

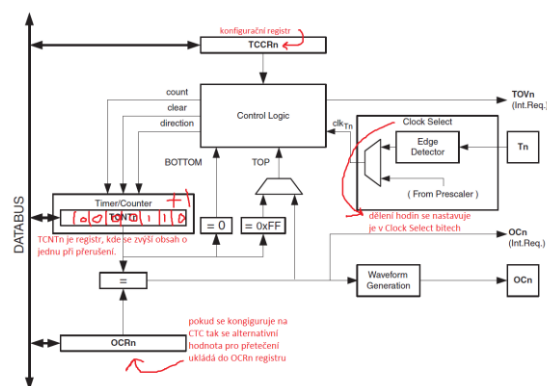
## ÚVAHA A ZADÁNÍ PRO LABORATORNÍ CVIČENÍ ČÍTAČE

Čítač je periferie integrovaná na palubě procesoru, která slouží k měření času. Časem je myšlen počet period taktovacího signálu po průchodu elektronickým obvodem pro dělení hodin.

Tato periferie má několik možností konfigurace, mezi které patří také počítací (zvyšuje o jednu) a odpočítávací (snižuje o jednu). Jak je zřejmé, jedná se o elektronický obvod skládající se z registrů a kombinační logiky.

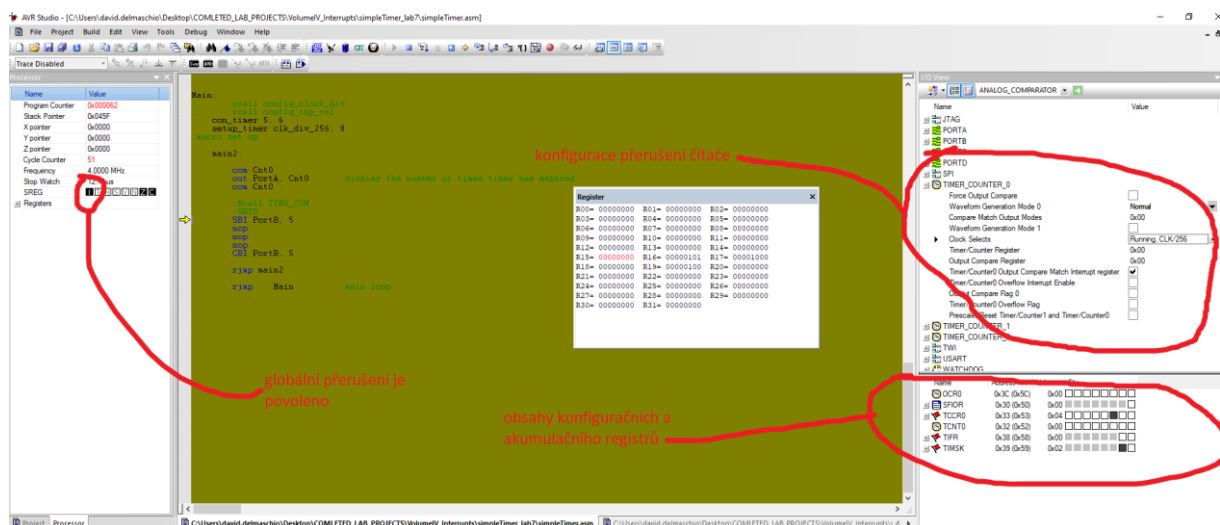
Jedním z jeho úkolů je číselným údajem uloženým do akumulárního registru zaznamenat počet událostí (period) nebo procesů a informovat, kolikrát proběhly. Je vytvořen tak, aby dokázal rychle čítat elektrické impulsy, jejichž frekvence je zaznamenána v kHz, MHz a GHz.

