

Министерство образования и науки Российской Федерации
Нижекамский химико-технологический институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования
«Казанский национальный исследовательский технологический университет»
(ФГБОУ ВО «КНИТУ»)

А.В. Долганов

ЭВМ и периферийные устройства

*МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторной работе № 6*

Нижекамск 2016

Лабораторная работа № 6

Адресация МП Intel в реальном режиме

1. ЦЕЛЬ РАБОТЫ

Целью работы является изучение адресации МП Intel в реальном режиме на примере команды mov.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ. ПРОГРАММНАЯ МОДЕЛЬ МП INTEL.

2.1. Режимы работы.

Корпорация Intel для использования в РС выпускала микропроцессоры различной разрядности. Наиболее известные из них:

- шестнадцать разрядные микропроцессоры (МП i8086),
- двадцати четырех разрядные микропроцессоры (МП i8286).
- тридцати двух разрядные микропроцессоры (от МП i8386 – до Pentium IV),
- шестидесяти четырех разрядные микропроцессоры. Но это уже принципиально новая модель архитектуры EPIC (Explicitly Parallel Instruction Computing – вычисления с явным параллелизмом команд).

Имеется еще расширение архитектуры МП, совместимых с IBM PC, до 64 бит (модели МП Hammer (K-8) корпорации AMD), но это расширение коснулось только формата данных и РОН.

Первые МП корпорация Intel выпускала, находясь в составе корпорации IBM. Программное обеспечение для этих РС производила корпорация Microsoft, также входящая в состав корпорации IBM. По этой причине, когда говорят о МП, используют термин "МП Intel архитектуры (МП IA)" или МП, совместимые с IBM PC.

В МП этих архитектур можно выделить четыре режима работы, четыре программные модели.

Реальный режим 16-ти битового микропроцессора МП i8086. Это основной режим 8-и 16-битовых первых персональных ЭВМ, программно совместимых с *IBM PC (режим МП IA-16)*.

Защищенный режим работы. Это основной 32-х битовый режим работы для 32-х битовых микропроцессоров МП IA (Intel архитектуры). В этом режиме доступны все команды и все архитектурные возможности. Используется механизм виртуальной памяти, все встроенные средства защиты, процедура переключения задач (*режим МП IA-32*).

Дополнительными режимами для них являются реальный и виртуальный 86 (режим V86).

Реальный режим 32 -х битового микропроцессора. Это 16-ти битовый режим адресации и обработки данных с прямым обращением к памяти. Режим эмулирует программную среду (модель) МП 8086 с некоторыми

дополнительными возможностями, включая новые РОНЫ, часть новых команд и расширений, как адресов, так и данных. Но в любом случае, 32-х битное смещение в сегменте не должно превышать значения 65 535 (64 Кбайт).

При запуске или перезагрузке МП устанавливается именно реальный режим. Он используется для подготовки программной среды для работы в защищенном режиме.

Виртуальный режим 86 – это разновидность формы эмуляции модели 8086. Здесь нет прямой адресации памяти. Он использует механизмы защиты и виртуальной памяти. Виртуальный режим 86 устанавливается из защищенного (для запуска программ, написанных для реального режима) с возможностью возврата в защищенный режим.

Вопросы для самопроверки:

1. Характеристики защищенного режима работы МП, определите возможности использования разрядности операндов и других средств архитектуры.

2. Характеристика реального режима работы МП, назначение режима, возможности использования разрядности операндов и других средств архитектуры.

3. Характеристика виртуального режима 86, назначение режима и возможности использования разрядности операндов и других средств архитектуры.

2.2. Программная модель 16-ти битового микропроцессора МП IA-16.

2.2.1. Модель памяти.

Одной из особенностей МП Intel – это использование сегментированной памяти. Полнокровное развитие персональных ЭВМ (РС) началась с появления 16-ти битовых микропроцессоров. Изначально микропроцессоры предназначались для управляющих систем. Но РС проектировались для широкого круга пользователей, как максимально дешевых индивидуальных (одна для одного) ЭВМ для эксплуатации без посредников (профессиональных программистов). Для этого они должны быть снабжены определенными средствами программирования на основе простейших языков программирования (типа Бейсик), управляющей (операционной) программной системой и простейшими офисными системами: текстовые редакторы, электронные таблицы и т.д.

Но 16-ти битовая адресация позволяла адресовать только $2^{16} = 65\,536$ байт (64 Кбайт). Этого было достаточно для управляющих ЭВМ, но не для РС.

Вероятно, по этой причине была выбрана модель структурированной математической памяти (Рис. 2.1.), состоящая из **16** сегментов по **65 536 байт (64 Кбайт)**

Этот прием позволил увеличить адресное пространство математической памяти в реальном режиме до $2^{20} = 1\,048\,576$ (1 Мбайт).

При этом первые 10 сегментов определялись, как память пользователей, а остальные 6 сегментов – как системные (для операционной системы).

