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)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №2

по дисциплине: Архитектура вычислительных систем

тема: «Структура команд процессора»

Выполнил: ст. группы ПВ-223

Дмитриев Андрей Александрович

Проверил:

Осипов Олег Васильевич

Белгород 2024 г.

# Содержание:

1. Название темы.
2. Цель работы.
3. Постановка задачи.
4. Вывод необходимых геометрических формул для построения изображения.
5. Реализации алгоритмов Брезенхейма для рисования отрезка и окружности.
6. Текст программы для рисования основных фигур.
7. Результат работы программы (снимки экрана).
8. Вывод о проделанной работе.

**Цель работы:** изучить структуру команд процессора, научиться составлять машинный код простейших команд.

**Задачи**:

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 варианта №2**

Символьное описание команд на языке Assembler:

MOV AX, 1001b

ADD AX, [EBX]

CMP DI, AX

SUB EAX, [ECX\*4+EAX+ 'Z']

AND WORD PTR [EDI], 12345h

Машинные коды в 16 системе счисления:

02D8

8B56 07

**Выполнение работы**

**Команда 1: MOV AX, 1001b**

Команда выполняет пересылку двоичного числа 1001 в регистр АX. Первый операнд имеет регистровую адресацию, второй является непосредственным операндом. Код операции данной команды MOV КОП=1011. Размер пересылаемых данных равен 2 байтам, значит w=1. Регистру AX соответствует поле reg=000. Число 1001b кодируется следующими двумя байтами. 1001b = 9h = 9. Байты числа представляются в памяти в обратном порядке, поэтому в коде команды первым будет младший байт 00001001b=09h, следующий – 00000000b=00h. К командам, которые оперируют данными размерам 2 байта, добавляется префикс 66h. Поля данной команды кодируются в следующей последовательности:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Префикс | КОП | w | reg | 1001b | |
|  | 1011 | 1 | 000 | 00001001 | 00000000 |
| 66h | B8h | | | 09h | 00h |

Длина команды 4 байта.

**Команда 2: ADD AX, [EBX]**

Команда выполняет сложение двойных слов из регистра AX и из памяти по адресу [EBX] и запись результата в регистр AX. Первый операнд имеет регистровую адресацию, второй – базовую со смещением и масштабированием.

Для данной команды ADD КОП=000000. d=1, т.к. данные пересылаются из поля r/m в поле reg. Поле w=1 – пересылка двойного слова. Для кодирования смещения отсутствует, поэтому mod=00. Регистру AX соответствует значение reg=000. r/m = 011, так как эффективный адрес задаётся регистром EBX. К командам, которые оперируют данными размерам 2 байта, добавляется префикс 66h. Построим машинный код:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Префикс | КОП | d | w | mod | reg | r/m |
|  | 000000 | 1 | 1 | 00 | 000 | 011 |
| 66h | 03h | | | 03h | | |

Длина команды 3 байта.

**Команда 3: CMP DI, AX**

Команда выполняет сравнение 16-битных регистров DI и AX. Код операции данной команды КОП=001110. w=1, т.к. размер операндов – два байта, d=1. Регистр DI кодируется полем reg=111, AX – полем r/m=000. Операндов в памяти нет, поэтому mod=11. К командам, которые оперируют данными размерам 2 байта, добавляется префикс 66h. Построим машинный код данной команды:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Префикс | КОП | d | w | mod | reg | r/m |
|  | 001110 | 1 | 1 | 11 | 111 | 000 |
| 66h | 3Bh | | | F8h | | |

Длина команды 3 байта.

**Команда 4: SUB EAX,** **[ECX\*4+EAX+ 'Z']**

Команда выполняет вычитание двойных слов из регистра EAX и из памяти по адресу DS: [ECX\*4+EAX+ 'Z'] и запись результата в регистр EAX. Первый операнд имеет регистровую адресацию, второй – базово-индексную со смещением и масштабированием. Для данной команды SUB КОП=001010. d=1, т.к. данные пересылаются из поля r/m в поле reg. Поле w=1 – пересылка двойного слова. Для кодирования смещения необходимо не менее 1 байта, поэтому mod=01. Регистру EAX соответствует значение reg=000. r/m = 100, так как эффективный адрес задаётся в байте SIB, который добавляется к коду команды. Поля SIB имеют значения: scale=10 (множитель 4), index=001 (EСX), base=000 (EAX). Смещение кодируется 1 байтом (‘Z’ = 90 = 5Ah).

Поля данной команды кодируются в следующей последовательности:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| КОП | d | w | mod | reg | r/m | scale | index | base | ‘Z’ |
| 001010 | 1 | 1 | 01 | 000 | 100 | 10 | 001 | 000 |  |
| 2Bh | | | 44h | | | 88h | | | 5Ah |

Длина команды 4 байта.

