**Uniwersytet Przyrodniczo-Humanistyczny w Siedlcach**

**Wydział Nauk Ścisłych**

**Kierunek: Informatyka**

Projekt indywidualny z przedmiotu Architektura Systemów Komputerowych

Temat: Wykonywanie złożonych obliczeń matematycznych

Autor: Damian Dąbrowski

**Siedlce 2016**

Spis treści

**1. Sformułowanie zagadnienia**

**2. Założenia**

**3. Ograniczenia**

**4. Program: rozwiązanie**

5. Instrukcja **dla** użytkownika

**6**. **Literatura**

**1. Sformułowanie zagadnienia**

Celem tego projektu jest opracowanie algorytmu do obliczania złożonego wyrażenia matematycznego oraz implementacja go w języku asembler za pomocą symulatora mikroprocesora sms32v50.

Wyrażenie: b3+a2-/((2/c+c3)\*3)\*a/4+(-23)

**2. Założenia**

* W rozwiązaniu wykorzystujemy możliwości symulatora procesora 8086, lista możliwości jest podana w pliku z dokumentacją: sms32v50.doc (jest on dostępny po instalacji systemu), procedury, procedury z parametrami, opis zadania, wyjście itd.
* Należy przygotować dokumentację, dokumentację dostarczamy w formie elektronicznej,
* Projekt należy obronić (ostateczny termin – dzień przed egzaminem, inaczej nie ma dopuszczenia do egzaminu, dopuszczalny termin dodatkowy – przed II terminem lub gdy koordynator przedmiotu określi inaczej),
* Za projekt można uzyskać maksymalnie 70pkt. Na studiach stacjonarnych i 40pkt na niestacjonarnych.
* W projekcie wykorzystujemy różne urządzenia zewnętrzne do opisu stanu obliczanego wyrażenia. Np. przepełnienie można zasygnalizować odpowiednimi światłami itd.
* Zmienne wczytujemy z klawiatury.
* Korzystamy z funkcji, tablic, stosu itd.

**3. Ograniczenia**

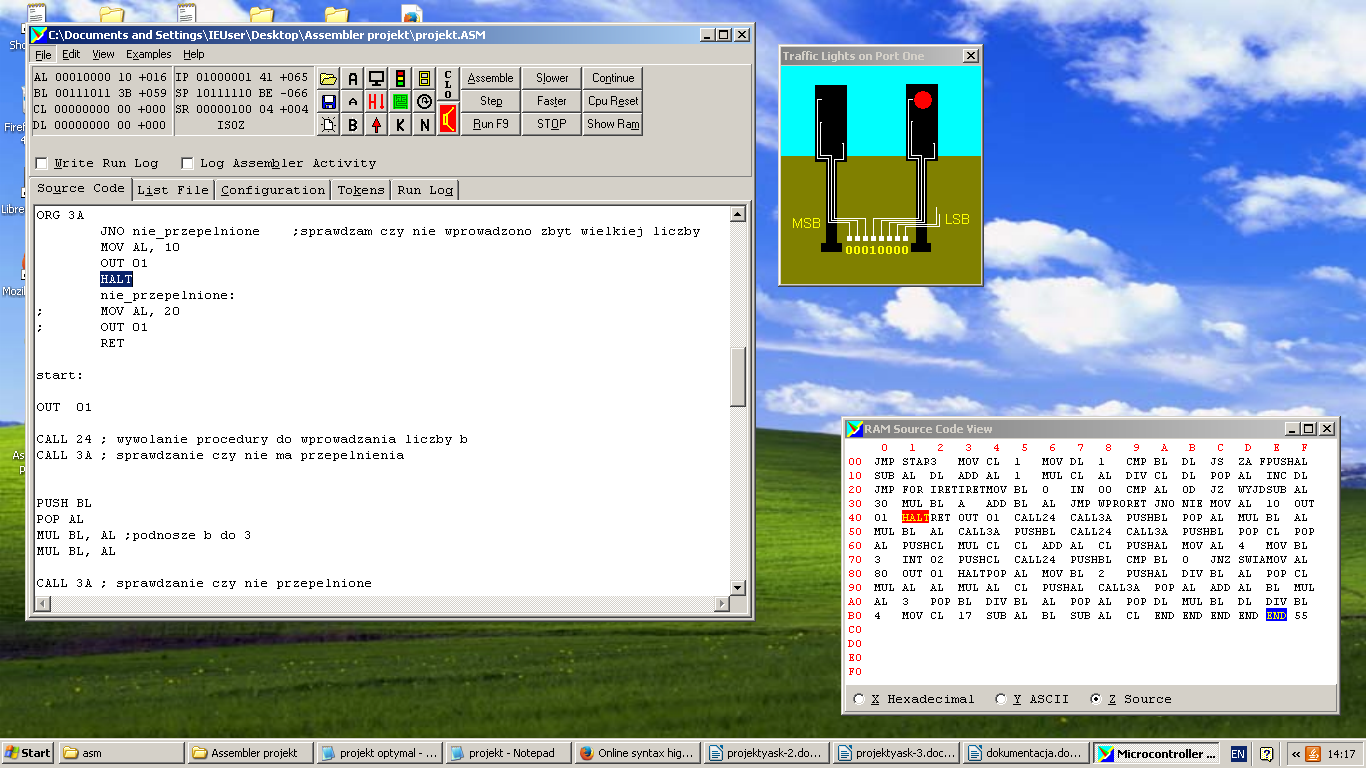
Użytkownik wprowadza cyfry z klawiatury, które składają się na liczby, gdy liczba jest już kompletna użytkownik zatwierdza ją enterem. ,

W przypadku przekroczenia dopuszczalnego zakresu podczas obliczeń ,który wynosi: -128 do 127, ustawiona zostaje flaga przepełnienia oraz zapala się czerwona lampka na sygnalizatorze, a wykonywanie programu zostaje zatrzymane.

