

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 № 7*

Нижекамск 2016

Лабораторная работа № 7

Система адресации СМ ЭВМ

1. ЦЕЛЬ РАБОТЫ

Целью работы является изучение системы адресации СМ ЭВМ на примере модели процессора семейства PDP-11.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ. ПРОГРАММНАЯ МОДЕЛЬ ПРОЦЕССОРОВ СЕМЕЙСТВА PDP-11 (КОРПОРАЦИИ DEC)

2.1. Отображение оперативной памяти.

Семейства ЭВМ – это первый ряд 16-тибитных программно совместимых моделей ЭВМ от микро до супер-мини. Для многих последующих высокопроизводительных ЭВМ архитектура PDP-11 стала стартовой. Корпорация DEC, развивая эту архитектуру, стала лидирующей на рынке мощных 64-битных рабочих станций на базе VAX-кластеров и моделей МП серии Alpha.

Максимальный размер математической памяти моделей ЭВМ PDP-11 – $2^{16} = 65\,536$ байт (64 Кбайт) или $2^{15} = 32\,768$ двухбайтных слов (32 К). Но максимальный размер физической памяти в четыре раза больше и может достигать до – $2^{18} = 262\,144$ байта (256 Кбайт) или 131 072 слова (256 К).

Адресация бит в байтах и словах – обратная (с младших разрядов), слова и двойные слова адресуются по номеру байта в слове (двойном слове).

Структура и адресация бит в байте и слов в физической памяти представлена на рис. 2.1.

Слова				Байты				
номера байт	15	8	7	0	номера байт	7	0	номера байт
1	старший		младший		0	младший		0
3	старший		младший		2	старший		1
5	старший		младший		4	* * *		
7	старший		младший		6	* * *		
* * *			* * *			* * *		
65 533	старший		младший		65 532	* * *		
65 535	старший		младший		65 534	* * *		
						* * *		
						* * *		
						младший		65 534
						старший		65 535

Рис. 2.1. Структура и адресация бит в байте и слов в памяти

Трансляция математических адресов производится с помощью диспетчера памяти (вариант реализации виртуальной памяти). Превышение

объема физической памяти над математической памятью позволяет реализовывать многопрограммные режимы работы.

Особенностью адресации памяти является то, что в пространство адресов математической и физической памяти отображаются не только адреса ячеек памяти, но и все пространство программно доступных регистров устройств ввода/вывода (номера портов).

Для адресов портов предусмотрено 4 Кбайт старших адресов пространства памяти.

2.2. Регистровый файл.

В регистровый файл входят:

- регистры общего назначения (РОНы),
- регистр слова состояния процессора.

Регистры общего назначения (РОНы). ЭТО – 8 шестнадцатибитных регистра, шесть из них используются только при явной адресации, два (R6 и R7) специализированы и кроме явной адресации могут использоваться по умолчанию.

R6 – шестнадцатибитный регистр указателя аппаратного стека SP (Stack Pointer). Используется аппаратурой процессора при прерываниях и передаче управления для сохранения адреса и состояния программы.

R7 – шестнадцатибитный счетчик команд PC (Program Counter). Вычисляет адрес следующей команды. Доступен программам пользователя по записи и чтению.

Регистр слова состояния процессора.

Шестнадцатибитный регистр слова состояния процессора PS (Processor Status) (рис. 2.2) содержит поля:

- Текущего и предыдущего режима работы (биты с 12 по 15). Поля задают режимы: пользователь/система. Используются механизмом защиты программ. В режиме «пользователь» запрещено использование ряда "привилегированных" команд.

- Приоритет процессора (биты с пятого по седьмой). Задают приоритет процессора. Процессор может иметь приоритет от 0 до 7. Приоритет процессора используется в арбитраже системного интерфейса как уровень маскирования запросов прерывания от внешних устройств.

- Бит слежения T (Tpar, четвертый бит). При включенном бите T – после выполнении каждой команды (кроме RТТ), производится процедура прерывания.

- Четыре бита кода условий (биты с нулевого по третий):

1. C – "перенос",
2. V – "переполнение",
3. Z – "результат нулевой",
4. N – "результат отрицательный".

Коды условий используются при выполнении команд ветвления.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Текущий режим			Послед. режим						Приоритет процессора		T	N	Z	V	C

Рис. 2.2. Структура регистра слова состояния процессора

2.3. Структура данных.

Данные представляются в двух видах: данные со знаком и данные без знака. Данные со знаком могут быть в форме с фиксированной запятой или в форме с плавающей запятой. Данные без знака – это адреса или алфавитно-цифровые символы.

Данные со знаком с фиксированной запятой – это целые числа в виде байтов, слов или двойных слов. Старший бит числа – знаковый. Отрицательные числа представлены в дополнительном коде.

Данные со знаком с плавающей запятой обрабатывается специальным сопроцессором обработки чисел с плавающей запятой (ППЗ). В ППЗ определены два формата:

- одинарной точности – двойное слово: знак мантиссы (1бит), сдвинутый порядок (1 байт), мантисса (3 байта),
- двойной точности – учетверенное слово: знак мантиссы (1бит), сдвинутый порядок (1 байт), мантисса (7 байт).

В обоих случаях, мантисса – это нормализованная дробь, в которой первая цифра после запятой должна быть единицей. При хранении в памяти первая цифра мантиссы является скрытым битом (всегда равным единице), не хранимой в памяти. Поэтому, в двойное или учетверенное слова памяти помещаются: знак, байт сдвинутого порядка и, соответственно, два или семь байт мантиссы. Отрицательные значения мантиссы заданы в прямом коде.

