**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-223

Пахомов Владислав Андреевич

Проверили:

ст. пр. Осипов Олег Васильевич

Белгород 2024 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:

OR AX, DX

MOV SI, 14789h

ADD AL, [ESI+8]

CMP BYTE PTR [EBP+4], 'j'

MOV AX, [EBX+EDI+17h]

Машинные коды команд в 16 системе счисления:

BB 6400

B8 7800

**Команда 1: OR AX, DX.**

Команда выполняет побитовое OR над регистрами AX и DX. Т.к. AX и DX имеют размер 16 байт, команда будет содержать **префикс = 66h**. Для OR **КОП = 000010**. Можем установить байт **d = 0** чтобы результат выполнения сохранился в регистр по адресу r/m. Так как размер пересылаемых данных равен 2 байтам, **w = 1**. Оба операнда имеют регистровую адресацию, **mod = 11**.Регистру AX соответствует 000. Поместим его в **r/m = 000**. Регистру DX 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**

Команда выполняет копирование числа в регистр SI. SI имеет размер 16 байт, команда будет содержать **префикс = 66h**. Для MOV **КОП = 1011**. Так как размер пересылаемых данных равен 2 байтам, **w = 1**. Код для регистра SI **reg = 110**. Число 14789h разбивается на 2 байта, идущие в обратном порядке. Первый байт = **89h**. Второй байт = **47h**. Лишние данные отсекаются.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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]**

Команда выполняет добавление числа в регистр AL из памяти с адресом [ESI+8]. Для ADD **КОП = 000000**. Для следования операторов используем **d = 1 –** результат запишется в reg. Размер пересылаемых данных = 1 байт, следовательно ставим **w = 0**. Для кодирования эффективного адреса достаточно 1 байта, следовательно **mod = 01**. Для AL **reg = 000**. **r/m = 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'**

Команда выполняет сравнение значения из ячейки [EBP+4] с ‘j. Для CMP **КОП = 10000000/111**. **mod = 01**, так как для кодирования смещения достаточно одного байта. **r/m = 101**. **4 = 00000100**. **j = 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]**

Команда выполняет копирование числа по адресу [EBX+EDI+17h]в регистр AX. AX имеет размер 16 байт, команда будет содержать **префикс = 66h**. Для MOV **КОП = 100010**. Так как размер пересылаемых данных равен 2 байтам, **w = 1**. Результат записывается в reg, следовательно **d = 1**. **mod = 01**, так как для кодирования смещения достаточно одного байта. Код для регистра AX **reg = 000**. **r/m = 100** для кодирования эффективного адреса. SIB включает в себя: **scale = 00**, **base = 111**, **index = 011**. Смещение **17h = 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**

КОП BB соответствует команде **MOV**. **w = 1**, следовательно используется отправка 16 или 32 байтовых данных. **reg = 011**, что соответствует регистру BX/EBX. Далее идут два байта данных 64h и 00h, что соответствует числу **0064h = 100**. Так как отправляются два байта данных, можем предположить, что используется регистр BX. Команде не хватает префикса 66h.

**MOV BX, 100**

**Команда 7: B8 7800**

Коп B8 соответствует команде **MOV**. **w = 1**, следовательно используется отправка 16 или 32 байтовых данных. **reg = 000**, следовательно регистр = AX/EAX. Далее идут два байта данных 78h и 00h, что соответствует числу **0078h = 120**. Так как отправляются два байта данных, можем предположить, что используется регистр AX. Команде не хватает префикса 66h.

**MOV AX, 120**

Защита лабы:

Для add r8, qword ptr ss:[rsp+0x30] (для x32)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Префикс | КОП | d | w | mod | reg | r/m | scale | index | base | 30h |
| 01001000 | 000000 | 1 | 1 | 01 | 011 | 100 | 00 | 100 | 100 | 00110000 |
| 48h | 03h | | | 5Сh | | | 24h | | | 30h |

Для add r8, qword ptr ss:[rsp+0x30] (для x64)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Префикс | g\_reg | g\_r/m | g\_i | КОП | d | w | mod | reg | r/m | scale | index | base | 30h |
| 01001 | - | - | - | 000000 | 1 | 1 | 01 | 011 | 100 | 00 | 100 | 100 | 00110000 |
| 48h | | | | 03h | | | 5Сh | | | 24h | | | 30h |

reg – первый операнд

g\_reg – группа регистров первого операнда

r/m – второй операнд

g\_r/m – группа регистров второго операнда или base

index – индекс при вычислении эффективного адреса

g\_i – группа регистров индекса эффективного адреса

**Вывод:** в ходе лабораторной изучили работы структуру команд процессора, научились составлять машинный код простейших команд.