**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**



ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

**Лабораторная работа №2**

по дисциплине: Архитектура вычислительных систем

тема: «Структура команд процессора»

Выполнил: ст. группы ПВ-233

Мовчан Антон Юрьевич

Проверили:

ст. пр. Осипов Олег Васильевич

Белгород 2025 г.

**Лабораторная работа №2  
Структура команд процессора  
Вариант 8**

**Цель работы:** изучить структуру команд процессора, научиться составлять

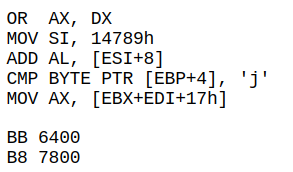
машинный код простейших команд.

**Задания для выполнения к работе:**

1. Ознакомиться с теоретическим материалом главы 2 учебника В.И. Юрова «Assembler» “Программно-аппаратная архитектура IA-32 процессоров Intel”.
2. В соответствии с вариантом задания определить по символьному описанию команд их машинный код (для 5 команд), а также по машинному коду команд определить их символьное описание (для 2 машинных кодов).

**Задание:**

Символьное описание команд на языке Assembler:



**Команда 1: OR AX, DX.**

Эта инструкция выполняет побитовую операцию OR между регистрами AX и DX. Поскольку оба регистра являются 16-битными, используется префикс 66h. Для операции OR код операции равен 000010. Устанавливаем d = 0, чтобы результат сохранялся в регистре, указанном в поле r/m. Так как обрабатываются данные размером 2 байта, w = 1. Оба операнда адресуются как регистры, поэтому mod = 11. Регистр AX имеет код 000, он помещается в поле r/m = 000. Для регистра DX код соответствует reg = 010.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Префикс | КОП | d | w | | mod | reg | r/m |
| 1100110 | 000010 | 0 | 1 | | 11 | 010 | 000 |
| 66h | 09h | | | D0h | | | |

Итоговая команда в машинном виде: 6609D0 и занимает 3 байта.

**Команда 2: MOV SI, 14789h**

Инструкция выполняет загрузку числа в регистр SI. Так как SI является 16-битным регистром, используется префикс 66h. Для MOV код операции равен 1011. Поскольку передаётся 2 байта данных, устанавливается w = 1. Регистр SI имеет код reg = 110. Значение 14789h разбивается на два байта, которые располагаются в обратном порядке: младший байт равен 89h, старший — 47h. Избыточные данные при этом отбрасываются.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Префикс | КОП | w | reg | | data | reg |
| 1100110 | 1011 | 1 | 110 | | 10001001 | 01000111 |
| 66h | BEh | | | 89h | | 47h |

Итоговая команда в машинном виде: 66BE8947h и занимает 4 байта.

**Команда 3: ADD AL, [ESI+8]**

Инструкция выполняет сложение числа из памяти по адресу [ESI+8] с регистром AL. Код операции для ADD равен 000000. Устанавливаем d = 1, чтобы результат сохранился в регистре reg. Так как обрабатывается 1 байт данных, w = 0. Для кодирования эффективного адреса используется 1 байт, поэтому mod = 01. Регистр AL имеет код reg = 000. В поле r/m указываем 110 для ESI, после чего задаётся смещение 8 (00001000).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| КОП | d | w | mod | reg | r/m | 8 |
| 000000 | 1 | 10 | 01 | 000 | 110 | 00001000 |
| 02h | | | 46h | | | 08h |

Итоговая команда в машинном виде: 024608 и занимает 3 байта.

**Команда 4: CMP BYTE PTR [EBP+4], 'j'**

Инструкция выполняет сравнение значения в ячейке памяти [EBP+4] с константой `j`. Для CMP код операции равен 10000000/111. Так как смещение занимает 1 байт, устанавливаем mod = 01. В поле r/m указываем 101 для EBP. Смещение 4 записывается как 00000100. Значение j представлено в виде 01101010 (6Ah).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| КОП | mod | КОП | r/m | 4 | ‘j’ |
| 10000000 | 01 | 111 | 101 | 00000100 | 01101010 |
| 80h | 7Dh | | | 04h | 6A |

Итоговая команда в машинном виде: 807D046A и занимает 4 байта.

**Команда 5: MOV AX, [EBX+EDI+17h]**

Инструкция выполняет загрузку числа из памяти по адресу [EBX+EDI+17h] в регистр AX. Так как AX является 16-битным регистром, используется префикс 66h. Код операции для MOV равен 100010. Поскольку передаются 2 байта данных, устанавливается w = 1. Результат сохраняется в регистре reg, поэтому d = 1. Для смещения, занимающего 1 байт, mod = 01. Регистр AX имеет код reg = 000. Поле r/m = 100 используется для кодирования эффективного адреса через SIB. В SIB: scale = 00, base = 111, index = 011. Смещение 17h записывается как 00010111.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Префикс | КОП | d | w | mod | reg | r/m | scale | index | base | 17h |
| 1100110 | 100010 | 1 | 1 | 01 | 000 | 100 | 00 | 011 | 111 | 00010111 |
| 66h | 8Bh | | | 44h | | | 1Fh | | | 17h |

Итоговая команда в машинном виде: 668B441F17 и занимает 5 байт.

**Команда 6: BB 6400**

Код операции BB соответствует инструкции MOV. Установлено w = 1, следовательно передаются данные размером 16 или 32 бита. Поле reg = 011 указывает на регистр BX/EBX. Далее идут два байта данных: 64h и 00h, что даёт число 0064h (100). Так как передаются два байта, можно сделать вывод, что используется регистр BX. Команде не хватает префикса 66h.

**MOV BX, 100**

**Команда 7: B8 7800**

Код операции B8 соответствует инструкции MOV. Установлено w = 1, следовательно передаются данные размером 16 или 32 бита. Поле reg = 000 указывает на регистр AX/EAX. Далее идут два байта данных: 78h и 00h, что даёт число 0078h (120). Поскольку передаются два байта, можно предположить использование регистра AX. Команде не хватает префикса 66h.

**Вывод:** в ходе л.р изучил структуру команд процессора, научился составлять машинный код простейших команд.