2.4. Структура команд.

Существенной новацией структуры команд ЭВМ PDP-11 является всесторонне развитая система адресации и использование стека при обращении к подпрограммам и при обслуживании прерываний. Эти механизмы стали нормой для многих последующих ЭВМ различных корпораций.

По количеству адресов операндов ЭВМ PDP-11 использует безадресные, одноадресные и двухадресные команды. Основной (полный) формат имеют двухадресные команды (рис.2.3). Они содержат поля:

- кода операции,
- адреса операнда источника,
- адреса операнда приемника.

15	12	11	9	8	6	5	3	2	0
Код операции	адрес источника				адрес приемника				
	режим		РОН		режим		РОН		

Рис.2.3. Структура двухадресной команды

Код операции. Код операции задает операцию, структуру команды и размер операнда.

Двухадресные команды – это, в основном, команды работы с битами. Исключения составляют команды пересылки и две команды арифметических операций (сложения и вычитания слов в форме фиксированной запятой).

Из возможных 16 кодов четырехбитового кода операции для кодирования операции используются только четыре. Остальные коды используются для обозначения других дополнительных структур команд. Эти дополнительные форматы команд представлены на рис. 2.4.

Все рассмотренные команды являются шестнадцатитрибитными, т.е. занимают одно слово, но в зависимости от типов используемых адресаций они могут включать в себя дополнительные поля длиной одно или два слова. Второе и третье слова команды это или адреса или непосредственно заданные значения операндов.

15	12	11	9	8	6	5	3	2	0	
Код операции				R – источник		адрес приемника				a
Код операции				R – источник		смещение				b
Код операции						адрес приемника				c
Код операции								R – источник		d
Код операции										e
Код операции					Смещение					f

Рис.2.4. Дополнительные структуры команд

Адресация операндов. В двухадресных командах первый адрес после кода операции является адресом источника, второй адрес является адресом приемника, т.е. адресом по которому выбирается второй операнд и в который записывается результат операции.

Для задания полного адреса, в команде предусмотрены два смежных трехбитных поля. Первое поле – это поле режима адресации, второе поле – поле РОНа, используемого для задания компоненты адреса.

Для обозначения содержимого полей в архитектуре PDP-11 используется **восьмеричная система счисления**.

Адресные поля команд всех структур могут быть расширены одним или двумя дополнительными словами в зависимости от используемых режимов адресации.

Структуры команд "a" и "b" – это тоже структуры двухадресных команд, но в качестве одного из операндов используется содержимое РОНа. В команде структуры «b» второй адрес задается смещением относительно счетчика команд. Это структура команды выхода из цикла. Содержимое указанного в команде РОНа автоматически уменьшается на размер выполненной команды. Пока содержимое РОНа не равно нулю, происходит передача управление на

команду по адресу равному содержимому счетчика команд плюс удвоенное смещение (в словах), которое рассматривается как число со знаком.

Команды со структурой "с" – это одноадресные команды, операнд адресуется и в памяти и в РОНах.

Команды со структурой "d" – это одноадресные команды. Адрес задается РОНом. Это могут быть команды обработки данных из стека, в этих случаях – указанный в команде РОН является указателем стека.

Команды структуры "е" – это безадресные операции.

Команды структуры "f" – это команды условной передачи управления, условие передачи управления задается кодом операции, а адрес перехода задается удвоенным смещением относительно счетчика команд. Смещение рассматривается как число со знаком.

Набор команд.

Все возможные коды операций двухадресной команды представлены в Табл. 1.

Таблица 1. Коды операций двухместных команд (команды с суффиксом В – байтовые).				
№	Код	№	Код	Описание
0000	–	1000		В двухместных полно-адресных командах не используется
0001	MOVB	1001	MOV	Пересылка содержимого источника в приемник
0010	CMPB	1010	CMR	Сравнение источника с приемником, установка кода условий
0011	BITB	1011	BIT	Тестирование битов (по маске*), установка кода условий
0100	BICB	1100	BIC	Сброс битов (по маске*).
0101	BISB	1101	BIS	Установка битов (по маске*)
0110	ADD	1110	SUB	Сложение – 0110 или вычитание 1110 с записью результата
0111	–	1111		В двухместных полно-адресных командах не используется
*Примечание: маска задается вторым адресом				

Кроме команд сложения и вычитания, все коды в этом формате по отношению к размерам операндов симметричны. На каждую операцию предусмотрено два кода операции: один для обработки полных операндов (слов), другой – для обработки байт. На языке ассемблера коды обработки байт содержат суффикс – В. Пример:

- MOV A₁ A₂ ;переслать слово,
- MOVB A₁ A₂ ;переслать байт,

где: A₁ A₂ – адреса памяти или РОН, заданные в команде.

2.5. Режимы адресации

Режимы адресации задаются трехразрядным полем режима адресации. Если команда содержит два адреса, то режимы адресации для каждого операнда задаются независимо друг от друга, каждый своим полем режима.

Трехразрядное поле режима адресации позволяет задать 8 основных режимов адресации. Режим адресации – это правило вычисления адреса

операнда по заданному РОНу и, возможно, одному из дополнительных слов команды. Использование дополнительных слов команды также определяется режимом адресации. Но не все РОНЫ в МП ЭВМ PDP-11 равнозначны. Регистр R7 – это счетчик команд. Использование этого регистра для задания режима адресации или бессмысленно или, для четырех режимов, дает эффекты новых режимов.

