

ЛАБОРАТОРНАЯ РАБОТА №1

Цель работы:

приобрести умения и практические навыки для работы с арифметическими командами и логическими операциями языка ассемблера, а также познакомиться с регистрами микропроцессора.

Теоретическая часть:

Выделяют несколько различных видов регистров микропроцессора. Для выполнения данной лабораторной работы понадобятся *регистры общего назначения*:

- *eax* – аккумулятор;
- *ebx* – базовый;
- *ecx* – счетчик;
- *edx* – данных.

| | 31 | 16 | 15 | 8 | 7 | 0 | |
|-----|----|----|----|---|----|----|----|
| EAX | | | | | AH | AL | AX |
| EDX | | | | | DH | DL | DX |
| ECX | | | | | CH | CL | CX |
| EBX | | | | | BH | BL | BX |

Рис. 1. Регистры общего назначения

Это 32-разрядные регистры, которые могут быть без ограничения использованы для любых целей. Младшие 16 разрядов этих регистров могут использоваться как самостоятельные 16-разрядные регистры с именами *ax*, *bx*, *cx* и *dx*. К старшим байтам этих регистров можно обращаться как к самостоятельным восьмиразрядным регистрам с именами *ah*, *bh*, *ch*, *dh*, а к младшим байтам – как к самостоятельным регистрам с именами *al*, *bl*, *cl*, *dl*.

Регистр *bx* используется как регистр для адресации и как вычислительный регистр. Регистр *cx* используется как счетчик, при этом в нем хранится число повторений команды или фрагмента программы. Регистр *dx* может быть использован в качестве расширителя аккумулятора (*ax*) при выполнении команд умножения и деления.

Каждый оператор языка может содержать несколько полей – поле метки, поле кода операции, поле операндов, поле комментариев. Единственное *обязательное поле* – поле операций. Оно определяет ту команду или директиву, которую должна выполнить ЭВМ. Команды транслируются в исполняемый код. Директива не вызывает появления кода, а управляет работой самого ассемблера.

Поле операндов – поле, которое содержит информацию о величинах, участвующих в операции. Количество операндов зависит от кода операции.

Поле метки – поле, которое используется для обозначения определенного места в программе. Метка получает значение, равное адресу следующей команды или данных. Если метка стоит перед командой, за ней ставится двоеточие. Метка состоит из букв, цифр и специальных символов. Метка не может начинаться с цифры.

Поле комментариев служит для указания дополнительной информации о команде или программе. Оно не влияет на выполнение программы, начинается с точки с запятой. Программы без комментариев не сопровождаемы.

Псевдокоманды определения данных (директивы) – директивы ассемблера, которые указывают, что в этом месте программы должны располагаться данные. Они задают тип данных, начальное значение и ставят в соответствие метку, которая служит для обращения к этим данным:

- *db* – 1 байт;
- *dw* – 2 байта (слово);
- *dd* – 4 байта (двойное слово);
- *dq* – 8 байт (четверное слово).

Поле операндов для этих директив может быть следующим:

- числа – десятичный код, шестнадцатеричные числа (в конце ставится *h*, начинаться должно с цифры или 0, если буква), двоичные числа (завершаются *b*) (например, *x dw 0Ah* или *y db 01100111b*);

– строки символов – последовательность символов, взятая в апострофы или кавычки. При определении строки используется *db* (например, *a db 'b'* или *s db 'stroka\$'*);

– знак резервирования памяти - ?.

В поле операндов допускается указывать список операндов, разделенных запятой. При *записи* – размещении данных в памяти – младшие байты операндов пишутся по младшим адресам памяти, а старшие – по старшим соответственно.

Для работы с данными в языке ассемблера существует множество различных команд, некоторые из которых далее будут рассмотрены более подробно.

Команда пересылки – *mov приемник, источник*. В данной команде *приемником* может быть регистр общего назначения, сегментный регистр (кроме регистра *cs*) или ячейка памяти. *Источник* – регистр общего назначения, сегментный регистр, ячейка памяти или непосредственный операнд. Операнды не могут одновременно находиться в памяти, должны быть одного размера, при этом нельзя выполнять пересылку из одного сегментного регистра в другой. Не допускается также пересылка в сегментный регистр непосредственного операнда.

