Содержательный процесс разработки ассемблерной программы

1. Выбрать сегментную структуру будущей программы, символические имена сегментов, регистры-указатели сегментов
2. Определить место размещения данных (в регистрах и/или в памяти), формат данных, символические адреса для данных в памяти
3. Разработать и изобразить укрупненный (для достаточно обширной логики) и детальный алгоритм действий для процессора
4. Написать исходный текст программы (с комментариями).
5. Транслировать в объектный код, получить протокол трансляции
6. Подготовить исходные данные для проверки исполнением всех ветвей алгоритма , используя конкретные внутрисегментные адреса из протокола трансляции. Выполнить расчет ожидаемых в регистрах и/или памяти результатов
7. Получить исполняемый файл программы.
7. Выполнить отладку исполняемого кода через Отладчик. Убедиться в соответствии расчетных и получаемых результатов

Пример документальной подготовки ассемблерной программы

Задача. В сегменте данных размещены два однобайтных беззнаковых числа и 4 символа. Получить разность чисел в формате двойного слова и записать в другой сегмент данных. Последний из символов заменить на первый числовой байт, взятый с обратным знаком.

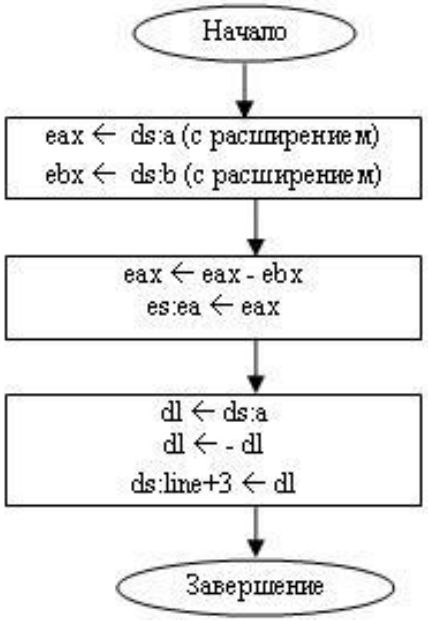
1. Сегментная структура программы
- два сегмента данных: dseg (указатель DS) и eseg (указатель ES)
- кодовый сегмент cseg (указатель CS)

2. Размещение данных в памяти: их символические адреса и форматы
ds:a – адрес первого числа. Однобайтное, беззнаковое
ds:b – адрес второго числа. Однобайтное, беззнаковое
ds:line – адрес последовательности из 4-х символьных байтов
es:ea – адрес для записи разности. Двойное слово .

3. Использование регистров:
4-х байтные регистры - для расширения числовых байтов до двойных слов,
1-байтный регистр – для получения значения байта с обратным знаком

4. Укрупненный алгоритм

Детальный алгоритм действий



5. Исходный текст программы со сведениями об авторе, теме и номере задания

```
; Тема «Пересылка данных» Задание 7. Иванов С. УИБ-311
.386
dseg segment use16
        a     db  34
        b     db  75h
        line  db  '####'
dseg ends
eseg segment use16
        ea   dd    ?
eseg end
cseg segment use16
assume ds:dseg, cs:cseg, es:eseg
; загрузка регистров-указателей сегментов ds и es
m1:    mov  cx, dseg
        mov  ds, cx
        mov  cx, eseg
        mov  es, cx
; расширим байты до 4-байтных, вычислим разность и запишем по адресу es:ea
        movzx  eax, ds:a
        movzx  ebx, ds:b
        sub  eax, ebx
        mov  es:ea, eax
; заменим последний символ на 1-й числовой байт с обратным знаком
        mov   dl,  ds:a
        neg   dl
        mov   ds:line+3, dl
; завершение исполнения
        mov ah, 4ch
        int  21h
cseg ends
end  m1
```