Таким образом, программист, меняя режимы и используя как обычные РОНЫ, так и счетчик команд, может задавать операнды в командах двенадцатью режимами адресации. Основные режимы адресаций представлены в (табл.2).

Таблица 2 Основные режимы адресаций			
№	Мнем.	Название	Описание
0	R	Регистровая	В регистре – операнд
1	(R)	Регистровая косвенная-	В регистре – адрес операнда
2	(R)+	Автоувеличение	В регистре – адрес операнда. После использования, адрес увеличивается на длину
3	@(R)+	Автоувеличение косвенная	В регистре – адрес адреса операнда. После использования, адрес увеличивается на 2.
4	– (R)	Автоуменьшение	Содержимое регистра уменьшается на длину операнда, результат используется как адрес
5	@ – (R)	Автоуменьшение косвенная	Содержимое регистра уменьшается на 2, результат используется как адрес адреса
6	X(R)	Индексная	Сумма содержимого регистра и индекса используется как адрес операнда, в качестве индекса используется второе или третье слово команды
7	@X(R)	Индексная косвенная	Сумма содержимого регистра и индекса используется как адрес адреса операнда, в качестве индекса используется второе или третье слово команды

Примеры (упорядочены по номерам режимов адресации):

0. CLR R0 очистка регистра R0.
1. CLR (R0) очистка ячейки памяти, адрес которой находится в R0
2.
 - a) CLR (R0)+ очистка ячейки памяти, адрес которой находится в R0, после использования адреса он увеличивается на 2; в цикле: последовательная очистка ячеек памяти по смежным адресам в направлении их увеличения;
 - b) mov R1, (R0)+ пересылка из вершины стека числа в R1, где R0 – указатель стека.
3. CLR &(R0)+ в цикле: последовательная очистка ячеек памяти по списку адресов.
- 4.

а) CLR $-(R0)$ очистка ячейки памяти, адрес которой находится в R0, после использования адреса он увеличивается на 2; в цикле: последовательная очистка ячеек памяти по смежным адресам но в направлении их уменьшения.

б) mov $-(R0), R1$ загрузка в стек числа из R1, где R0 – указатель стека.

5. CLR $\&-(R0)$ в цикле: последовательная очистка ячеек памяти по обратному списку адресов.

6. CLR X(R0) очистка ячейки памяти, расположенной по адресу $X+(R0)$, X–число во втором слове команды.

7. CLR $\&X(R0)$ очистка ячейки памяти, адрес которой расположен по адресу $X+(R0)$, X–число во втором слове команды.

Режимы адресаций с использования счетчика команд представлены в табл. 3.

Таблица 3. Режимы адресаций с использования счетчика команд				
№	Мнемоника		Название	Описание
2	(PC)+	#A	Непосредственный операнд	Непосредственный операнд задан в команде вторым или третьим словом
3	@(PC)+	@#A	Абсолютный адрес	Абсолютный адрес операнда задан в команде вторым или третьим словом
6	A(PC)	A	Относительный адрес	Адрес операнда, относительно счетчика команд задан в команде вторым или третьим словом
7	@A(PC)	@A	Косвенный относительный адрес,	Адрес адреса операнда, относительно счетчика команд задан в команде вторым или третьим словом.

Примеры использования режимов адресации с использованием счетчика команд. Примеры упорядочены по номерам режимов адресации:

Режим автоувеличения (2).

С использованием обычного РОНа

$\beta+0$) MOV (R0)+, R1 пересылка содержимого ячейки памяти, адрес которой хранится в R0 (например 1000), в регистр R1: (1000)→ R1; здесь $\beta+0$ –адрес выполняемой команды (содержимое счетчика команд – R7), после исполнения команды он увеличивается и становится равным $\beta+2$; по этому адресу должна находиться следующая команда (X),

$\beta+2$) X ;

С использованием счетчика команд:

$\beta+0$) MOV (R7)+, R1 та же команда, но в качестве регистра используется счетчик команд (R7); после выборки команды он "продвигается" до ($\beta+2$); команда предполагает пересылку в R1 содержимого

ячейки памяти, адрес которой хранится в данном случае в счетчике команд (R7), следовательно, в R1 пересылается X. Но, так как режим адресации с автоувеличением, то содержимое (R7), уже как РОН увеличивается на 2 и становится равным $\beta+4$,

$\beta+2$) X–данные

$\beta+4$) следующая команда.

Таким образом, при использовании в качестве РОНа счетчика команд, команда увеличивается на два байта, содержимое которых используется в качестве непосредственного операнда. На ассемблере для команд с такой адресацией предусмотрена упрощенная форма:

MOV #X, R1, где #X – непосредственный операнд.

Косвенный режим автоувеличения с использованием счетчика команд (3).

Косвенный режим автоувеличения отличается от прямого режима тем, что операнд интерпретируется как адрес операнда. Следовательно, команда:

$\beta+0$) MOV (R7)+, R1

$\beta+2$) X

удлинится на слово X, где X используется в качестве абсолютного адреса операнда. Для команд с абсолютной адресацией в ассемблере предусмотрена упрощенная форма записи:

MOV @#X, R1, где X – абсолютный адрес операнда.

Индексный режим с использованием счетчика команд (6).

$\beta+0$) MOV X(R7)+, R1

$\beta+4$) Следующая команда.

Здесь команда содержит два слова. В первом слове записана собственно команда, а во втором – значение величины X.