Команды арифметики

– *add приемник, источник* – сложение – *приемник = приемник + источник*. Приемником может быть регистр общего назначения, ячейка памяти. Источником – регистр общего назначения, ячейка памяти, непосредственный операнд. Складывать значения в двух ячейках памяти одновременно нельзя;

– *sub приемник, источник* – вычитание – *приемник = приемник – источник*. Приемник и источник – по аналогии со сложением;

– *mul источник* – умножение без знаков – источником может быть регистр или ячейка памяти. Второй сомножитель – содержимое аккумулятора

(*al*, *ax*), то есть содержимое аккумулятора умножается на источник. То есть, результат вычисляется следующим образом: $ax(al) = ax(al) * источник$;

– *div источник* – деление без знаков – делимое по умолчанию в аккумуляторе, то есть содержимое аккумулятора делится на источник. Если частное не помещается в отведенную память (разрядную сетку) возникает ошибка (такое случается, когда делимое большое, а делитель очень мал). То есть, результат вычисляется следующим образом: $ax(al) = ax(al) \setminus источник$. Приемник и источник – по аналогии с умножением.

Логические команды

– *not приемник* – инверсия разрядов, применяется ко всем разрядам приемника;

– *and приемник, источник* – часто применяется для установки в ноль отдельных разрядов приемника;

| | | |
|------------|----------|----------|
| <i>and</i> | 0 | 1 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |

Табл. 1. Таблица истинности для команды *and*

– *or приемник, источник* – применяется для установки в единицу отдельных разрядов приемника;

| | | |
|-----------|----------|----------|
| <i>or</i> | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 1 |

Табл. 2. Таблица истинности для команды *or*

– *xor приемник, источник* – применяется для выборочной инверсии отдельных разрядов приемника, а также для обнуления значения регистра.

| | | |
|------------|----------|----------|
| <i>xor</i> | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |

Табл. 3. Таблица истинности для команды *xor*

Операнды не могут одновременно находиться в памяти и должны быть одного размера. Источником может быть регистр, ячейка памяти, непосредственный операнд. Выполняются команды над операндами поразрядно в соответствии с таблицей истинности каждой конкретной операции.

Задание 1. Дано X, Y. Вычислить Z. Задание можно выполнить как с использованием переменных, так и без, оба варианта допустимы.

Варианты:

1. $X=10, Y=16. Z=(2X+Y)*3;$
2. $X=8, Y=11. Z=(X+Y\backslash 2)*2;$
3. $X=22, Y=13. Z=(X-3Y)/4;$
4. $X=10, Y=11. Z=(4X+Y)*5;$
5. $X=13, Y=15. Z=(X+2Y)*6;$
6. $X=38, Y=17. Z=(5X-Y)/3;$
7. $X=10, Y=15. Z=(X+Y\backslash 4)*2;$
8. $X=29, Y=16. Z=(X\backslash 2-Y)*2;$
9. $X=49, Y=28. Z=(X\backslash 7-Y)*7;$
10. $X=51, Y=42. Z=(X-Y\backslash 3)*9;$
11. $X=12, Y=13. Z=(5X+Y)*2;$
12. $X=18, Y=10. Z=(X+Y\backslash 3)*4;$
13. $X=21, Y=17. Z=(X-2Y)/7;$
14. $X=20, Y=11. Z=(2X+Y)*3;$
15. $X=13, Y=14. Z=(X+3Y)*3;$
16. $X=39, Y=17. Z=(5X-Y)/2;$
17. $X=19, Y=12. Z=(X+Y\backslash 2)*2;$
18. $X=21, Y=16. Z=(X\backslash 3-Y)*4;$
19. $X=49, Y=28. Z=(X\backslash 7-Y)*7;$
20. $X=71, Y=42. Z=(X-Y\backslash 3)*2;$
21. $X=12, Y=19. Z=(2X+Y)*7;$

