МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт информационных технологий |
| Кафедра математического и программного обеспечения |
| Программирование на ассемблере |
|  |

ЛАБОРАТОРНАЯ РАБОТА №1

|  |  |
| --- | --- |
| Исполнитель: | студент  группы 1ПИб-02-3оп-23  Богданов  Ренат Алексеевич |
| Руководитель: | Виноградова Людмила Николаевна |
| Оценка: |  |
| Подпись: |  |

2024 год

**Задание на лабораторную работу**

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

Вариант 19. a + 2 \* b – 3 \* (c – d).

**Код программы**

.model small

.stack 100h

.data

a db 14

b db 22

c db 35

d db 7

result db ?

.code

main:

mov ax, @data

mov ds, ax

; a+2\*b

mov bl, a

mov al, 2

imul b

add al, bl

mov cl, al

; 3\*(c-d)

mov al, c

mov bl, d

sub al, bl

mov bl, 3

imul bl

; a+2\*b-3\*(c-d)

mov bl, al

mov al, cl

sub al, bl

mov result, al

; Exit program

mov ax, 4C00h

int 21h

end main

**Результаты работы программы**

Таблица 1. Значения регистров и флага SF

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A10 | B10 | C10 | D10 | SF (+/-) | AL16 (AL10) | Итог |
| 1 | 2 | 3 | 4 | 0 (+) | 8 (8) | 8 |
| 1 | 2 | 4 | 3 | 0 (+) | 2 (2) | 2 |
| 14 | 22 | 35 | 7 | 1 (-) | E6 (230) | -26 |
| -1 | -1 | -3 | -5 | 1 (-) | F7 (247) | -9 |
| 0 | 0 | 0 | 0 | 0 (+) | 0 (0) | 0 |

**Контрольные вопросы**

1. *Для чего служит команда MOV?*

Команда MOV служит для пересылки данных между регистрами и ячейками памяти. Она копирует байт или слово из операнда-источника в операнд-приемник.

1. *Какие требования предъявляются к операндам команды MOV?*

* Операндами команды могут быть регистры или адресные выражения с прямой и косвенной адресацией.
* Операндами команды не могут быть два адресных выражения, но могут быть два регистра.
* Сегментный регистр CS не может быть операндом-приемником.
* При использовании в качестве операнда-приемника сегментных регистров DS, SS или ES, операнд-источник не может быть непосредственным значением.
* Оба операнда должны иметь одинаковый размер (байт или слово).
* В качестве операнда-источника может использоваться непосредственное значение, но при указании непосредственных значений меньших 256 и в качестве операнда-приемника адресных выражений с косвенной адресацией необходимо указать тип операнда (байт или слово).

1. *Какие арифметические команды существуют для МП 8086?*

* ADD: сложение.
* ADC: сложение с переносом.
* SUB: вычитание.
* SBB: вычитание с заемом.
* MUL: умножение беззнаковых чисел.
* IMUL: умножение знаковых чисел.
* DIV: деление беззнаковых чисел.
* IDIV: деление знаковых чисел.
* INC: увеличение значения на 1.
* DEC: уменьшение значения на 1.
* NEG: изменение знака.
* CBW: расширение байта до слова с учетом знака.
* CWD: расширение слова до двойного слова с учетом знака.

1. *Чем отличаются команды IMUL и MUL?*

* MUL: рассматривает свои операнды как беззнаковые числа.
* IMUL: рассматривает свои операнды как знаковые числа.
* Использование команды MUL для умножения чисел со знаком может привести к неверному результату, если числа отрицательные.
* Команда MUL может умножать 8-битные и 16-битные числа, результат записывается в AX или DX:AX соответственно.
* Команда IMUL также может умножать 8-битные и 16-битные числа, результат записывается в AX или DX:AX соответственно.

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

При делении может возникнуть ситуация, когда результат деления не помещается в операнде назначения (AX или AL) или при делении на 0. В этом случае процессором генерируется исключение переполнения при делении, и управление от программы передается обработчику этого исключения. Обработчик исключения обычно выдает сообщение об ошибке и завершает выполнение программы с передачей управления операционной системе. Поэтому важно избегать подобных ситуаций для предотвращения потери данных.

1. *Из каких операций состоит процесс компиляции программы на ассемблере?*

Ассемблирование: использование транслятора (например, Turbo Assembler) для преобразования исходного текста программы на ассемблере в машинный код (OBJ-модуль).

Компоновка: использование компоновщика (например, Turbo Linker) для преобразования OBJ-модуля в исполняемый файл форматов COM или EXE.

При успешном ассемблировании и компоновке генерируется исполняемый файл, который может быть запущен и выполнен операционной системой.

**Вывод**

В ходе выполнения лабораторной работы были освоены основы программирования на ассемблере, а также практические навыки работы с арифметическими командами процессора 8086. В частности, были использованы команды MOV, ADD, SUB и IMUL для выполнения сложения, вычитания и умножения.

Были закреплены навыки работы с регистрами и сегментными регистрами, а также организация данных и адресации в памяти. Особое внимание уделено корректной последовательности выполнения операций и учету знаковых свойств операндов.