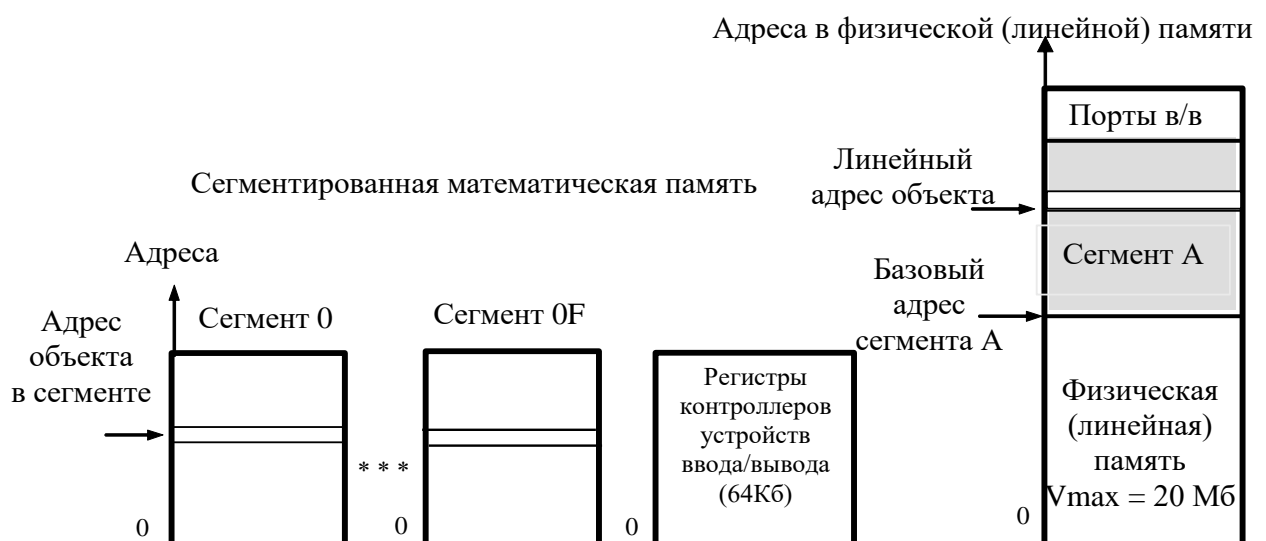


Рис. 2.1. Сегментирование математической памяти и адресация физической памяти

Кроме этого, вводился еще один дополнительный сегмент (64Кбайт) для адресации портов. Порты – это программно-доступные (адресуемые) регистры контроллеров устройств ввода-вывода. Но это отдельное (дополнительное) адресное пространство процессора.

Биты в памяти, как и в регистрах, нумеруются с младших разрядов. Многобайтные операнды адресуются по номеру младшего байта. Размещение программных объектов без соблюдения правил целочисленных границ не рекомендуются

Использование сегментированной математической памяти для расширения адресного пространства имеет и отрицательные моменты. Это несоответствие адресных структур математической и физической (линейной) памяти. **При каждом обращении к физической памяти необходима процедура преобразование адреса обращения из адреса сегментированной математической памяти в адрес линейной физической памяти. Это процедура трансляции сегмента.**

Процедура трансляции сегментов заключается в вычислении адреса линейной, плоской физической оперативной памяти по адресу сегмента и смещению в сегментах.

Вопросы для самопроверки:

1. Понятие сегментированной памяти.
2. Размер сегмента в МП IA-16.
3. Максимальное число сегментов в МП IA-16.
4. Процедура трансляции сегментов в МП IA-16.
5. Место хранения базовых адресов сегментов.

2.2.2. Порты ввода/вывода.

МП позволяет адресовать порты в адресном пространстве портов и ячейки оперативной памяти в отдельном адресном пространстве оперативной

памяти.

Для адресации портов используются команды ввода: вывода: IN, INS, OUT, OUTS (пересылки между РОНами и портами), а для адресации ячеек памяти – любые адресные команды.

Для доступа к порту ввода/вывода, расположенному в адресном пространстве памяти могут использоваться любые работающие с памятью команды. Это дает некоторую гибкость работы с портами. Например, команда MOV позволяет пересылать данные между портом и любым регистром. Команды AND, OR, и TEST могут манипулировать отдельными битами управляющих регистров портов при программировании процедур ввода/вывода. Сегменты логической памяти, порты ввода/вывода и их отображение (возможное размещение) в памяти представлены на рис. 2.1.

Если при отображении портов на адресное пространство памяти используется сегментация, то поля AVL в дескрипторах сегментов позволяют отметить сегменты как непереключаемые и не подлежащие подкачке.

Все сегменты (включая сегмента портов) в плоской памяти задаются двумя независимыми параметрами: сегментом (базовый адрес сегмента) и адресом объекта в сегменте (смещением в сегменте).

Адресное пространство портов:

- $2^{16} = 65\,536$ однобайтных портов с нумерацией от 0 до 65 535 или
- $2^{15} = 32\,768$ двухбайтных портов с нумерацией от 0 до 65 534 или
- $2^{14} = 16\,384$ четырехбайтовых портов с нумерацией от 0 до 65 532.

В командах ввода вывода адреса портов от 1 до 255 могут задаваться прямыми адресами:

- IN al, 24 – ввод из 24-го порта байта в регистр al,
- IN ax, 24 – ввод из 24-го порта слова в регистр ax,
- IN eax, 24 – ввод из 24-го порта двойного слова в регистр eax,

Задание в командах портов с адресами больших 255 производится с использованием регистра dx:

- MOV dx, 1024 ;засылка адреса порта в dx
- IN al, dx ;ввод из 1024-го порта байта в регистр al,
- IN ax, 24 ;ввод из 1024-го порта слова в регистр ax,

Вопросы для самопроверки:

1. Понятие портов,
2. Методы адресации портов,
3. Адресация портов в МП IA,
4. Команды обращения к портам в МП IA-16.

2.2.3. Регистровый файл.

Регистровый файл МП IA –16 содержит 14 пользовательских регистров:

- регистры общего назначения (РОНы) – 8 регистров,
- сегментные регистры – 4 регистров,
- указатель команд – 1 регистр,

- регистр состояния (флагов) – 1 регистр.

Все эти регистры в той или иной степени доступны программам пользователей.

Регистры общего назначения.

РОНы МП Intel (Рис.2.2) не являются классическими регистрами общего назначения. Они могут задаваться и явными адресами и по умолчанию (кодом операции, постбайтом и т.д.).

С точки зрения использования регистров по умолчанию, они делятся на две группы:

- регистры данных – ax, cx, dx и bx (по возрастанию адресов),
- регистры адресов – sp, bp, si и di (по возрастанию адресов)

Регистр bx, хотя и относят к группе регистров данных, но он же является и одним из регистров адреса базы данных (для сегмента DS).

Регистр dx может использоваться в качестве адресного регистра портов в командах ввода/вывода.

Регистры cx, si и di по умолчанию используются при обработке строк.

При работе с 16-ти битовыми операндами и компонентами адреса используются 16- битовые РОНы – ax, cx, dx, bx, sp, bp, si и di (по возрастанию адресов).