- 22. $X=1, Y=11. Z=(X+Y\backslash 2)*5;$
- 23. $X=25, Y=1. Z=(X-3Y)/8;$
- 24. $X=11, Y=10. Z=(4X+Y)*2;$
- 25. $X=11, Y=7. Z=(X+2Y)*3;$
- 26. $X=46, Y=110. Z=(5X-2Y)/3;$
- 27. $X=25, Y=16. Z=(X+Y\backslash 4)*4;$
- 28. $X=22, Y=10. Z=(X\backslash 2-Y)*2;$
- 29. $X=42, Y=28. Z=(X\backslash 7-Y)*4;$
- 30. $X=45, Y=42. Z=(X-Y\backslash 3)*8.$

Задание 2. Дано двоичное число 10110110, записанное в регистр *ah*. Задание выполняется с использованием логических команд. Для того, чтобы верно выполнить задание, необходимо подобрать такое восьмизначное двоичное число, чтобы при выполнении операции по заданию из изначального числа получилось число с измененными разрядами. Это число следует подбирать в соответствии с таблицами истинности логических команд.

Варианты:

- 1. обнулить второй и четвертый разряды регистра;
- 2. инвертировать третий и шестой разряды регистра;
- 3. обнулить первый и второй разряды регистра;
- 4. инвертировать третий и седьмой разряды регистра;
- 5. обнулить второй и четвертый разряды регистра;
- 6. инвертировать первый и пятый разряды регистра;
- 7. обнулить второй и седьмой разряды регистра;
- 8. инвертировать третий и пятый разряды регистра;
- 9. обнулить третий и пятый разряды регистра;
- 10. инвертировать первый и четвертый разряды регистра;
- 11. обнулить третий и четвертый разряды регистра;
- 12. инвертировать второй и седьмой разряды регистра;
- 13. обнулить третий и седьмой разряды регистра;
- 14. инвертировать второй и пятый разряды регистра;

15. обнулить четвертый и пятый разряды регистра;
16. инвертировать второй и шестой разряды регистра;
17. установить в единицу нулевой и шестой разряды регистра;
18. инвертировать первый и второй разряды регистра;
19. установить в единицу нулевой и третий разряды регистра;
20. инвертировать второй и третий разряды регистра;
21. установить в единицу третий и шестой разряды регистра;
22. инвертировать четвертый и пятый разряды регистра;
23. обнулить седьмой и пятый разряды регистра;
24. установить в единицу нулевой разряд и обнулить первый разряд регистра;
25. инвертировать первый и шестой разряды регистра;
26. инвертировать нулевой и седьмой разряды регистра;
27. обнулить второй и пятый разряды регистра;
28. установить в единицу третий разряд и обнулить седьмой разряд регистра;
29. установить в единицу шестой разряд и обнулить пятый разряд регистра;
30. инвертировать третий и четвертый разряды регистра.

Порядок работы:

1. Открыть программу *DOSBox*, в ней открыть диск *D*, найти папку *TASM*, открыть ее.

2. Создать файл с расширением *.asm*. Имя файла рекомендуется выбирать коротким, так как в дальнейшем потребуется набирать его каждый раз при вводе команд.

3. Ввести текст шаблона программы (с крайней левой позиции), можно без табуляции, на месте комментариев (начинаются с точки с запятой) написать команды в соответствии с выданным заданием:

```
.model tiny
.code
N:   push cs
     pop ds
     ; ввести команды программы
     mov ax, 4c00h
     int 21h
     .data
     ; ввести директивы распределения данных
end N
```

4. Сохранить программу.

5. Выполнить команды до получения файла *.exe* (следовать руководству по работе с *DOSBox*).

6. Если ошибок нет, то запустить эмуляцию программы и пошагово выполнить ее.

7. Подготовить отчет о проделанной работе.

