



Instytut Teleinformatyki



Wydział Fizyki, Matematyki i Informatyki
Politechnika Krakowska

Mikroprocesory i mikrokontrolery

„Opóźnienia”

laboratorium: 08
autor: mgr inż. Katarzyna Smelcerz

Kraków, 2015

Spis treści

Spis treści	2
1. Wiadomości wstępne	3
1.1. Niezbędne wiadomości	3
1.2. Generowanie prostego opóźnienia.....	3
2. Przebieg laboratorium	4
2.1. Zadanie 1. Na ocenę 3.0 (dst)	4
2.2. Zadanie 2. Na ocenę 4.0 (db).....	4
2.3. Zadanie 3. Na ocenę 5.0 (bdb)	4

1. Wiadomości wstępne

Pierwsza część niniejszej instrukcji zawiera podstawowe wiadomości teoretyczne dotyczące omawianego tematu. Poznanie tych wiadomości umożliwi prawidłowe zrealizowanie praktycznej części laboratorium.

1.1. Niezbędne wiadomości

Pamięć programu, wewnętrzna pamięć RAM, mapa pamięci, rejestry, tryby adresowania oraz lista instrukcji, dostępna w dokumentacji:
<http://www.atmel.com/images/doc2503.pdf>

Schemat płytki ewaluacyjnej AVRZL15
<http://www.cyfronika.com.pl/uruchomieniowe/zl15avr.pdf>

Bieżące laboratorium jest przeprowadzane w środowisku AVR Studio4. Do fizycznego zaprogramowania mikrokontrolera używany jest programator (np. KamPROG)

1.2. Generowanie prostego opóźnienia

Celem laboratorium jest zaimplementowanie opóźnienia poprzez stworzenie prostego licznika w parze rejestrów R16 oraz R17 (Listing 1). Licznik działa, tak długo, aż nie nastąpi przepełnienie rejestrów – gdy w wyniku operacji arytmetycznej wystąpi zero, zakończy się działanie podprogramu.

Listing 1 Przykładowy licznik zaimplementowany na parze rejestrów roboczych

```
Delay:                                     ;odczekanie pewnego czasu

        LDI  R16,0                          ; R16 = 0
        LDI  R17,0                          ; R17 = 0

Counter:
        INC  R16                            ; inkrementuj R16
        BRNE Counter                       ;wykonaj skok do etykiety, gdy R16 różne od 0
        INC  R17                            ; inkrementuj R17
        BRNE Counter                       ;wykonaj skok do etykiety, gdy R17 różne od 0
        ret
```

2. Przebieg laboratorium

Druga część instrukcji zawiera zadania do praktycznej realizacji, które demonstrują zastosowanie technik z omawianego zagadnienia.

2.1. Zadanie 1. Na ocenę 3.0 (dst)

Proszę uruchomić program z Listingu 1 w trybie symulacji i prześledzić pracę krokową. Proszę zwrócić uwagę, którą flagę i na jaką wartość ustawia instrukcja BRNE (proszę obserwować rejestr SREG).

2.2. Zadanie 2. Na ocenę 4.0 (db)

Proszę napisać program wywołujący (polecenie recall), opóźnienie przedstawione na Listingu 1. Po wciśnięciu przycisku na joysticku, ma zmieniać się stan diody D2. Po upływie pewnego czasu, dioda ma powrócić do stanu pierwotnego.

2.3. Zadanie 3. Na ocenę 5.0 (bdb)

Proszę rozbudować zadanie na ocenę 4.0 (db) z optymalizacją kodu. W poprzednich laboratoriach do sprawdzenia stanu pinu były wykonywane polecenia przedstawione na Listingu 2. Proszę zastąpić działanie poleceń ANDI i CPI jednym, tj. tylko ANDI. Podpowiedź: proszę zastanowić się nad przesunięciem bitowym.

Listing 2

```
IN R16, PINA ; wczytanie PINA do R16
ANDI R16, 0x03 ;sprawdzenie stanu pinu PA0
CPI R16, 0x02
```