	15	8	7	0	
ax	ah		al		Акумулятор
cx	ch		cl		Счетчик
dx	dh		dl		Данные
bx	bh		bl		База (для DS)
	sp				Указатель стека
	bp				База кадра стека SS)
	si				Индекс источника
	di				Индекс приемника

Рис. 2.2. РОНы

При работе с байтовыми операндами используются 8 – битовые РОНы: al, cl, dl, bl, ah, ch, dh и bh (по возрастанию адресов).

Сегментные регистры и трансляция сегментов.

При использовании сегментированной математической памяти необходима процедура преобразования адреса сегментированной (математической памяти) в адреса линейной физической памяти при каждом обращении к памяти.

Адрес объекта в сегментированной памяти задается двумя компонентами базовым адресом сегмента и смещением в сегменте. В команде задается только смещение в сегменте. Базовый адрес сегмента задается содержимым одного из четырех сегментных регистров (рис. 2.3).

МП IA-16 содержали 4 сегментных регистров по 16 бит:

- **ES** – дополнительного сегмента данных,
- **CS** – сегмента кода,
- **SS** – сегмента стека,
- **DS** – сегмента данных,

№	19	Базовые адреса сегментов	4	3	0
00	ES – Доп. сегмент данных				0000
01	CS – Сегмент кода				0000
10	SS – Сегмент стека				0000
11	DS – Сегмент данных				0000

Рис.2.3. Сегментные регистры IA-16.

Использование регистров **CS**, **SS**, **DS** и **ES** возможно по умолчанию и с использованием префикса замены сегмента. Базовые адреса сегментов используются для преобразования адреса сегментированной памяти в линейную.

Емкость физической памяти в МП IA-16 позволяет разместить одновременно до 16 полных сегмента по 64 Кбайт (2^{16} байт). Но одновременно, без изменения содержимого сегментных регистров, в реальном режиме IA-16 можно обращаться только к четырем, по числу сегментных регистров.

Переход на использование остальных сегментов (кроме кодовых), размещенных в оперативной памяти можно производить:

- при помощи загрузки базовых адресов в соответствующие сегментные регистры командами пересылки (**mov**),
- командами загрузки сегментных регистров (**LES**, **LDS** и **LSS**).

Все операции работы со стеком ориентированы на словарную организацию стека. По этой причине адрес указателя стека и базовый адрес сегментного регистра **SS** должны быть **четными**.

Переходы на новые кодовые сегменты производятся командами межсегментных передач управления.

Преобразование адреса сегментированной памяти в адрес линейной (плоской) памяти производится **процедурой трансляции сегмента: сложением базового адреса сегмента с адресом операнда в сегменте**.

Но сегментные регистры - 16-и битные, а базовые адреса должны быть 20 - битными. Поэтому при использовании сегментных регистров их содержимое умножается на 16, добавлением четырех нулей после младшего разряда

Таким образом, адрес в сегменте математической памяти МП IA-16 задается тремя компонентами: **B** (база), **X** (индекс) и **disp** (смещение).

Для вычисления адреса обращения к физической памяти (при сегментированной структуре математической памяти) требуются, по крайней мере, две процедуры:

- **Вычисления по компонентам (B), (X) и disp прямого адреса в сегменте.**

• **Преобразования адреса в сегменте сегментированной памяти в адрес линейной физической памяти.**

Результатом первого преобразования адреса является получение одной компоненты логического адреса – **смещения в сегменте**, результатом второго преобразования – получения **смещения в линейной памяти**.

Для уменьшения неопределенности при использовании вышеописанных смещений используют термины:

- **disp** – для обозначения смещения в кадре сегмента,
- **offset** – для обозначения смещения в сегменте,
- **линейный адрес** – для обозначения адреса (смещения) в линейной (не сегментированной) памяти.

Схема вычисления смещения в сегменте (**offset**) по компонентам адреса, включая смещение в кадре сегмента (**disp**), и трансляции сегмента (вычисление линейного адреса в несегментированной памяти) представлена на рис.2.4.

Трансляция сегментов связана с проблемой с (20-й адресной шиной) оперативной памяти. Эта проблема заключается в следующем.

Максимальный адрес сегмента, который может быть записан в сегментный регистр (с учетом четырех нулей в младших разрядах) составляет число:

$A_{\max} = 2^{20} - 2^4 - 1$. При использовании этого адреса, как базового адреса сегмента, то в оставшиеся ячейки памяти можно будет записать только 16 слов. В МП с 20-й битовой адресной шиной, остальные данные сегмента запишутся в ячейки оперативной памяти, начиная с нулевого адреса.

В МП IA-32 при работе в реальном режиме (эмуляции МП IA-16) приходится учитывать это явление, используя прием (заворота) 20-й шины адреса.

В защищенном режиме, если включен механизм виртуальной памяти, происходит еще одно преобразование адреса – трансляции страниц.

В (МП IA-16) – трансляция страниц (виртуальная память) не реализована.

Указатель команд IP.

Это 16-ти битный регистр – счетчик, вычисляющий адрес следующей команды (в кодовом сегменте) по числу байт в текущей команде. Программе указатель команд доступен только частично по командам передачи управления. В начальное состояние **IP** устанавливается автоматически в процедуре включения процессора. Прямой доступ к **IP** по записи или чтению не предусмотрен (но возможен доступ к содержимому **IP** после его копирования в стек).