Вопросы к теоретическому материалу

1. Перечислите регистры общего назначения и их названия.
2. Охарактеризуйте регистр *ax*.
3. Охарактеризуйте регистр *bx*.
4. Охарактеризуйте регистр *cx*.
5. Охарактеризуйте регистр *dx*.
6. Опишите строение регистров общего назначения.
7. Из скольки полей состоит строка кода на языке Ассемблера? Как они называются?
8. Какое поле является обязательным?
9. Какие поля являются необязательными?
10. Охарактеризуйте поле метки.
11. Охарактеризуйте поле кода операции.
12. Охарактеризуйте поле операндов.
13. Охарактеризуйте поле комментариев.
14. Что указывает при написании программы на место, где должны располагаться данные?
15. Перечислите псевдокоманды определения данных.
16. Что может быть указано в поле операндов для псевдокоманд определения данных?
17. Верна ли следующая запись: *x db 01100111*? Почему?
18. Верна ли следующая запись: *x db 01100001b*? Почему?
19. Верна ли следующая запись: *x dw 01h*? Почему?
20. Верна ли следующая запись: *x dw ABCh*? Почему?
21. Верна ли следующая запись: *x dw 0B1h*? Почему?
22. Верна ли следующая запись: *x dw ? ?*? Почему?
23. Укажите синтаксис команды пересылки данных.
24. Что может быть приемником для команды пересылки данных?
25. Что может быть источником для команды пересылки данных?

26. Назовите команду для сложения двух операндов и укажите ее синтаксис.

27. Что может быть приемником для команды сложения?

28. Что может быть источником для команды сложения?

29. Назовите команду для нахождения разности двух операндов и укажите ее синтаксис.

30. Что может быть приемником для команды нахождения разности?

31. Что может быть источником для команды нахождения разности?

32. Назовите команду для нахождения произведения двух операндов и укажите ее синтаксис.

33. Что может быть приемником для команды нахождения произведения?

34. Что может быть источником для команды нахождения произведения?

35. Назовите команду для нахождения частного двух операндов и укажите ее синтаксис.

36. Что может быть приемником для команды нахождения частного?

37. Что может быть источником для команды нахождения частного?

38. Назовите команду для полной инверсии всех разрядов приемника. Укажите ее синтаксис.

39. Назовите команду для установки в ноль отдельных разрядов приемника. Укажите ее синтаксис.

40. Назовите команду для установки в единицу отдельных разрядов приемника. Укажите ее синтаксис.

41. Назовите команду для инверсии отдельных разрядов приемника. Укажите ее синтаксис.

42. Что может быть источником для логических команд?

ПРОЦЕСС СДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

По итогам выполнения каждой лабораторной работы студент:

1. демонстрирует преподавателю правильно работающие программы;
2. демонстрирует приобретенные знания и навыки, отвечая на несколько небольших вопросов преподавателя по составленной программе, возможностям ее доработки и теме лабораторной работы в целом;
3. демонстрирует отчет по выполненной лабораторной работе.

Итоговая оценка складывается из оценок по трем указанным составляющим.

Отчет по лабораторной работе оформляется по шаблону, представленному в приложении 1. Требования к отчету представлены в приложении 2.

ПРИЛОЖЕНИЕ 1. ШАБЛОН ОТЧЕТА
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Казанский национальный исследовательский технический университет
им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации
Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №__
по дисциплине
СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Работу выполнил

Студент гр.43__

Фамилия И.О.

Принял

Преподаватель Григорьева В.В.

Казань 2024

1. Цель работы.

2. Задание на лабораторную работу – вставляется задание на лабораторную работу, соответствующее индивидуальному, выданному преподавателем, варианту студента.

3. Результат выполнения работы – формируется описание хода выполнения работы (разработанных подпрограмм, классов, переменных, структур данных и т.п.) и вставляются скриншоты с результатами работы разработанных программ (скриншоты должны быть подписаны, например, Рис. 1. Начальное состояние программы и т.п.).

4. Листинг программы – вставляется код разработанной программы с комментариями.

ПРИЛОЖЕНИЕ 2. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА

Лист документа должен иметь книжную ориентацию, поля документа должны составлять: левое – 3 см, правое – 1,5 см, верхнее – 2 см, нижнее 2 см.

Нумерация страниц – внизу страницы, по центру, особый колонтитул для первой страницы.

Междустрочный интервал – 1,5 (полуторный), отступ первой строки – 1,25.

Текст документа должен быть выполнен с использованием шрифта Times New Roman, размер – 14, выравнивание – по ширине. Заголовки выполняются тем же шрифтом, но размера 16, полужирное начертание, размещение – по центру.

Рисунки должны размещаться по центру, они нумеруются по порядку. Перед рисунком в тексте на него должна быть ссылка. Подпись рисунка должна располагаться по центру и быть выполнена шрифтом Times New Roman, размер – 12. Сначала происходит нумерация рисунка, а затем пишется его название.