**4. Program: rozwiązanie**

|  |
| --- |
| **JMP** start  DB 3  ORG 3; prodecdura wyliczajaca symbol newtona AL to n a BL to k, wynik w CL  **MOV** CL, 1 ; wynik  **MOV** DL, 1 ; iterator  for:  **CMP** BL, DL ; i<=k zamieniam na k>=i, i skacze dopiero jesli jest ujemna flaga  **JS** za\_for  **PUSH** AL  **SUB** AL, DL ; n - i  **ADD** AL, 1 ; n- i +1  **MUL** CL, AL ; wynik \*= n- i + 1  **DIV** CL, DL ; wynik \*= (n- i + 1)/i  **POP** AL  **INC** DL ;i++  **JMP** for  za\_for:  ;wynik w CL;  **IRET**  ;równowazne z MOV CL, 4  **IRET**  ORG 24 ; procedura do wprowadzania liczb - do BL  **MOV** BL, 0  wprowadz\_cyfre:  **IN** 00 ; podaje liczbe  **CMP** AL, 0D ; sprawdzam czy wprowadzona liczba jest eneterem  **JZ** wyjdz\_cyfra  **SUB** AL, 30 ; przetwarzam znak ASCII na cyfre  **MUL** BL, A ; mnoze poprzednia cyfre tak aby znalaza sie rzad wyzej  **ADD** BL, AL ; dodaje cyfre do liczby  **JMP** wprowadz\_cyfre  wyjdz\_cyfra:  **RET**  ORG 3A  **JNO** nie\_przepelnione ;sprawdzam czy nie wprowadzono zbyt wielkiej liczby  **MOV** AL, 10  **OUT** 01  HALT  nie\_przepelnione:  **RET**  start:  **OUT** 01  **CALL** 24 ; wywolanie procedury do wprowadzania liczby b  **CALL** 3A ; sprawdzanie czy nie ma przepelnienia  **PUSH** BL  **POP** AL  **MUL** BL, AL ;podnosze b do 3  **MUL** BL, AL  **CALL** 3A ; sprawdzanie czy nie przepelnione  **PUSH** BL  **CALL** 24 ; wywolanie procedury do wprowadzania liczby a  **CALL** 3A  **PUSH** BL  **POP** CL  **POP** AL  **PUSH** CL ; wrzucam a na stos w celu pozniejszego wykrzystania  **MUL** CL,CL  **ADD** AL, CL ; dodaje b^3 + a^2  **PUSH** AL  **MOV** AL, 4 ; wprowadzanie danych do obliczenia symbolu newtona  **MOV** BL, 3  **INT** 02 ; przerwanie  **PUSH** CL; wrzucam wynik symbolu na stos  **CALL** 24 ; wywolanie procedury do wprowadzania liczby c  **PUSH** BL  **CMP** BL, 0 ; sprwadzam czy nie ma dzielenia przez 0  **JNZ** swiatla  **MOV** AL, 80 ;czerwone  **OUT** 01  HALT  swiatla:  **POP** AL  **MOV** BL, 2  **PUSH** AL  **DIV** BL, AL ; 2/c  **POP** CL  **MUL** AL, AL  **MUL** AL, CL ; c^3  **PUSH** AL  **CALL** 3A ; sprawdzam czy nie przepelnione  **POP** AL  **ADD** AL, BL; 2/c + c^3  **MUL** AL, 3; (2/c + c^3)\*3 ; wynik  **POP** BL ;zdejmuje symbol  **DIV** BL, AL ; symbol /wynik  **POP** AL ; zdejmuje b^3 + a^2 ;  **POP** DL; zdejmuje a ze stosu  **MUL** BL, DL; od symbolu do \*a  **DIV** BL, 4 ; wynik/4  **MOV** CL, 17  **SUB** AL, BL ; b^3 + a^2 minus dwumian do a/4  **SUB** AL, CL  END |

**5. Instrukcja dla użytkownika**

Użytkownik wprowadza po jednej cyfrze, która z kolei składa się na liczbę, zakończenie wprowadzania tych cyfr użytkownik sygnalizuje enterem.

Jeżeli użytkownik wprowadzi liczbę wykraczającą poza zakres program zatrzyma się i zasygnalizuje przepełnienie za pomocą czerwonej lampki na sygnalizatorze.

Przykładowy wynik programu dla b=3, a=2, c=1 to +008

Wynik wyrażenia umieszczany jest w rejestrze AL

**6. Literatura**

1. Kruk S.: Asembler - podręcznik użytkownika. Wyd. MIKOM. Warszawa 1999

2. Kruk S.: Asembler. Kurs programowania dla średnio zaawansowanych.

Wyd. MIKOM. Warszawa 2001

3. Instrukcja użytkownika symulatora sms32v50 online, adres: http://www.softwareforeducation.com/sms32v50/sms32v50\_manual/index.htm