Graficky na ATmega16 tento obvod vypadá zhruba takto:



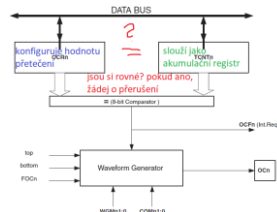
Jak je z obrázku patrné, tento obvod obsahuje také konfigurační registry, jako například TCCRn a OCRn. Jsou tam i jiné, se kterými je potřeba se seznámit. Klidně se zeptejte při hodině, nebo na Teamsech.

V podstatě nejjednodušší způsob jak se s touto periferií seznámit je pomocí simulátoru. Pokud si načtete zdroj pro čítač z teams, a spustíte simulaci, uvidíte něco takového:

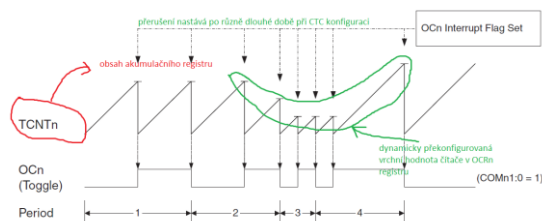


Do konfiguračních registrů se ukládají konfigurační hodnoty, kterými se nastaví požadavky na funkci čítače. V podstatě se jedná o dvě zásadní možnosti jak periférii čítače zkonfigurovat. Je to overflow a compare match.

Přetečení po dosažení maximální hodnoty čítače je situace, kdy například 8-bitový čítač dosáhne hodnoty 255 a pak přeteče. Obvykle se tomuto stylu konfigurace říká overflow, nebo-li OVF konfigurace.



Compare match je způsob konfigurace, kde místo maximální možné hodnoty čítače programátor zadá libovolnou hodnotu menší nebo rovnou 255. Často se tomuto stylu konfigurace říká CTC konfigurace. Zde je grafické znázornění obsahu akumulčního registru při konfiguraci na CTC.



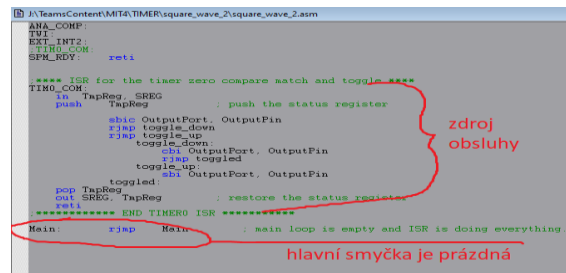
Přetečení čítače znamená, že periférie čítače vyšle jádru žádost o přerušení.

#### Průběh přerušení (všeobecně):

- 1) Přijde žádost o přerušení (na jádro od periférie)
- 2) Uloží návratovou adresu (kde byl přerušen)
- 3) Skočí do tabulky vektorů přerušení do adresy, která odpovídá typu žádosti. V tomto místě je „jmp jméno-obsluhy“ -> TIM0\_CON  
*Vše do teď bylo čistě HW, od 3 a dál je zas FW*
- 4) Skočí do podprogramu obsluhy
- 5) Reti „od popne“ návratovou adresu a nahodí I. Bit v SREG
- 6) Skočí do místa, kde byl přerušen

Jak na to a co dál? Je potřeba překonfigurovávat čítač. Je to možné udělat tak, že se přizpůsobí obsluha čítače tak aby se na výstupním pinu vygenerovala čtvercová vlna různých frekvencí. Tyto frekvence pak změříme oscilopem a porovnáme je s vypočtenými hodnotami.

