

Grupa: B	Temat: Instrukcje sterujące
---------------------------	---------------------------------------

Data: 31.03.2021	Wykonał: Piotr Drabik
----------------------------	---------------------------------

Godzina: 12:30	II rok Informatyka Stosowana	Ocena i uwagi prowadzącego:	Prowadzący: dr hab. Witold Kozłowski
--------------------------	---	--	---

Opis zadania

Celem zadania jest zbadanie zależności czasowych instrukcji ustawiających stan portów na mikrokontrolerze. Określimy również ilość cykli pracy mikrokontrolera. Wartości odczytywać będziemy z symulatora zawartego w programie BASCOM-AVR oraz z oscyloskopu. Podczas zadania sprawdzimy wartości czasowe równoważnego programu napisanego w języku ASSEMBLER. W tym wypadku wartości odczytamy jedynie z oscyloskopu. Porty w mikrokontrolerze służą do jego komunikacji z otoczeniem i co ważne, porty te są dwukierunkowe, więc mogą być albo wejściami albo wyjściami.

Instrukcje Set/Reset

Aby sprawdzić działanie instrukcji **Set** i **Reset** musimy przygotować kod:

<code>\$regfile = "m8def.dat"</code>	Informuje kompilator o pliku dyrektyw mikroprocesora
<code>\$crystal = 8000000</code>	Informuje kompilator o częstotliwości oscylatora taktującego mikrokontrolera
<code>Config PINB.0 = Output</code>	Ustawiamy Pin B 0 jako wyjście
<code>Do</code>	Początek pętli
<code> Set Portb.0</code>	Ustawiamy wyjście PB0 na „1”
<code> Reset PORTB.0</code>	Ustawiamy wyjście PB0 na „0”
<code>Loop</code>	Koniec pętli głównej programu
<code>End</code>	Koniec programu

Zależności czasowe instrukcji Set/Reset odczytane z symulatora programu BASCOM-AVR

Instrukcje	Czas [ms]	Cykle
Set	0.77025	6162
Reset	0.7705	6164
Loop	0.77075	6166
Set	0.771	6168
Reset	0.77125	6170
Loop	0.7715	6172

Jak wynika z powyższej tabeli, zarówno instrukcja Set, jak i Reset wykonywane są w dwóch taktach zegara. Widać również, że każda z tych instrukcji zajmuje **0,00025 ms**, czyli **250 ns**.

Podłączając oscyloskop do naszego układu możemy odczytać, że stan „1” na oscyloskopie trwa 252ns, a zbocze opadające („0”) – 500ns, wartość ta jednak wynika z wykonania instrukcji Reset w dwóch taktach, oraz instrukcji Loop również w dwóch taktach. Wynika z tego czas wykonywania instrukcji **Reset = 250ns** oraz **Loop = 250ns**. Wartości te pokrywają się prawie idealnie z wartościami odczytanymi z oscyloskopu.

Wynika z tego, że okres zapalenia i zgaszenia diody trwa

$$T = 252 \text{ ns} + 500 \text{ ns} = 752 \text{ ns},$$

więc możemy policzyć częstotliwość:

$$f = 1/T = 1/752\text{ns} = 1,3297\text{MH}$$

z czego wynika, że w ciągu jednej sekundy dioda zapali się i zgaśnie **1 329 700** razy.

Instrukcja Toggle

Aby sprawdzić działanie instrukcji Toggle musimy przygotować kod:

<code>\$regfile = "m8def.dat"</code>	Informuje kompilator o pliku dyrektyw mikroprocesora
<code>\$crystal = 8000000</code>	Informuje kompilator o częstotliwości oscylatora taktującego mikrokontrolera
<code>Config PINB.0 = Output</code>	Ustawiamy Pin B 0 jako wyjście
<code>Do</code>	Początek pętli
<code>Toggle</code>	zmiana stanu pinu B na przeciwny
<code>Loop</code>	Koniec pętli głównej programu
<code>End</code>	Koniec programu

Zależności czasowe instrukcji Toggle odczytane z symulatora programu BASCOM-AVR

Instrukcje	Czas [ms]	Cykle
Loop	0.77125	6170
Toggle	0.7715	6172
Loop	0.772	6176
Toggle	0.77225	6178

Z powyższej tabeli można odczytać, że instrukcja **Toggle** wykonuje się w czterech taktach zegara, oraz że czas wykonania instrukcji **Toggle** wynosi **0,0005mS**, czyli **500ns**.

Po podłączeniu oscyloskopu do naszego układu możemy odczytać, że stan „1” trwa **744ns**, podobnie jak czas „0” logicznego – również **744ns**.

Wynika z tego, że okres zapalenia i zgaszenia diody trwa

$$T = 744\text{ns} + 744\text{ns} = 1\,488\text{ns}$$

więc możemy policzyć częstotliwość:

$$f = 1/T = 1/1\,488\text{ns} = 0,62720\text{MHz}$$

z czego wynika, że w ciągu jednej sekundy dioda zapali się i zgaśnie **627 200** razy. Widzimy, że Toggle miga ponad dwa razy wolniej niż instrukcje Set/Reset.

Program w języku ASSEMBLER

Aby sprawdzić działanie programu w języku ASSEMBLER musimy przygotować kod:

```
.nolist
.include "m8def.inc"
ldi r16, 0b00000001
out ddrb, r16
ldi r17, 0b00000000
loop:
    out portb, r16
    out portb, r17
    rjmp loop
.exit
```

Wartości dla programu napisanego w ASSEMBLERZE możemy odczytać jedynie za pomocą oscyloskopu. Obserwujemy, że czas trwania „1” logicznej wynosi 128ns, a czas trwania „0” logicznego wynosi **360ns**.

Wynika z tego, że okres zapalenia i zgaszenia diody trwa

$$T = 128\text{ns} + 360\text{ns} = 488\text{ns}$$

więc możemy policzyć częstotliwość:

$$f = 1/T = 1/488\text{ns} = 2,04918\text{MHz}$$

z czego wynika, że w ciągu jednej sekundy dioda zapali się i zgaśnie **2 049 180** razy.

Podsumowanie

Jak możemy zauważyć, program napisany w języku ASSEMBLER ma najwyższą częstotliwość z badanych programów, a najwolniejszy jest program napisany z użyciem instrukcji Toggle.