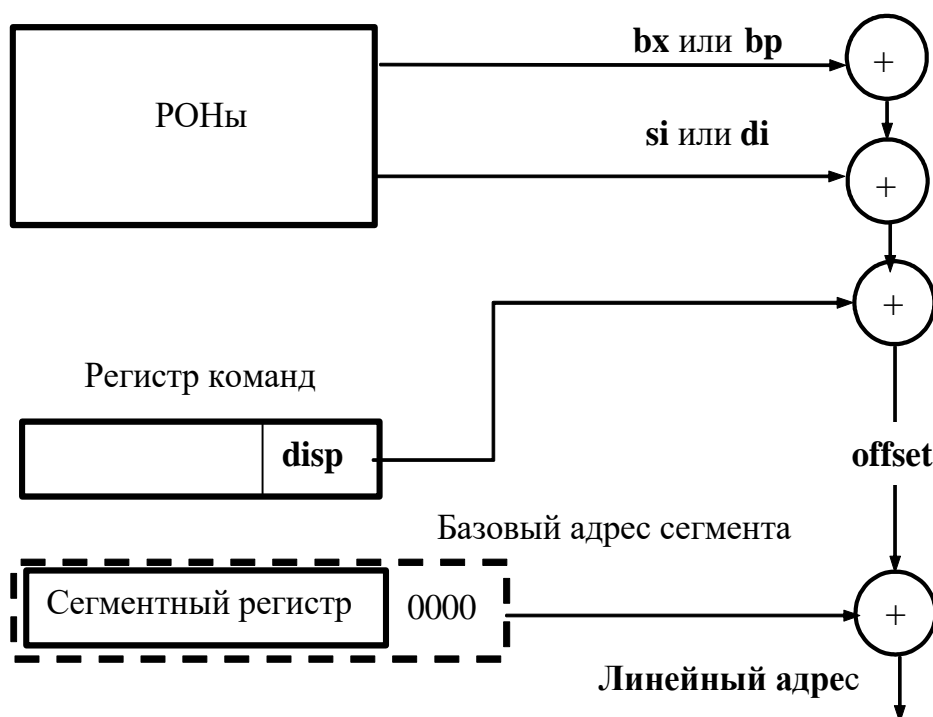


Рис.2.4. Схема вычисления смещения в сегменте (offset) и трансляции сегмента (вычисление линейного адреса.)

Регистр состояния (флагов).

Это 16-ти битный регистр, в котором сгруппированы управляющие сигналы режимов работы основных систем процессора и флаги кодов условий для условных команд передач управления.

Флаги кодов условий передач управления:

- CF – арифметический перенос в старшие разряды,
- PF – четность (наличие четности кода результата),
- AF – дополнительный (десятичный) перенос из младшей тетрады,
- ZF – нулевой результат,
- SF – знак результата (для дополнительного кода),
- OF – переполнение,
- TF – трассировка (прерывание после выполнения каждой команды для работы в режиме отладки программы),
- IF – разрешение прерывания,
- DF – направление обработки строк.

Вопросы для самопроверки:

1. Основные группы пользовательских регистров,
2. Группа регистров данных в составе POH МП IA-16?
3. Группа регистров адресов в составе POH МП IA-16
4. Предназначение регистра IP в МП IA-16,
5. Предназначение регистра F (флагов) в МП IA-16,
6. Основные биты кода условий передачи управления в регистра F (флагов),
7. Назначение сегментных регистров в МП IA-16,

8. Число сегментных регистров в МП IA-16,
9. Спецификации сегментных регистров в МП IA-16,
10. Команды загрузки сегментных регистров в МП IA-16,
11. Причины использования задания адреса в сегменте тремя компонентам,
12. Компоненты адреса, используемые при задании адресов объектов в сегменте памяти,
13. Компонента *disp* при задании адреса в МП IA,
14. Компонента *offset* при задании адреса в МП IA,
15. Термин "линейный адрес" при задании адреса в МП IA,
16. Процедура трансляции сегмента при обращении к памяти в МП IA-16.

2.2.4. Структура команд.

Имеется две формы представление машинных команд:

- на языке программирования машинного уровня (ассемблере),
- на аппаратном уровне с использованием битовых кодов.

Перекодировку команд производит ассемблер совместно с рядом программ операционной системы.

Команды на языке ассемблера. На языке ассемблера команды содержат символный код операции и один или два адреса. Команды могут быть одноадресными или двухадресными. Первый адрес в команде – адрес приемника, второй – источника. Оба адреса могут быть адресами РОНов, Адресом оперативной памяти может быть только один.

Исключением является:

- команда **MOVS** – команда строковой передачи данных память – память (адреса ячеек памяти задаются содержимым пар регистров базы/индекс)
- команды работы со стеком, которые перемещают данные из памяти в стек, который также находится в памяти.

Адрес оперативной памяти может задаваться прямым адресом (*offset*) или компонентами (*D*, *X*, *disp*).

Примеры команд на уровне ассемблера:

1. **mov ax, bx** ;копирование в **ax** содержимого **bx**,
2. **mov ax, offset 356** ;копирование в **ax** содержимого ячейки памяти по адресу 356,
3. **mov ax, [CS:bx +356]** ;копирование в **ax** содержимого ячейки памяти в сегменте **CS** по адресу, равному сумме содержимого регистра **bp** и смещения – 356,
4. **mov ax, [bp +si+356]** ;копирование в **ax** содержимого ячейки памяти по адресу, равному сумме содержимого регистров **bp**, **si** и смещения 356,
5. **mov byte ptr z, 15** ; копирование в **ax** содержимого ячейки памяти по адресу **z** (конструкция **byte ptr z** уточняет, что адрес **z** является адресом одного байта, а не слова).
- 6.

Машинная кодировка команд IA – 16.

МП IA-16 используют команды переменного формата от 1 до 7 байт. Кодировка команд производилась с учетом частоты использования команды в типовых программах.

Команда может иметь (рис. 2.5.):

- возможно, префикс (1 или 2 однобайтных префиксов),
- код адресации (1 байт),
- возможно, постбайт (1 байт),
- возможно, байты смещения – disp (1 или 2 байта),
- возможно, байты непосредственного операнда (1 или 2 байта).

0, 1 или 2 байта	1 байт	0,1 байт	0,1 или 2 байта	0,1 или 2 байта
префикс	код операции	постбайт	байты смещения	непосредствен ный операнд

Рис.2.5. Структура команд МП IA -16.

Элементы команды:

Префикс.

Это необязательный байт, модифицирующий процедуру выполнения команды.

МП IA – 16 предусматривает использование двух префиксов:

- повторения,
- замены сегмента памяти, используемого по умолчанию (только для данной команды).