Překonfigurování je možné také udělat dynamicky bez přeprogramování procesoru. To je pomocí tlačítka, pak už je možné se bavit o generátoru čtvercových průběhů. Nechám to na vás jak se rozhodnete. Můžete si přepisovat [.hex](#) je mi to jedno.



```

D:\TeamContent\MITA\TIMER\square_wave_2\square_wave_2.asm
;*** ISR for the timer zero compare match and toggle ***
TIMO_COM:
    in     TapReg, SREG          ; push the status register
    push  TapReg
    sbic  OutputPort, OutputPin
    rjmp  toggle_down
    rjmp  toggle_up
    toggle_down:
        sbi  OutputPort, OutputPin
        rjmp  toggled
    toggle_up:
        sbi  OutputPort, OutputPin
        rjmp  toggled
    toggled:
        pop  TapReg
        out  SREG, TapReg        ; restore the status register
        reti
;***** END TIMER0 ISR *****
Main:    rjmp  Main             ; main loop is empty and ISR is doing everything

```

Ze snímku simulátoru je vidět, že obsluha při každém průchodu přepne hodnotu na výstupním pinu na hodnotu opačnou, to znamená při každém přetečení z logické 0 na logickou 1, a z logické 1 na logickou 0. Běžně se tomuto ději říká toggle.

## ZADÁNÍ

Dokonči zdroj pro timer v assembleru, zjisti přesnost čítače na ATmega16, porovnej naměřenou hodnotu s vypočítanou, sestroj grafy a udělej závěr.

Zde již pokračuje bývalý žák v jeho bádání a postupech...

## Použité přístroje:

Osciloskop – elektronický měřicí přístroj s obrazovkou, který zaznamenává průběh měřeného napětového signálu. Existuje několik typů osciloskopů, například analogové (klasické, paměťové, vzorkovací) nebo digitální (spolupracují s osobním počítačem).

V našem případě byl použit digitální, tedy čtvercovou vlnu jsme mohli vidět na PC softwarově.

Osciloskop - parametry– max input: 30Vpp, mx output: 10Vpp ,2x 12MHz PC-Scope -> používaný vstup 2

Programátor – pro spojení desky s PC => (AVR-ISP)

Deska – Krystal typu: HC49/S QM- 14,7456 MHz, používaný typ portu -> D

Počítač – Použité programy: AVR\_ISP\_prog, PcLab2000LT, AVR studio 4

## Postup:

Napsal jsem si program v AVR, který má za funkci čítat při každém přejetí mainu dokud hodnota v (TIMER\_COUNTER\_0 -> TCNT0) nedojede do hodnoty overflow (TIMER\_COUNTER\_0 -> OCR0) => smyčka. Následně při dojetí do této hodnoty se nahodí na portu v mém případě zvoleném D I. Bit. – přesnější průběh přerušeni v teorii.

Po testování a ověření, že to dělá to, co má jsem si zapojil osciloskop k PC spolu programátorem a deskou, na desku jsem nahrál hex mého programu přes AVR\_ISP\_program a na osciloskopu (PcLab2000LT) jsem mohl sledovat čtvercovou vlnu spolu s potřebnou frekvencí, kterou jsem si zapsal .Poté jsem si vypočítal frekvenci, která by měla vyjít a rozdíl mezi naměřenou a reálnou frekvencí v procentech.

Pro výpočet frekvence můžete použít následující vzorec:

$$f_{\text{vyp}} = \frac{\text{Hodnota krystalu}}{2 * \text{dělení hodin} * \text{top hodnota}}$$

dělení hodin - (bez ,8,64,256,1024)

top hodnota – hodnota pro overflow

hodnota krystalu – pro nás tedy stálá konstanta vzhledem k tomu, že jsme jiný krystal nepoužívali (14,7456 MHz)

Příklad výpočtu pro *top hodnotu* 96 při *dělení* 8, označen tučně v tabule:

$$f_{\text{vyp}} = \frac{14,7456 * 10^6}{2 * 8 * 96 * 10^3} = \underline{\underline{9,6\text{kHz}}}$$

Výpočet chyby:

$$d_x = 100 - \left( \frac{9,22}{9,6} * 100 \right) = \underline{\underline{3,96\%}}$$

Tabulky:

dělení x-krystalu	top hodnota	f (KHz) - přečtena	f (KHz) - vypočítaná	Chyba (%)
-------------------	-------------	--------------------	----------------------	-----------

Obrázky