6. Протокол трансляции

```
1  ;  Задание1.  Иванов С. УВВ-211,  вариант 7
2                                      .386
3          0000                        dseg segment use16
4          0000  22                   a     db   34
5          0001  75                   b     db  75h
6          0002  23 23 23 23           line  db  '####'
7          0006                        dseg ends
8          0000                        eseg segment use16
9          0000  ?? ?? ?? ??                 ea   dd   ?
10         0004                        eseg ends
11         0000                        cseg segment use16
12                               assume ds:dseg, cs:cseg, es:eseg
13   ; загрузка регистров – указателей сегментов
14         0000  B9 0000 s    m1:      mov cx, dseg
15         0003  8E D9                 mov ds, cx
16         0005  B9 0000 s             mov cx, eseg
17         0008  8E C1                 mov es, cx
18 ; расширим байты до 4-байтных, определим разность и запишем в es:ea
19         000A  66| 0F B6 06 0000 r          movzx eax, ds:a
20         0010  66| 0F B6 1E 0001 r          movzx ebx, ds:b
21         0016  66| 2B C3             sub eax, ebx
22         0019  66| 26: A3 0000 r           mov es:ea, eax
23 ; заменим последний символ на 1-й числовой байт с обратным знаком
24         001E   8A 16 0000 r                mov dl, ds:a
25         0022  F6 DA                 neg dl
26         0024  88 16 0005 r                 mov ds:line+3, dl
27 ; завершение исполнения
28         0028  B4 4C                 mov ah, 4ch
29         002A  CD 21                 int 21h
30         002C                        cseg ends
31                               end m1
```

7. Данные для отладки: исходные и ожидаемые результаты

Исходные данные (в hex)		
Симв.адрес	**Адрес размещения**	**Значения байтов памя-ти**
a	**DS: 0000**	**22**
b	**DS: 0001**	**75**
line	**DS: 0002 ÷ 0005**	**23 23 23 23**

Расчет ожидаемых результатов:

- разность: 00 00 00 22h – 00 00 00 75h = ff ff ff ad
- первый числовой байт с обратным знаком: 00 - 22h = de

Ожидаемый результат (в hex)		
Симв. адрес	**Адреса размещения**	**Значения байтов памя-ти**
line	**DS: 0002 ÷ 0005**	**23 23 23 DE**
ea	**ES: 0000 ÷ 0003**	**AD FF FF FF**

8. Протоколы отладки

1) Исходные данные

На рис.1 показаны выделенные в окне сегмента данных исходные байты, которые начинаются с адреса ds:0000 - 22 75 23 23 23 23

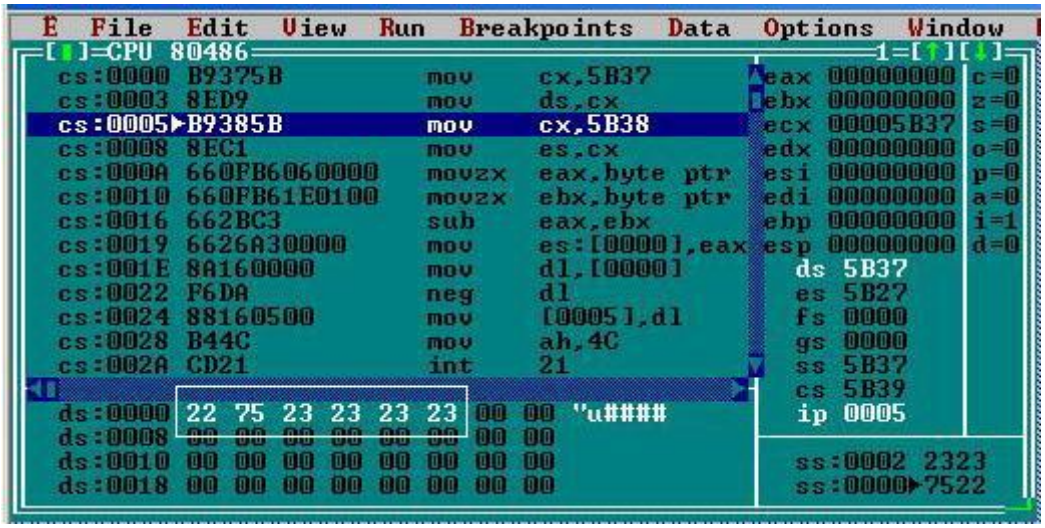


Рис.1 – Исходные байты в сегменте данных

2) Измененный последний байт строки символов.