Префикс повторения. В МП IA – 16 для обработки строк данных используются обычные команды, но с использованием префикса повторения. Префикс повторения обеспечивает повторные действия над элементами строк данных. Предусмотрены два варианта префикса повторения, отличающиеся заданием конца повторений: количеством повторений:

1. подсчетом заданного числа повторения (в регистре cx),
2. кодом условия результата очередного повторения операции в регистре флагов.

Префикс замены сегмента памяти.

В МП IA-16 предусмотрены 4 сегментных регистра. Использование любого из них может быть задано по умолчанию или явно, кроме сегмента команд.

По умолчанию используются сегменты:

- **CS** – для выборки команд по **IP** (счетчику команд),
- **DS** – для выборки данных, адреса которых используют базу – **bx**,
- **SS** – для выборки данных, адреса которых используют базу – **bp**,
- **ES** – в командах обработки строк.

Изменения использования сегмента по умолчанию можно указав в команде явно используемый сегмент:

1. **mov ax, [CS:bx +356]** база **bx** указывает на использование сегмента данных **DS**, но запись **CS:bx** явно указывает на работу с данными, помещенными в кодовый сегмент.

2. **mov ax, ma [bx +356]** в этом варианте используется константа, зарезервированная псевдооператорами ассемблера **DB** (определить байт) или **DW** (определить слово):

Segment CS задание сегмента,

DW ma 45 зарезервировать двухбайтную ячейку для данных **ma** с начальным значением 45.

Код операции.

Это единственный из обязательных элементов команды (1 байт), задающий операцию. Возможны операции, заданные одним кодом операции. Это одноадресные команды, операнд для которых задается или выставляется по умолчанию. Поле кода операции может содержать собственно код операции и от одного до 3 специальных битов:

- **w** – размера операнда (7-й бит), при **w = 0**, размер операнда – байт, при **w = 1**, размер операнда – слово.
- направления передачи результата для двухместной команды (**d** – 6-й бит);
 1. при **d = 1** – адресом приемника является РОН, заданный полем **reg** постбайт,
 2. при **d = 0** – адресом приемника является адрес памяти;
 3. в одноместной команде используется только источник и 6-й бит байта кода операции (**s=1**) определяет автоматическое расширение байта данных до размера слова.

Возможен однобайтовой вариант одноместной команды, в котором адрес данных (номер РОНа) размещается в трех младших разрядах кода операции.

Постбайт.

Это необязательный элемент команды. Задаёт один (для одноадресной команды) или два адреса операндов, один из которых используется и как адрес результата.

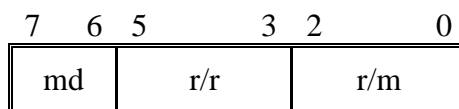


Рис. 2.6. Структура постбайта

Постбайт (рис. 2.6) содержит 3 поля:

- **md** – поле режима адресации,
- **r/r** – поле адреса регистра,
- **r/m** – поле адреса второго регистра или кода алгоритма вычисления адреса памяти.

Поле md. Это поле режима адресации.

Задаёт тип команды:

- регистр/регистр (**r/r**);
- регистр/память (**r/m**).

Для команд регистр/память задает процедуру вычисления адреса операнда в памяти:

- $md = 00$ – режим адресации **r/m** без использования смещения,
- $md = 01$ – режим адресации **r/m** с использованием смещения 1 байт),
- $md = 10$ – режим адресации **r/m** с использованием смещения 2 байт),
- $md = 11$ – режим адресации **r/r**.

Задании $md \neq 11$ определяет команду типа регистр память. В этом случае поле **r/m** содержит номер РОНа, в котором находится один из операндов, а поле **r/m**, – номер алгоритма вычисления адреса памяти, по которому хранится второго операнда.

Значения поля **r/m** от 000 до 011, задают формулы вычисления адреса памяти второго операнда по 3 компонентам: **B**, **X** и **disp**.

Значения поля **r/m** от 100 до 111, задают формулы вычисления адреса памяти второго операнда по двум компонентам: **B** и **disp**.

Количество выбираемых смещений **disp**, расположенных после постбайта, определяется по полю постбайта – **md**.

Исключение: В адресации операндов, в качестве исключения, имеется возможность задавать адрес операнда в памяти прямым 16-ти разрядным адресом. Признаком прямой адресации является комбинация значений **md = 00** и **r/m=110** (в таблице 1 выделен жирным шрифтом).

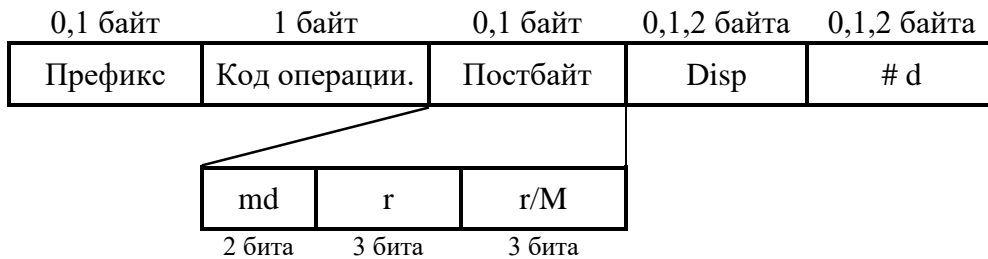
Количество выбираемых смещений **disp**, расположенных после постбайта, определяется по полю постбайта – **md**.

В таблице 1. представлены все формулы вычисления адресов РОН и ячеек памяти в зависимости от значений полей постбайта: **md**, **r/r**, **r/m**.

На рис. 2.7 представлена структура команд МП Intel в реальном режиме и схемы вычисления адресов операндов.