Компиляция и выполнения этой команды имеет ряд особенностей:

- В процедуре вычисления адреса операнда величина X складывается с содержимым счетчика команд. Это – адресация не относительно фиксированной базы, а относительно счетчика команд. Это очень ценная адресация, которая позволяет проектировать "перемещаемые программы", т.е. непривязанные к конкретным адресам памяти.

- В ассемблере для этой адресации предусмотрена упрощенная запись: MOV X, R1, где X – относительный адрес.

- Программист не следит за адресами программного кода и в командах всегда указывает абсолютные адреса операндов (согласно их распределения в своей "математической" памяти).

- Номера портов устройств ввода/вывода отображены на пространство адресов памяти, но имеют постоянные адреса в старшей половине памяти и требуют только абсолютную адресацию.

В связи с указанными особенностями, программисты, при написании программы на ассемблере, все адреса пишут в абсолютных значениях. Но, при

использовании адресов портов, они заказывают абсолютную адресацию (@#A, режим 3 с R7), а при использовании адресов данных – относительную (A, режим 6 с R7). При этом компилятор оставляет указанный адрес без изменения в первом случае и вычисляет относительный адрес данных, как абсолютный минус продвинутое значение счетчика команд (увеличенного на 4 для первого адреса и на 6 – для второго), во втором случае.

Косвенный индексный режим с использованием счетчика команд (8).

$\beta+0$) MOV @X(R7)+, R1

$\beta+4$) Следующая команда.

В командах с косвенным индексным режимом адресации с использованием счетчика команд величина X задает не адрес операнда, а адрес адреса операнда, относительно "продвинутого" счетчика команд.

В ассемблере для этой адресации предусмотрена упрощенная запись: MOV @X, R1, где @X – относительный адрес адреса операнда.

Вопросы для самопроверки:

1. *Максимальный размер математической памяти в PDP-11.*
2. *Максимальный размер физической памяти в PDP-11.*
3. *Особенности использования РОН.*
4. *Биты кодов условия регистра состояния процессора.*
5. *Количество разрядов кодирования полного адреса данных в команде PDP-11.*
6. *Основные поля в команде PDP-11, задающие адреса операндов.*
7. *Основные режимы адресации в PDP-11.*
8. *Режимы адресации с использованием счетчика команд в PDP-11.*
9. *Способ задания непосредственного операнда в PDP-11.*
10. *Способ задания абсолютного адреса в PDP-11.*
11. *Способ задания относительного адреса в PDP-11.*
12. *Способ задания косвенного относительного адреса в PDP-11.*

3. ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ ДЛЯ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

3.1. Введение

В СМ ЭВМ принята 8-ричная система счисления. Адрес слова – это адрес его младшего байта. Адресовать можно слова и байты. Адрес слова – всегда четный.

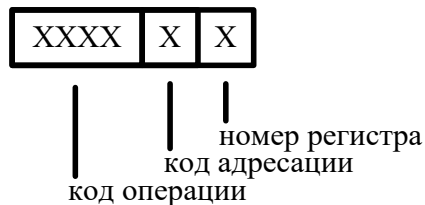
В формировании адреса обязательно принимают участие регистры. Всего регистров 7. Они нумеруются от 0 до 7. Регистр 7 является счетчиком команд. Адрес операнда указывается сочетанием двух чисел. Первое указывает на способ вычисления исполнительного адреса и называется кодом адресации. Всего кодов адресации 7. Второе указывает номер регистра, который принимает участие в вычислении адреса.

В СМ имеется 12 видов адресации. Минимальная длина машинной команды – слово. При использовании некоторых видов адресации длина машинной команды может увеличиваться до 3 слов.

Команда в СМ может быть одноадресной и двухадресной.

Одноадресная команда:

Формат команды:



Пример:

CLR R1 – команда на ассемблере

005001 – машинный код, где

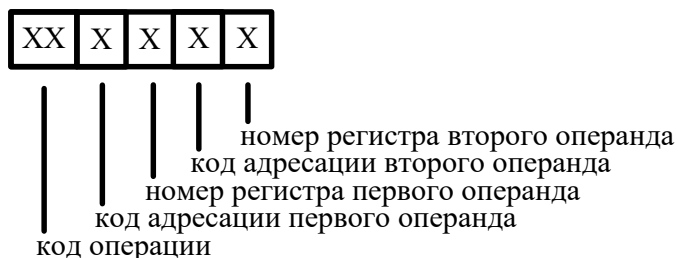
0050 – код операции,

0 – код адресации,

1 – номер регистра.

Двухадресная команда:

Формат команды:



Первый операнд является источником, а второй операнд – приемником информации.

Пример:

MOV @R0,R5 – команда на ассемблере,

011005 – машинный код, где

01 – код операции,

1 – код адресации первого операнда,

0 – номер регистра первого операнда,

0 – код адресации второго операнда,

5 – номер регистра второго операнда.

