

ПРОГРАММИРОВАНИЕ НА АССЕМБЛЕРЕ  
ЛАБОРАТОРНАЯ РАБОТА №1

Тема: Арифметические команды языка ассемблер

Цель работы

Изучить команду пересылки данных MOV МП 8086. Изучить арифметические команды МП 8086. Научиться использовать транслятор Turbo Assembler и компоновщик Turbo Linker. Ознакомиться с отладчиком Turbo Debugger.

Задание

- написать программу на ассемблере, вычисляющую значение выражения с использованием арифметических команд сложения, вычитания, умножения и деления;
- проверить работу программы в отладчике.

Вариант 2	$(a - b) / (2 + c) * d$
-----------	-------------------------

Программа

```
CODE SEGMENT
    ORG 100h
    ASSUME CS:CODE
START:
    MOV CX, c
    ADD CX, 2

    MOV AX, a
    MOV BX, b
    SUB AX, BX

    IDIV CX

    MOV DX, d
    IMUL DX

a DW 12
b DW 4
c DW 4
d DW 2
CODE ENDS
END START
```

## Тестирование

№	a	b	c	d	Результат
1	12	4	4	2	2
2	13	-7	-1	3	60
3	70	-7	-9	11	-121
4	-7	-23	4	-11	-22
5	-23	-13	0	4	-20

1)

```

01 CODE SEGMENT
02 ORG 100h
03 ASSUME CS:CODE
04 START:
05 MOV CX, c
06 ADD CX, 2
07
08 MOV AX, a
09 MOV BX, b
10 SUB AX, BX
11
12 IDIV CX
13
14 MOV DX, d
15 IMUL DX
16
17 a DW 12
18 b DW 4
19 c DW 4
20 d DW 2
21 CODE ENDS
22 END START

```

2)

```

01 CODE SEGMENT
02 ORG 100h
03 ASSUME CS:CODE
04 START:
05 MOV CX, c
06 ADD CX, 2
07
08 MOV AX, a
09 MOV BX, b
10 SUB AX, BX
11
12 IDIV CX
13
14 MOV DX, d
15 IMUL DX
16
17 a DW 13
18 b DW -7
19 c DW -1
20 d DW 3
21 CODE ENDS
22 END START

```

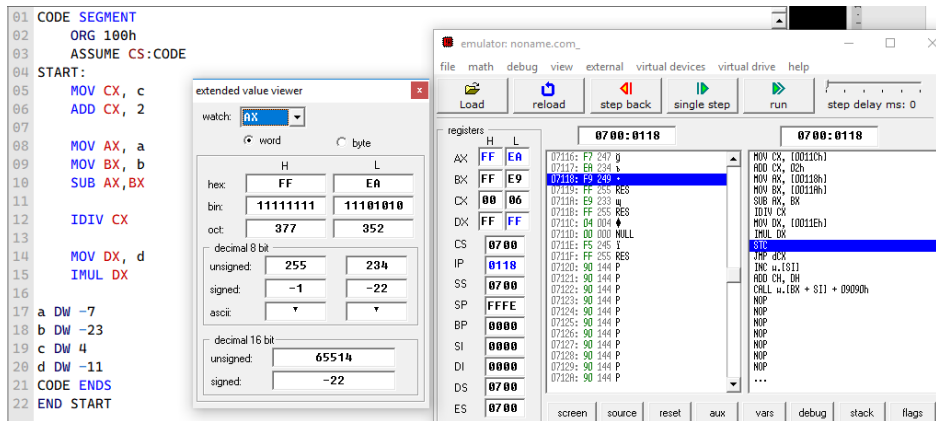
3)

```

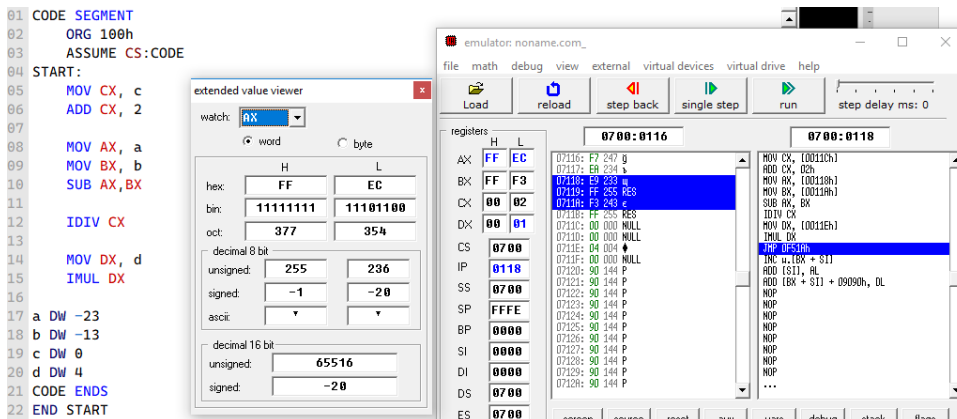
01 CODE SEGMENT
02 ORG 100h
03 ASSUME CS:CODE
04 START:
05 MOV CX, c
06 ADD CX, 2
07
08 MOV AX, a
09 MOV BX, b
10 SUB AX, BX
11
12 IDIV CX
13
14 MOV DX, d
15 IMUL DX
16
17 a DW 70
18 b DW -7
19 c DW -9
20 d DW 11
21 CODE ENDS
22 END START

