ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(государственный технический университет)

Кафедра 304
(вычислительные машины, системы и сети)

Лабораторная работа по курсу «Ассемблер»

Отчёт по работе *№1*.

Реализации линейного процесса
(наименование работы)

на языке программирования Assembler

Вариант задания *№2*.

Лабораторную работу выполнили:

студенты гр. 13-501, Резвяков Денис, Коршунов Евгений.
(должность) *(Ф. И. О.)*

Лабораторную работу принял:

зам. декана фак. №3, Алещенко Алла Степановна _____.
(должность) *(Ф. И. О.)* *(подпись)*

«*10*» *ноября* 2009 г.
(дата приёма)

Цель работы: Освоить программирование линейных участков программ с использованием арифметических команд языка Ассемблера.

Задание

1. Разобрать раздел справочно-обучающей системы ASML десятичной коррекции чисел при вычитании. Особое внимание уделить числовым примерам. Законспектировать этот раздел.

2. Разработать программу на Турбо-Паскале для реализации линейного процесса, содержащую также вычисление эталонного результата.

Расчётная формула: $c \times d + (a \bmod b) + (a^2 - 7)$.

Аргументы: $a = 29$, $b = 18$, $c = 31$, $d = -14$.

Десятичная коррекция чисел при вычитании

В BCD-формате каждый байт содержит две десятичные цифры (по одной в каждой тетраде). Максимальное значение тетрады равно 9, максимальное значение байтового числа — 99. Многоразрядные числа представляются последовательностью байт. Например, число 0011.0100 в двоичном формате равно десятичному числу 52, а тоже число в BCD-формате равно десятичному числу 34.

Вычитание чисел производится в два этапа: сначала байты вычитаются как обычные двоичные числа, а затем результат корректируется командой **DAS** по следующим правилам:

1) Если AF = 1 или младшая тетрада AL больше девяти, то из содержимого AX вычитается 6 и AF устанавливается в 1.

2) Если CF = 1 или старшая тетрада AH больше девяти, то из содержимого AX вычитается 60h и CF устанавливается в 1.

После коррекции в регистре AX остается упакованное десятичное число в диапазоне от 0 до 99.

Команда DAS имеет длину 1 байт и выполняется за 4 машинных такта.

Устанавливаются все арифметические флажки в зависимости от полученного результата, за исключением OF и AF, состояние которых после коррекции не определено. CF интерпретируется как флажок заёма, он равен 1, если вычитаемое больше уменьшаемого.

Вычитание BCD-чисел с повышенной точностью аналогично двоичному вычитанию с повышенной точностью. Но после каждого байтового вычитания необходимо производить десятичную коррекцию.

Пример вычитания BCD-слов

Вычтем: `0000.0011.0000.0111`(307)–`0000.0010.0000.1000`(208).

1. Вычитание младших байт командой SUB.

`0000.0111 – 0000.1000 = 1111.1111`. Произошло два заёма, поэтому флажки AF и CF устанавливаются в состояние 1.

2. Десятичная коррекция промежуточного результата: AF = 1 или младшая тетрада > 9, значит из младшей тетрады вычитается шестёрка (0110). CF = 1 или старшая тетрада > 9, значит и из старшей тетрады вычитается 6 (0110), а флажок CF остается в состоянии 1. Результат после коррекции:

`1111.1111 – 0000.0110 – 0110.0000 = 1001.1001`.

3. Вычитание старшего байта числа командой SBB:

`0000.0011 - 0000.0010 - 1` (заём) `= 0000.0000`. AF = CF = 0.

4. Десятичная коррекция вычитания старших байт не изменяет промежуточного результата, т.к. значения тетрад допустимые и флажки CF и AF равны 0.

Окончательный результат равен десятичному числу 99.

Код программы реализации линейного процесса

```
program Laba1;

var
  a, b, c, d, r1, r2: integer;

begin
  a :=  29;
  b :=  18;
  c :=  31;
  d := -14;

  asm
    mov  AX, c  {помещаем значение C в регистр AX}
    imul d      {умножаем C (в AX) на D}
    mov  r2, AX {копируем результат из AX в переменную r2}

    mov  AX, a  {копируем значение A в регистр AX}
    cwd         {расширяем значение в AX до (DX:AX), до двойного слова}
    idiv b      {делим (DX:AX) на значение B}
    add  r2, DX {добавляем остаток от деления (из DX) в переменную r2}

    mov  AX, a  {копируем значение A в регистр AX}
    imul AX     {умножаем A (в AX) само на себя}
    sub  AX, 7  {вычитаем из произведения 7}
    add  r2, AX {добавляем разность (из AX) в переменную r2}
  end;

  r1 := c*d + (a mod b) + (a*a - 7);
  WriteLn('Result by Pascal: ', r1:4);
  WriteLn('Result by Assembler: ', r2:4);
  ReadLn;
end.
```

Результат работы программы

```
Result by Pascal:  411
Result by Assembler:  411
```