**Команда 5: AND WORD PTR [EDI], 12345h**

Команда выполняет логическое «И» 16-ричного числа 12345 из ячейки памяти, адрес которой содержится в регистре EDI. Размер непосредственного операнда указан явно (WORD PTR). Данной команде AND соответствует КОП=100000. Первый операнд имеет базовую адресацию, второй является непосредственным операндом. mod=00, так как поле смещения отсутствует. r/m=111 – эффективный адрес равен значению в регистре EDI. К командам, которые оперируют данными размерам 2 байта, добавляется префикс 66h. Данная команда кодируется следующим образом:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Префикс | КОП | d | w | mod | reg | r/m | 12345h |
|  | 100000 | 0 | 1 | 00 | 100 | 111 |  |
| 66h | 81h | | | 27h | | | 4523h |

Длина команды 5 байтов.

**Машинный код 1: 02D8**

Первый байт: 02h=00000010b. КОП=000000 соответствует команде ADD, у которой операнды располагаются в памяти или в регистрах. Разложим команду на части:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| КОП | d | w | mod | reg | r/m |
| 000000 | 1 | 0 | 11 | 011 | 000 |
| 02h | | | D8h | | |

mod=11, значит оба операнда имеют регистровую адресацию. d=1, значит первый операнд закодирован в поле reg, а второй – в r/m. w=0 соответствует 1 байту пересылаемых данных. reg=011 соответствует регистру BL, r/m=000 – регистру AL. Таким образом, машинный код 8BF0 соответствует команде **ADD BL, AL**.

**Машинный код 2: 8B56 07**

Первый байт: 8Bh=10001011b. КОП=100010 соответствует команде MOV, у которой операнды располагаются в памяти или в регистрах. Разложим команду на части:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| КОП | d | w | mod | reg | r/m | 7 |
| 100010 | 1 | 1 | 01 | 010 | 110 |  |
| 8Bh | | | 56h | | | 0700h |

mod=01, значит смещение задаётся одним байтом. d=1, значит первый операнд закодирован в поле reg, а второй – в r/m. w=1 соответствует 2 или 4 байтам пересылаемых данных. reg=010 соответствует регистру EDX, r/m=110 – регистру ESI. Таким образом, машинный код 8B56 07 соответствует команде MOV EDX, [ESI+7].

Защита:

mul dword ptr ds:[ecx+ebx\*4+0x02]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| КОП | d | w | mod | reg | r/m | scale | index | base | 2 |
| 111111 | 1 | 1 | 01 | 100 | 100 | 10 | 011 | 001 |  |
| F7h | | | 64h | | | 99h | | | 02h |

mul dword ptr ds:[ecx+**ecx**\*4+0x02]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| КОП | d | w | mod | reg | r/m | scale | index | base | 2 |
| 111111 | 1 | 1 | 01 | 100 | 100 | 10 | 001 | 001 |  |
| F7h | | | 64h | | | 89h | | | 02h |

mul dword ptr ds:[ecx+ebx+0x02]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| КОП | d | w | mod | reg | r/m | scale | index | base | 2 |
| 111111 | 1 | 1 | 01 | 100 | 100 | 00 | 011 | 001 |  |
| F7h | | | 64h | | | 59h | | | 02h |

mul dword ptr ds:[ecx+ebx\***2**+0x02]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| КОП | d | w | mod | reg | r/m | scale | index | base | 2 |
| 111111 | 1 | 1 | 01 | 100 | 100 | 01 | 011 | 001 |  |
| F7h | | | 64h | | | 59h | | | 02h |

mul dword ptr ds:[ecx+ebx\***8**+0x02]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| КОП | d | w | mod | reg | r/m | scale | index | base | 2 |
| 111111 | 1 | 1 | 01 | 100 | 100 | 11 | 011 | 001 |  |
| F7h | | | 64h | | | 59h | | | 02h |

mul dword ptr ds:[ecx+ebx\*4+**0x1402**]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| КОП | d | w | mod | reg | r/m | scale | index | base | 1402h |
| 111111 | 1 | 1 | 10 | 100 | 100 | 10 | 011 | 001 |  |
| F7h | | | A4h | | | 99h | | | 1402h |

mul **r**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Префикс | КОП | d | w | mod | reg | r/m |
| xxxx|xxxx | 111111 | x | x | xx | xxx | xxx |

mul **w** ptr ds:[**r**+**r**\***x+v**]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| КОП | d | w | mod | reg | r/m | scale | index | base | value |
| 111111 | x | x | xx | xxx | xxx | xx | xxx | xxx | xxxx|xxxx… |

req – используемый регистр

mod – показывает скольки байт равно число сдвига

scale – масштабный множитель

w – размерность регистра

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