, o kolik buněk má procesor skočit. Vidíme to na instrukcích

bra aa01

bra aa02

které jsou obě kódovány jako 0x37003 , ačkoli jsou v jiném místě programu.

u instrukce bra aa04 vidíme skok dozadu , je kódována jako 0x37fff9 , takže „o kolik“ je 0xfff9 , což je krásné záporné číslo v druhém doplňku.

V zásadě můžeme používat jak goto, tak bra ,

jediný rozdíl je, že goto „doskočí“ dál – v intervalu ∈ [0 ... 8388606].,

zatímco bra skáče o ∈ [-32768 ... +32767]. .