Его адрес ds:0005. На рисунке 2 он показан в окне сегмента данных: DE

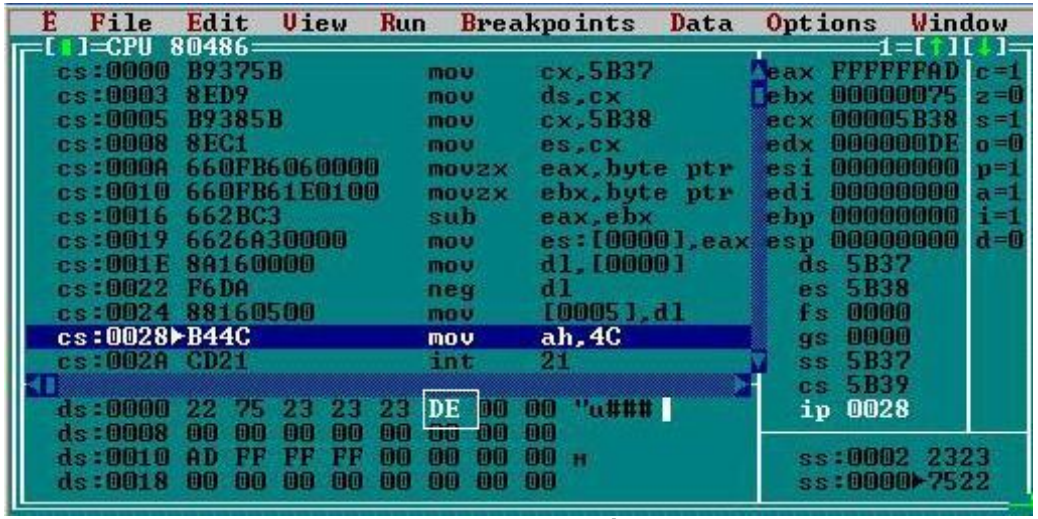


Рис.2 – Измененный байт по адресу ds:0005

3) Полученная разность байтов в формате двойного слова.

Ее полученное 4-х байтное значение FF FF FF AD показано на рис.3. Оно выделено в окне сегмента данных, начинается с адреса es:0000.

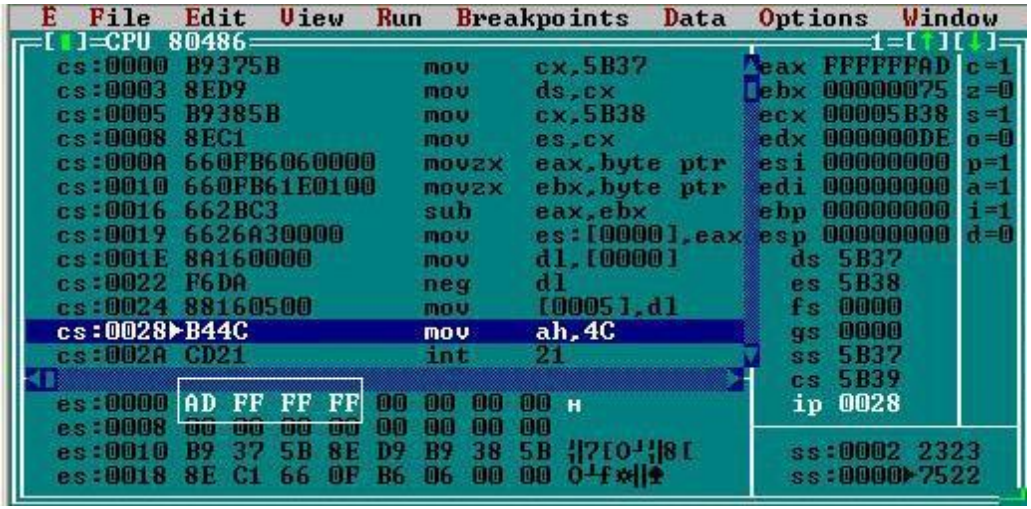


Рис.3 – Полученная разность

Выводы: ожидаемые результаты исполнения программы соответствуют расчетным.