3.2. Список используемых инструкций

Мнемоника	Восьмичный код	Название инструкции	Пояснение
CLR	0050	очистка слова	0 заносится в слово по адресу, определяемому операндом
CLRB	1050	очистка байта	0 заносится в байт по адресу, определяемому операндом
INC	0052	увеличение слова	Значение, находящееся по заданному адресу, увеличивается на 1
INCB	1052	увеличение байта	
DEC	0053	уменьшение слова	Значение, находящееся по заданному адресу, уменьшается на 1
DECB	1053	уменьшение байта	
MOV	01	пересылка слова	Содержимое адреса, определенного первым операндом, пересылается по адресу, заданному вторым операндом. Содержимое адреса, заданного первым операндом не меняется
MOVB	11	пересылка байта	
ADD	06	сложение	Содержимое первого операнда складывается с содержимым второго операнда и результат пересылается по адресу второго операнда
SUB	16	вычитание	Содержимое адреса, указанное в первом операнде, вычитается из содержимого адреса, заданного вторым операндом, и результат пересылается по адресу второго операнда
CMP	02	сравнение слов	Содержимое адреса, определенного первым операндом, сравнивается с содержимым адреса, заданного вторым операндом. В результате вырабатываются коды условий, которые анализируются командами переходов
CMPB	12	сравнение байт	

3.3. ВИДЫ АДРЕСАЦИИ СМ ЭВМ

3.3.1. ПРЯМАЯ АДРЕСАЦИЯ

1). Прямая адресация через регистр

A: RN код адресации: **0**

Операнд задается непосредственно в регистре, определяемом выражением R.

Пример:

CLR R0 ; очищение регистра 0.

2). Прямая адресация с автоувеличением

A: (RN)+ код адресации: **2**

Регистр R содержит адрес операнда и после выполнения инструкции (за исключением особых случаев) увеличивается на 2 для операций над словами и на 1 для операций над байтами.

Пример:

CLR (R0)+ ; очищение слова по адресу, содержащемуся в регистре 0 и увеличение содержимого регистра 0 на 2.

3). Прямая адресация с автоуменьшением

A: -(RN) код адресации: 4

Перед выполнением операции регистр R, содержащий адрес операнда, уменьшается на 2 для операций над словами и на 1 для операций над байтами.

Пример:

CLR -(R0) ; содержимое регистра 0 уменьшается на 2, указывая на адреса очищаемых слов.

4). Индексная

A: E(RN) код адресации: 6

Значение выражения E хранится в памяти, как второе или третье слово инструкции. Действительный адрес вычисляется как сумма величины E и содержимого регистра R. Величина E есть смещение, содержимое регистра R – база.

Пример:

CLRB -2(R0) ; действительный адрес очищаемого байта равен 2 плюс содержимое регистра 0.

3.3.2. КОСВЕННАЯ АДРЕСАЦИЯ

1). Косвенная адресация через регистр

A: @RN или (RN) код адресации: 1

Регистр R содержит адрес операнда.

Пример:

CLR @R0 ; очищение слово по адресу, находящемуся в регистре 0

2). Косвенная адресация с автоувеличением

A: @(RN)+ код адресации: 3

Регистр R содержит указатель (адрес) адреса операнда. После выполнения инструкции содержимое регистра R увеличивается на 2.

Пример:

CLR @(R0)+ ; содержимое регистра 0 указывает на ячейку памяти, содержащую адрес очищаемого слова. После выполнения операции содержимое регистра 0 увеличивается на 2

3). Косвенная адресация с автоуменьшением

A: @-(RN) код адресации: 5

Содержимое регистра R уменьшается на 2 и рассматривается как адрес адреса операнда.

Пример:

CLR @-(R0) ; содержимое регистра 0 уменьшается на 2, указывая на ячейку памяти, содержащую адрес очищаемого слова.

4). Косвенная индексная

A: @E(RN) код адресации: 7

Выражение E плюс содержимое регистра R дает указатель адреса операнда, как и при индексной. Величина E есть смещение, содержимое регистра R – база.

Пример:

CLR @114(R0) ; если регистр 0 содержит 100, то указатель получает значение 214, и если ячейка памяти 214 содержит 2000, то очищается содержимое ячейки 2000.

3.3.3. АДРЕСАЦИЯ С ИСПОЛЬЗОВАНИЕМ РС (счетчика команд)

1). Непосредственная

A: #E код адресации: 2

Этот режим адресации обеспечивает работу с операндами, заданными непосредственно в инструкции. При непосредственном режиме адресации операнд E запоминается во втором или третьем слове инструкции. Этот режим реализуется как автоувеличение содержимого регистра 7.

Пример:

MOV #100,R0 ; переслать 100 в регистр 0.

2). Абсолютная

A: @#E код адресации: 3

В этом режиме E указывает абсолютный адрес, который хранится во втором или третьем слове инструкции. Значение слова, следующего непосредственно за инструкцией, воспринимается как абсолютный адрес операнда.

Пример:

MOV @#100,R0 ; переслать значение содержимого ячейки 100 в регистр 0.

Абсолютная адресация интерпретируется как косвенное автоувеличение регистра 7. В приведенном примере инструкция MOV, помещенная в памяти, имеет вид:

ячейка 20: 013700 ; первое слово инструкции

ячейка 22: 000100 ; второе слово инструкции

ячейка 24: следующая инструкция

Инструкция MOV хранится по адресу 20, т.е. PC=20. После чтения инструкции MOV PC становится равным 22. Код адресации источника равен 3, номер регистра источника равен 7, т.е. это косвенная адресация с автоувеличением, где роль регистра исполняет счетчик команд (PC). То есть содержимое PC (в нашем примере 22) рассматривается как адрес адреса операнда, иными словами, ячейка 22 содержит абсолютный адрес операнда.

После выполнения команды PC увеличивается на 2 и указывает на следующую инструкцию.

3). Относительная

A: E код адресации: 6

Этот вид адресации интерпретируется как индексный режим с использованием регистра 7(PC) в качестве индексного регистра. Смещение для вычисления адреса хранится во втором или третьем слове инструкции и

задается как число, которое прибавляется к содержимому РС и дает адрес операнда. Таким образом, смещение есть (Е - РС), где Е- адрес операнда.

