|  |  |  |  |
| --- | --- | --- | --- |
| * + - * 1. Termin zajęć   Czwartek  13:15-15:00 | **Podstawy Techniki Mikroprocesorowej** | | |
| **Osoby wykonujące ćwiczenie:**  Michał Bernacki-Janson, Adam Czekalski | | | **Grupa nr:**  **D** |
| **Tytuł ćwiczenia:**  Arytmetyka, logika, pomięć, diody i brzęczyki | | | **Ćwiczenie nr:**  **1** |
| **Data wykonania ćwiczenia** | | 16.03.2023 | **Ocena:** |
| **Data oddania sprawozdania** | | 26.03.2023 |  | |

1. **Wstęp**

Celem pierwszych zajęć było zapoznanie się z działaniem mikrokomputera 8051. Wykonano programy w języku assembler wykonujące: podstawowe operacje arytmetyczne z wyświetleniem wyników na diodach, sortowanie bąbelkowe, wyświetlanie sekwencji na diodach i program, który obsługiwał brzęczyk.

1. **Program wykonujący podstawowe operacje arytmetyczne na dwóch argumentach 8-bitowych i dwóch argumentach 16-bitowych, prezentacja wyników na diodach.**

Kod z komentarzem:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Wykonujemy dodawanie, odejmowanie, mnożenie, dzielenie oraz dodawanie dwóch liczb 16-bitowych zapisanych na dwóch rejestrach. Na diodach wyświetlane są wyniki każdej z operacji arytmetycznej.

1. **Program realizujący ciekawe zapalanie/gaszenie diod podłączonych do portu P1.**

Kod z komentarzem:

Obraz zawierający tekst

Opis wygenerowany automatycznie

W programie wykorzystano „pętlę” tworzącą opóźnienie. Jej działanie opiera się na tym, że do rejestru r0 wpisujemy wartość FF, następnie do rejestru r1 wpisujemy wartość FA, a potem, dopóki wartość przechowywana w rejestrze r1 nie będzie równa 0, dekrementujemy ją. Następnie dekrementujemy wartość w rejestrze r0 i znów ładujemy wartość FA do rejestru r1 i powtarzamy cały ten proces, dopóki wartość w rejestrze r0 nie będzie równa 0. „Pętlę” wywołujemy po każdym zapaleniu sekwencji diod.

Tak wygląda sekwencja w praktyce:

Obraz zawierający diagram

Opis wygenerowany automatycznie

1. **Program wykonujący dźwięk za pomocą brzęczyka podłączonego do portu P3.2**

Obraz zawierający tekst

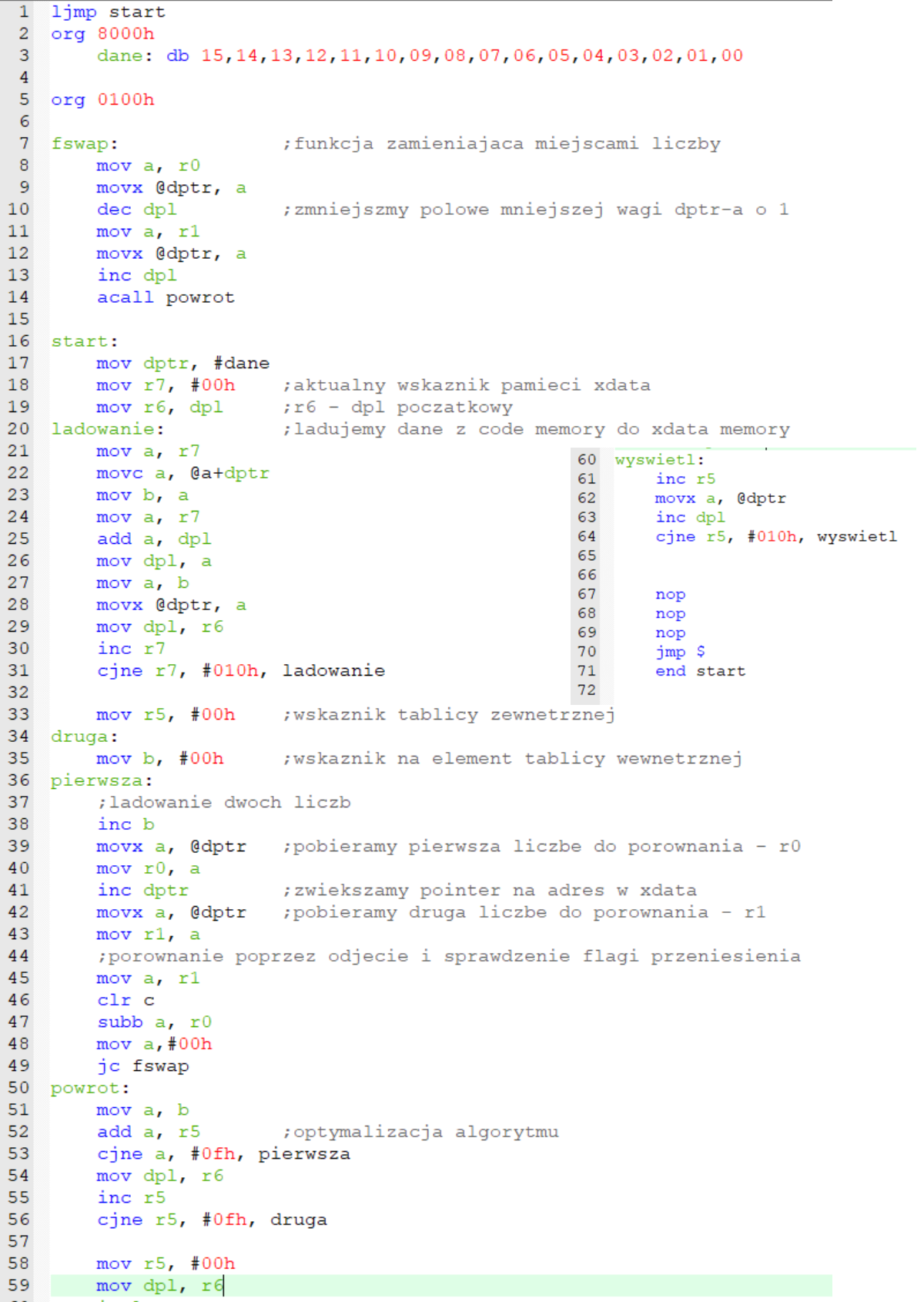
Opis wygenerowany automatycznie

Tak jak w przypadku programu z sekwencją diod, tutaj też wykorzystujemy pętlę tworzącą opóźnienie w celu spowolnienia zmiany stanu brzęczyka. Na początku zastosowano większe opóźnienie (przy etykiecie „one” do rejestru r1 przenoszono większą wartość), co za tym idzie zmniejszono częstotliwość, w efekcie czego brzęczyk stukał. Następnie zmniejszono opóźnienie w programie, co spowodowało wydobycie się z brzęczyka dźwięku o większej częstotliwości.

Dźwięk z brzęczyka został wydobyty poprzez negację bitu 2 w porcie P3.

1. **Program realizujący sortowanie bąbelkowe lub znajdujący minimum albo maksimum tablicy 1-wymiarowej rozpoczynającej się od adresu 8000H pamięci zewnętrznej danych (XRAM) i obejmującej 16 kolejnych komórek tej tablicy.**

Kod z komentarzem:



Program realizuje algorytm sortowania bąbelkowego. Dane do programu dla wygody wprowadzania najpierw są wprowadzane do Code Memory, po czym zostają przepisane do Xdata Memory. Pobierane są dwie kolejne liczby do rejestrów, a następnie są porównywane. Z powodu braku operacji porównywania, użyto odejmowania – jeśli wynik odejmowania jest ujemny to flaga C zostanie ustawiona na 1 i dokonuje się skok do fswap (zamiana miejscami elementów). Algorytm został zoptymalizowany przez ograniczenie iteracji pętli wewnętrznej o aktualny iterator pętli zewnętrznej.