```

4)



5)



## Контрольные вопросы

### 1. Для чего служит команда MOV?

По команде MOV происходит пересылка значения второго операнда на место первого операнда, флаги команда не меняет.

### 2. Какие требования предъявляются к операндам команды MOV?

Запрещена пересылка из одной ячейки памяти в другую, из одного сегментного регистра в другой и запрещена запись непосредственного операнда в сегментный регистр.

Размер пересылаемой величины определяется по типу операндов, указанных в команде. Если известны размеры обоих операндов, то эти размеры должны совпадать, иначе ассемблер зафиксирует ошибку.

### 3. Какие арифметические команды существуют для МП 8086?

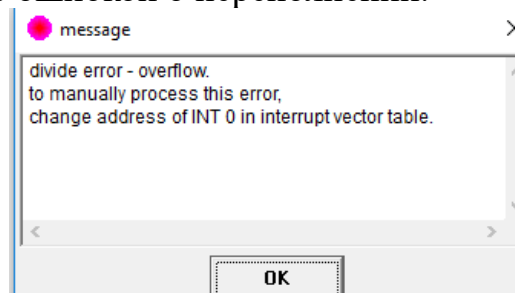
В ассемблере существует несколько основных арифметических команд: ADD – сложение операндов, SUB – разность операндов, INC – увеличение на 1, DEC – уменьшение на 1, NEG – изменение знака числа, ADC – сложение с учётом переноса, SBB – вычитание с учётом переноса, MUL и IMUL – умножение целых чисел, DIV и IDIV – деление целых чисел.

4. Чем отличаются команды IMUL и MUL?

MUL – умножение целых чисел без знака, IMUL – умножение целых чисел со знаком.

5. Что произойдет, если результат от деления не помещается в операнде назначения?

При выполнении деления возможно появление ошибки «деление на 0» или переполнение, то есть, если результат не помещается в операнде значения ассемблер выдаст окно с ошибкой о переполнении.



6. Из каких операций состоит процесс компиляции программы на ассемблере?

Для компиляции программ, написанных на языке ассемблера, используются следующие программы: транслятор Turbo Assembler (исполнительный файл TASM.EXE) и компоновщик Turbo Linker (исполнительный файл TLINK.EXE).

Транслятор выполняет ассемблирование программы – транслирует команды процессора в машинный объектный код и генерирует OBJ-модуль. На этапе ассемблирования транслятором могут выдаваться ошибки и предупреждения. Появление ошибки прекращает работу транслятора и OBJ-модуль не генерируется – необходимо устранить ошибку и провести повторное ассемблирование программы. Предупреждения не прекращают работу транслятора, OBJ-модуль генерируется, но программа может выполняться некорректно из-за наличия каких-то логических ошибок в ней, на что и указывают предупреждения.

Компоновщик выполняет преобразование OBJ-модуля в исполнительный файл COM или EXE. При использовании компоновщика имеется возможность объединить несколько OBJ-модулей в один исполнительный файл. Также на этапе компоновки есть возможность подключить к программе какие-либо библиотеки.