Пример:

Инструкция MOV 100,R3, размещенная в памяти по абсолютному адресу 20 имеет вид:

ячейка 20: 016703 ;первое слово инструкции

ячейка 22: 000054 ;второе слово инструкции

ячейка 24: следующая инструкция

Инструкция MOV хранится по адресу 20, т.е. РС=20. После чтения инструкции MOV РС становится равным 22. Код адресации источника равен 6, номер регистр источника равен 7, т.е. это адресация с индексацией, где роль регистра исполняет счетчик команд (РС). Значение смещения процессор считывает по адресу, указанному в РС (в нашем примере содержимое РС = 22), после чего значение РС становится равным 24. Для вычисления адреса операнда-источника значение смещения прибавляется к содержимому указанного регистра, т.е: адрес операнда вычисляется как смещение + РС = 54+24=100

Пример:

CLR 100 ; очистить ячейку 100

4). Относительно-косвенная

А: @Е код адресации: 7

Этот вид адресации подобен относительной адресации, за исключением того, что выражение Е используется в качестве указателя адреса операнда. То есть, операнд, следующий за инструкцией, прибавляется к содержимому РС. Эта сумма дает указатель адреса операнда.

Пример:

MOV @100,R0 ; переслать в регистр 0 содержимое ячейки, адрес которой находится в ячейке 100

4. 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 №2\ADR_CM](#)

2. Скопируйте папку ADR_CM на свой компьютер.

3. Внутри папки запустите файл CM_CONTR.exe.

4. В открытом окне (см. рис.4.1) введите свою **фамилию и группу**.

5. Выберите режим адресации команды (при первом обращении по умолчанию выбирается режим адресации: прямая через регистр, см. рис. 4.2.).

6. В открытом окне (см. рис. 4.3) в поле "**Команда**" прописана команда на языке ассемблера CM ЭВМ. "**Машинный код**" – это поле, где вы должны набрать машинный код этой команды. В окошках "**Регистры**" и "**Память**" необходимо ввести значения, которые получают соответствующие регистры или ячейки памяти после выполнения команды.

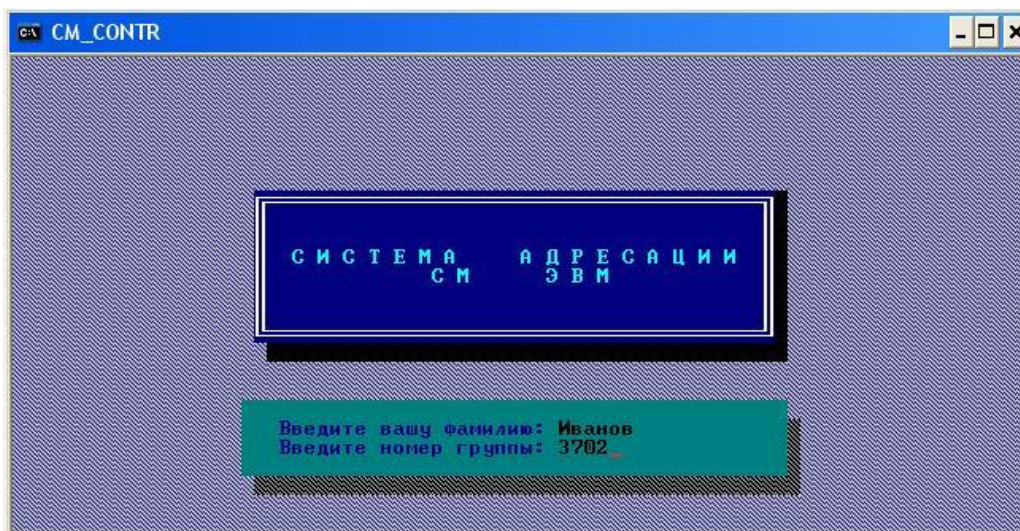


Рис. 4.1. Окно регистрации студента.

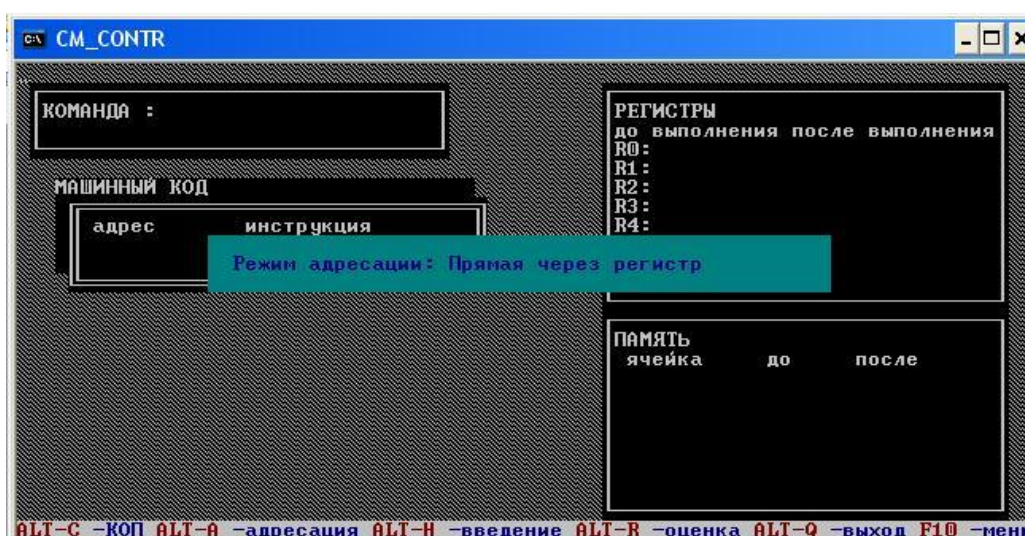


Рис. 4.2. Режим адресации: режим адресации.