Таблица 1. Постбайтовые режимы адресации МП IA-16					
Поле r/m	Поле md				
	00	01	10	11	
				w = 0	w = 1
000	DS:[bx + si]	DS:[bx + si + disp (1 байт)]	DS:[bx + si + disp (2 байт)]	al	ax
001	DS:[bx + di]	DS:[bx + di + disp (1 байт)]	DS:[bx + di + disp (2 байт)]	cl	cx
010	SS:[bp + si]	SS:[bp + si + disp (1 байт)]	SS:[bp + si + disp (2 байт)]	dl	dx
011	SS:[bp + di]	SS:[bp + di + disp (1 байт)]	SS:[bp + di + disp (2 байт)]	bl	bx
100	DS:[si]	DS:[si + disp (1 байт)]	DS:[si + disp (2 байт)]	ah	si
101	DS:[di]	DS [di + disp (1 байт)]	DS [di + disp (2 байт)]	ch	di
110	DS:[disp -2 ба-та]	SS:[bp + disp (1 байт)]	SS:[bp + disp (2 байт)]	dh	bp
111	DS:[bx]	DS [bx + disp (1 байт)]	DS [bx + disp (2 байт)]	bh	bx

Структура команд МП Intel в реальном режиме.



если $r/M \leq 11$, то – команда типа r/M; базовая индексная адресация; 1-й операнд находится в регистре, указанном полем r, адрес M **вычисляется как: (B)+(X)+ #d16**, где B и X задаются кодом поля r/M.

при: 00 } Disp – отсутствует
 01 } Disp = 1 байт
 10 } Disp = 2 байта

При 11 Команда типа r/r, операнды находятся в регистрах, указанными полями r и r/M.

Вариант относительно-индексной адресации памяти

если $r/M \geq 100$, то команда типа r/M, первый операнд находится в регистре, указанном полем r, адрес M **вычисляется как: (B)+ Disp**, где: B – задается кодом поля r/M постбайта.

при 00 } Disp – отсутствует
 01 } Disp = 1 байт
 10 } Disp = 2 байта

Исключение: при md=00 и r/M = 110, адрес задается прямым 16-ти битным смещением (#d16).

Выбор сегмента: при указании в качестве компоненты адреса – (bp), используется сегмент стека (SS), иначе – сегмент данных (DS)

Рис.2.7. Структура команд МП Intel в реальном режиме и схемы вычисления адресов операндов.

Вопросы для самопроверки:

1. Структура команд в МП IA-16,
2. Назначение префикса в командах МП IA-16,
3. Назначение постбайта в командах МП IA-16,
4. Поля постбайта в командах МП IA-16,
5. Поля постбайта в командах МП IA-16,
6. Тип сегмента используемого по умолчанию при выборке команд в МП IA-16,
7. Тип сегмента используемого по умолчанию при выборке операнда в МП IA-16,
8. Тип сегмента используемого по умолчанию при работе со стеком в МП IA-16,
9. Тип сегмента используемого по умолчанию при работе с командами обработки строк в МП IA-16,
10. Способы замены сегментов, используемых по умолчанию.
11. Назначение постбайта в МП IA-16,

12. Основные поля постбайта в МП IA-16,
13. Назначение поля md постбайта в МП IA-16,
14. Назначение поля r/r постбайта в МП IA-16.

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Лабораторная работа расположена по следующему пути на сервере кафедры:

Z:\Документация\По предметно\Организация ЭВМ и систем (ОЭВМиС)\Лабораторные занятия\Лабораторная работа _1\ADR_PC_WIN

2. Скопируйте папку **ADR_PC_WIN** на свой компьютер.
3. Внутри папки запустите файл **main.exe**.
4. В открытом окне (см. рис.3.1) нажмите на кнопку «**Практика**».

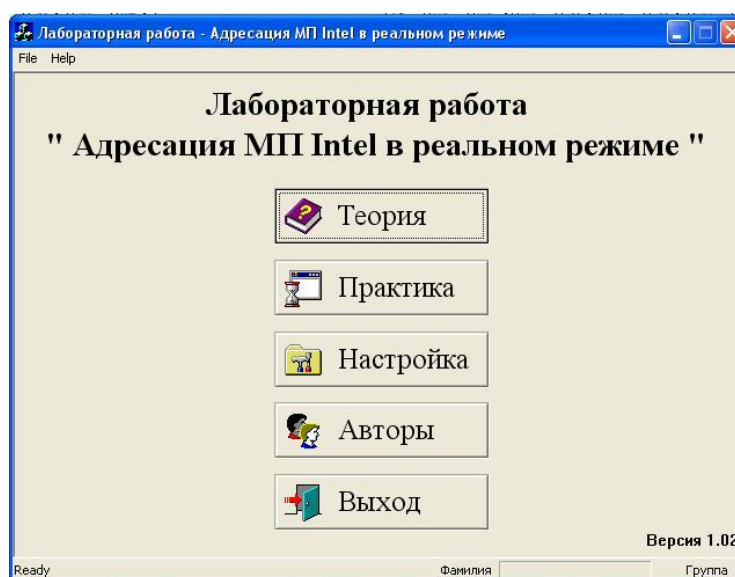


Рис. 3.1. Общий вид программы.

5. В окне «**Регистрация**» введите свою **фамилию** и **группу** (см. рис. 3.2.).

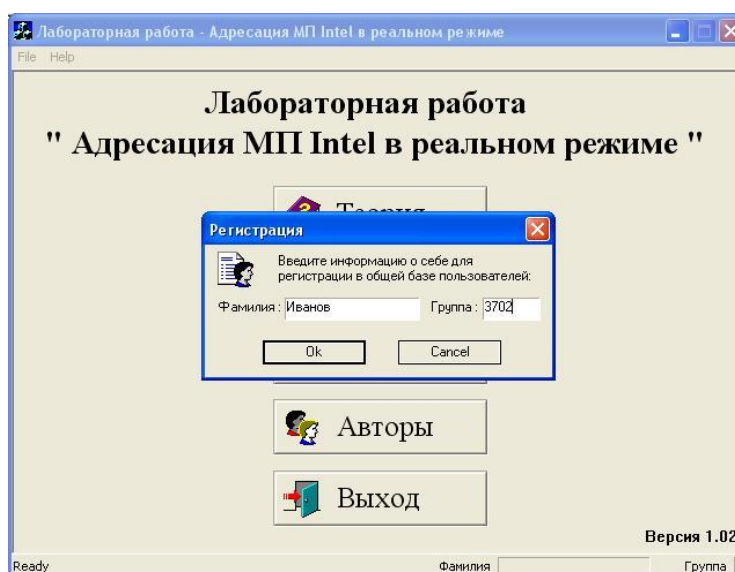


Рис. 3.2. Окно регистрации студента.

6. Выберите тип адресации команды, указанной в поле «команда» и нажмите кнопку «Выбрать» (см. рис.3.3).

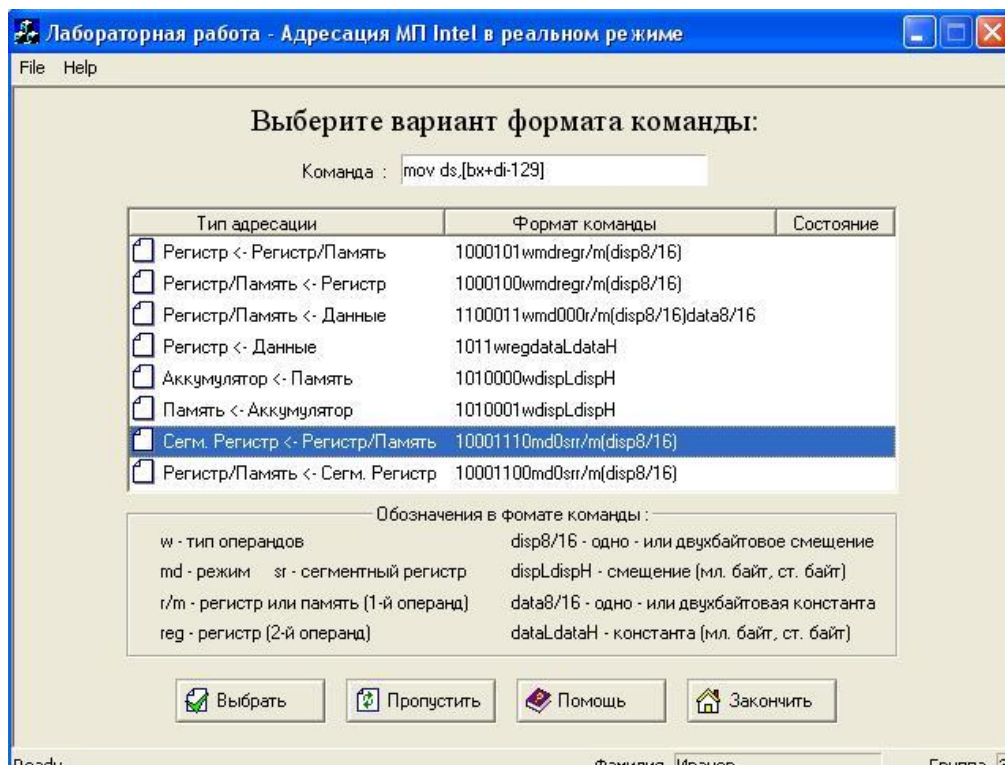


Рис. 3.3. Окно выбора формата команды.

7. Формат команды необходимо перевести в машинный код, выбирая в соответствующих полях окна двоичные расшифровки частей команды (см. рис.3.4.).

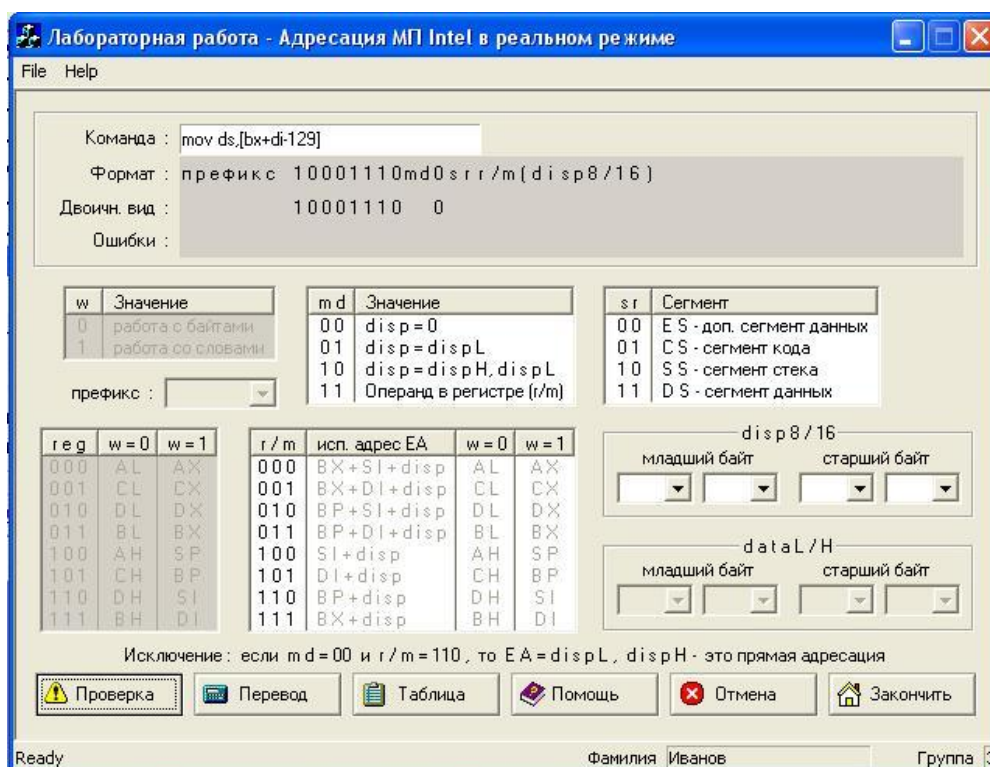


Рис. 3.4. Окно преобразования команды в машинный код.

8. В случае правильного выбора машинного кода команды появится следующее окно (см. рис. 3.5), выполненное вами задание отметится галочкой, в поле состояния пропишется «зачтено» (см. рис. 3.6).