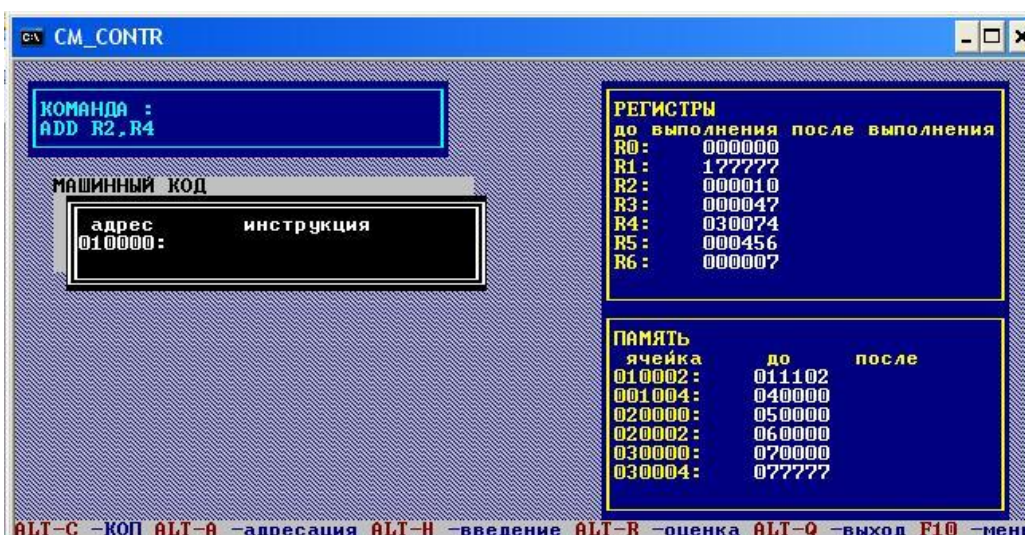


Рис. 4.3. Окно выполнения задания

Следующая команда предлагается только после получения правильного машинного кода. Если вы не можете правильно определить машинный код предложенной команды, у вас есть возможность сменить задание. Для этого

необходимо вызвать основное меню нажатием клавиши **F10** и выбрать пункт **"Выбор примера"**. В этом пункте меню у вас есть две возможности: выбор новой команды в рамках текущей темы или выбор новой темы (см. рис .4.4).

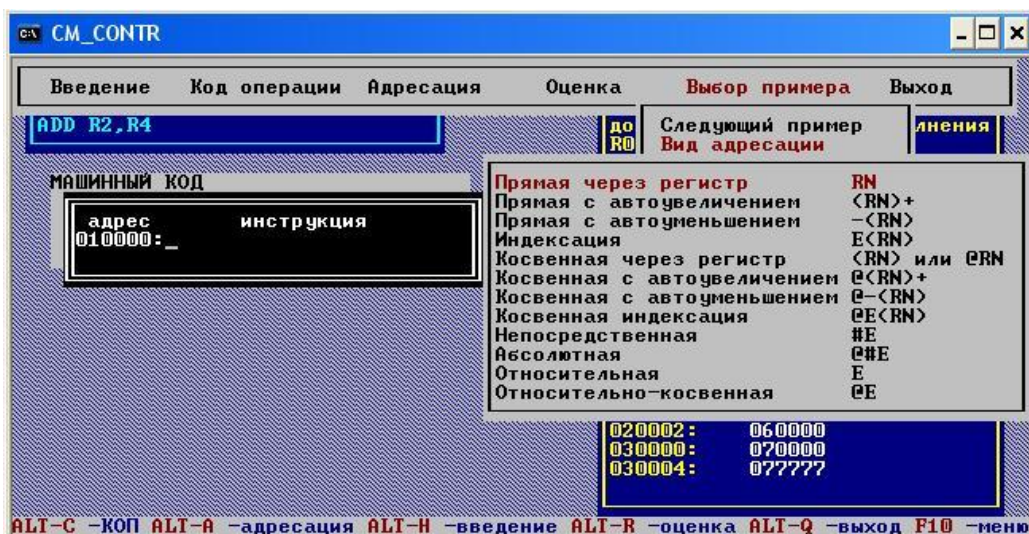


Рис. 4.4. Окно выполнения задания

Если вы выбираете новую команду, старое задание заносится в список ваших "долгов". "Долги" – это задания, которые вы не выполнили по данной теме. Перед окончанием работы по текущей теме вы должны выполнить все "долги". Если вы выполняете все задания без ошибок, тема засчитывается ("+"), иначе – не засчитывается ("-"). Если вы выбираете новую тему, старая тема не засчитывается (см. рис. 4.5.).