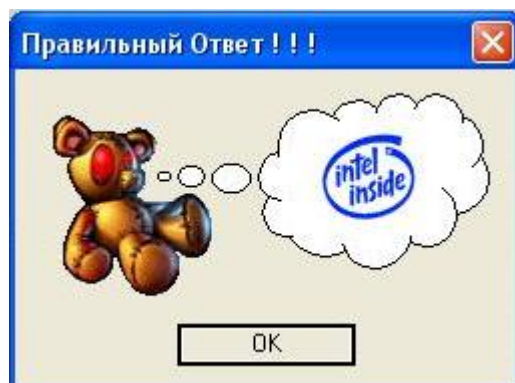


Рис. 3.5. Окно правильного выполнения задания.

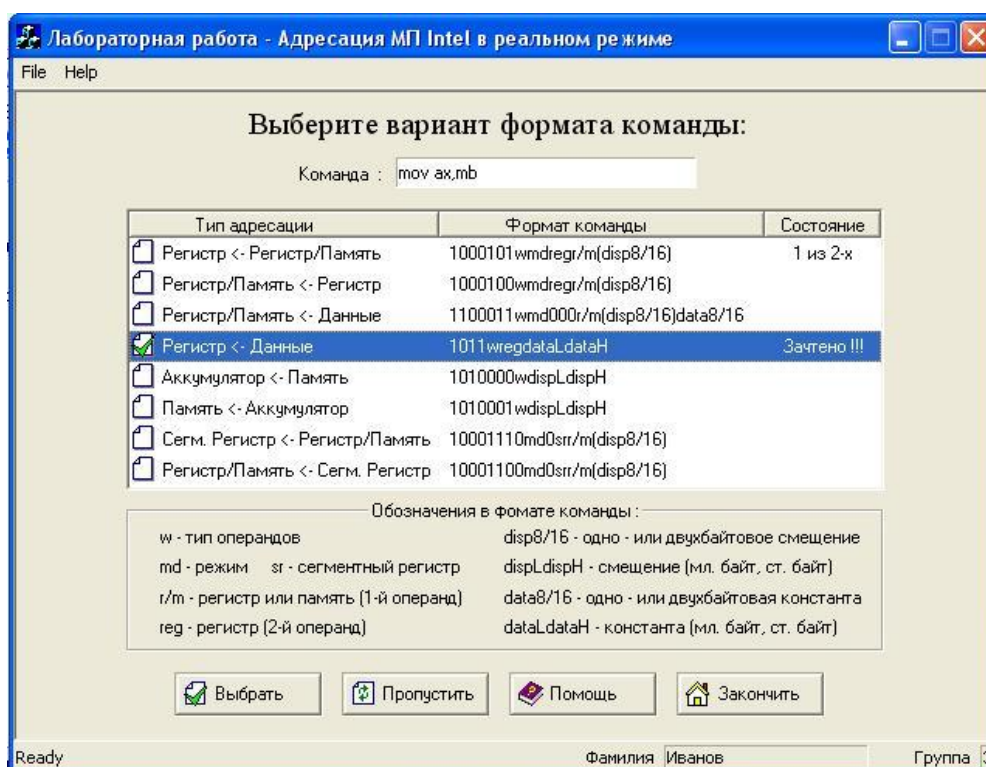


Рис. 3.6. Окно выбора формата команды после правильного преобразования команды в машинный код.

9. Необходимо выполнить все задания.

4. СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен включать:

а) окно окончания лабораторной работы, которое необходимо показать преподавателю (см. рис. 3.7.);

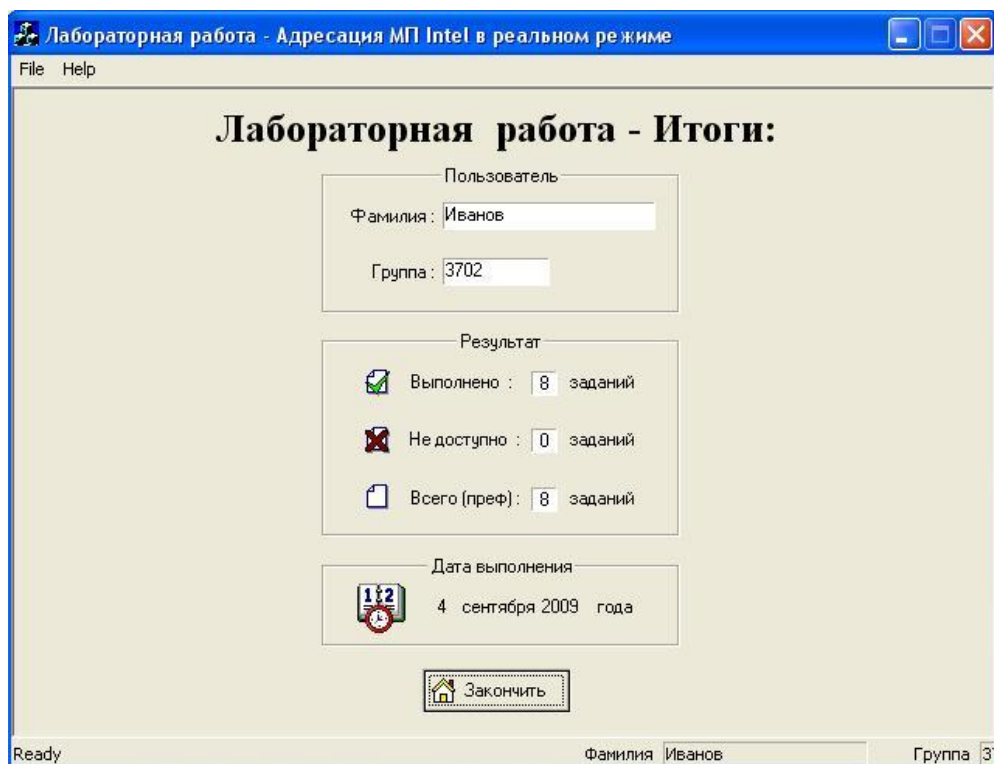


Рис. 3.7. Окно о завершении работы студентом.

б) ответить на контрольные вопросы

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

- а) вопросы в конце каждого пункта теоретического материала (вопросы для самопроверки),
- б) вопросы преподавателя.