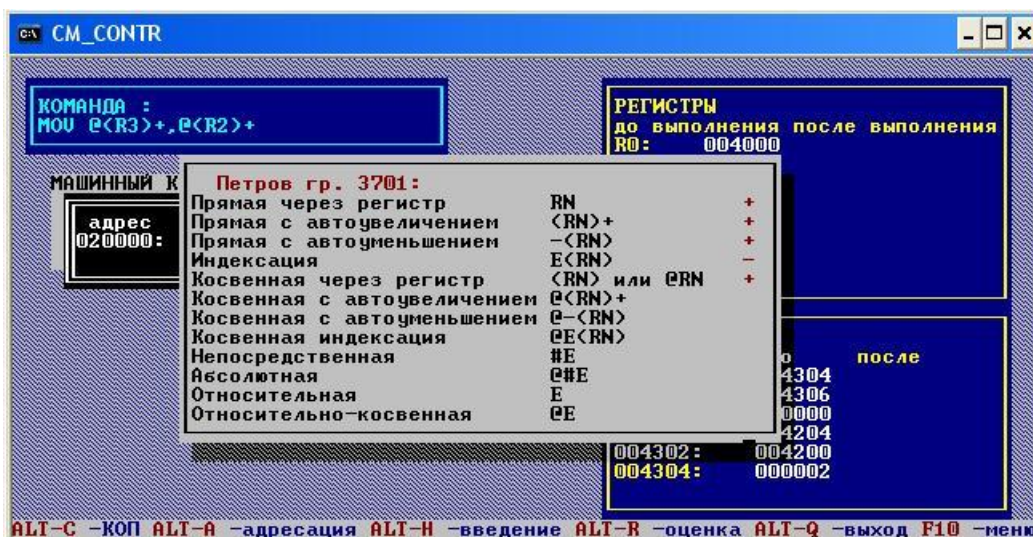


Рис. 4.5. Окно выполненных (усвоенных) студентом заданий

7. В случае правильного выполнения задания появится следующее окно (см. рис. 4.6).
8. Необходимо выполнить все задания.

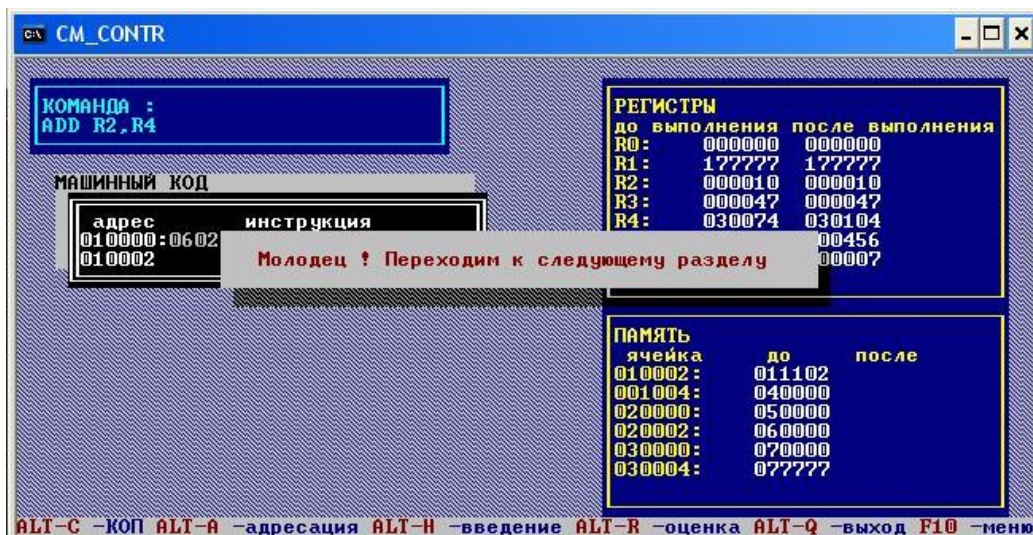


Рис. 4.6. Окно правильного выполнения задания.

4. СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен включать:

- окно окончания лабораторной работы, которое необходимо показать преподавателю (см. рис. 4.7.);

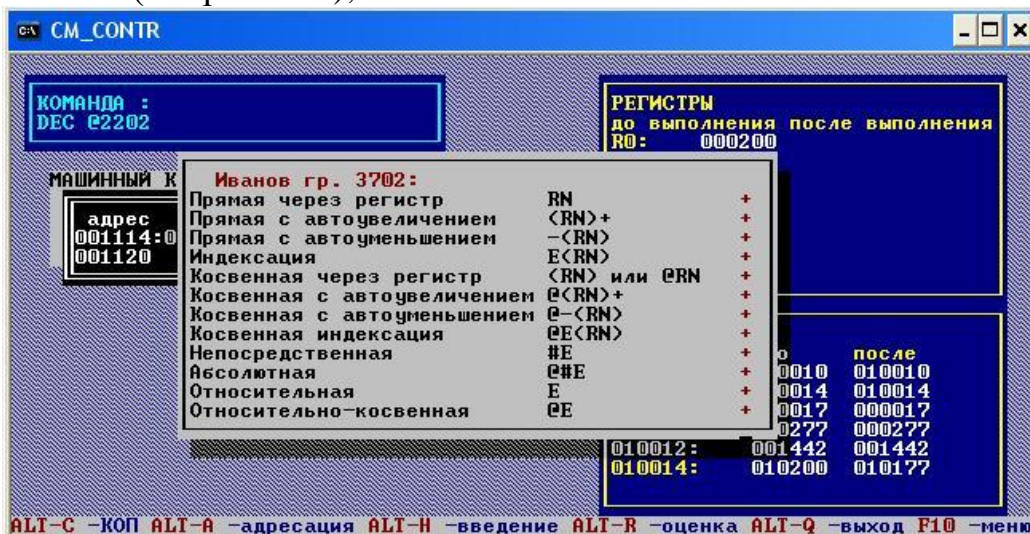


Рис. 4.7. Окно о завершении работы студентом.

- ответить на контрольные вопросы

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

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