

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ильшат Ринатович Мухаметьянов

Должность: директор

Дата подписания: 13.07.2023 12:35:18

Уникальный идентификатор:

aba80b84033c9ef196388e9ea0434f90a83a40954ba270e84bcb64f02d1d8d0

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический

университет им. А.Н. Туполева-КАИ»

(КНИТУ-КАИ)

Чистопольский филиал «Восток»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ**  
по дисциплине  
**ОРГАНИЗАЦИЯ ЭВМ И СИСТЕМ**

Индекс по учебному плану: **Б1.В.09**

Направление подготовки: **09.03.01 Информатика и вычислительная техника**

Квалификация: **Бакалавр**

Профиль подготовки: **Вычислительные машины, комплексы, системы и сети**

Типы задач профессиональной деятельности: **проектный,**

**производственно-технологический**

Рекомендовано УМК ЧФ КНИТУ-КАИ

Чистополь

2023 г.

## Введение

Данные методические указания предназначены для проведения лабораторных работ по дисциплине «Организация ЭВМ и систем»

Цикл лабораторных работ включает задания различного уровня.

Лабораторные работы ориентированы на изучение структуры команд и способов адресации ЭВМ с классической архитектурой фон Неймана.

№ п/п	№ раздела	Наименование лабораторных работ	Трудоемкость (час.)
1	1	Прямая адресация	6
2	1	Косвенная адресация	6
3	1	Индексная адресация	6
4	1	Непосредственная адресация	4
5	1	Абсолютная адресация	4
6	1	Относительная адресация	6

# **1 Лабораторная работа №1 «Прямая адресация »**

## **1.1 Цель работы**

Целью настоящей лабораторной работы является изучение выполнения различных команд с прямой адресацией через регистр, с прямой адресацией с автоувеличением и с прямой адресацией с автоуменьшением.

## **1.2 Общие положения**

В лабораторной работе №1 задаются различные команды выполнения операций в виде формулы на языке Ассемблер с использованием способа прямой адресации операндов.

Пример 1 «Прямая адресация через регистр»

Пример 2 «Прямая адресация с автоувеличением»

Пример 3 «Прямая адресация с автоуменьшением»

В соответствии с заданной формулой команды необходимо перевести команду ассемблера в формат машинной команды и определить все этапы выполнения каждой команды.

## **1.3 Задание на лабораторную работу №1**

1. Ознакомиться с построением адресации и форматом команд ЭВМ (Приложение 2).
2. Перевести команду ассемблера в формат машинной команды.
3. Определить адрес следующей команды.
4. Определить содержимое регистров и ячеек памяти в соответствии с выполняемой командой.
5. Зафиксировать результаты выполнения команды в виде копии экрана компьютера.

## **1.4 Ход работы**

В процессе работы выполнить следующие действия:

- 1) включить компьютер;
- 2) загрузить программную модель выполнения команд ADR SM;

- 3) выбрать соответствующий способ прямой адресации;
- 4) выполнить команду с выбранным способом адресации;
- 5) зафиксировать результаты выполнения команды в виде копии экрана компьютера.

## 1.5 Выполнение команд с прямым способом адресации

### 1.5.1 Прямая адресация через регистр

В данном случае в формуле команды на языке Ассемблер указывается номер регистра:

A: R<sub>N</sub>

Код адресации в этом случае равен: **0**

Операнд задается непосредственно в регистре, определяемом выражением R<sub>N</sub>

Команда: **DEC R1**

Данная команда в машинном коде:

	адрес	Инструкция 1 слово	Инструкция 2 слово	Инструкция 3 слово
СчК	030006:	005301		
СчК+2	030010			
СчК+4				

**Количество регистров, изменивших состояние?**

1

**Количество ячеек памяти, изменивших состояние?**

0

Регистр	До	После
R0	000000	000000
R1	011000	010777
R2	050555	050555
R3	046666	046666
R4	000463	000463
R5	000004	000004
R6	000006	000006

Ячейка	До	После
--------	----	-------

030000	020200	
030002	020220	
030004	020000	
030006	017777	
030010	003476	
030012	000454	

Для задания команды в машинном коде необходимо указать код операции адрес источника и адрес приемника (Рис.1).

Код операции определяем по таблице из Приложения 3.

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

Рисунок 1- Структура двухадресной команды

Поскольку рассматриваемая в данном примере команда одноадресная, то код операции представляет собой четыре восьмеричных цифры.

В оставшиеся две позиции команды вставляем код адресации (приложение 3), который в данном примере имеет значение **0**, и номер регистра общего назначения (РОН), который имеет значение **1**, т.к. в формуле задан регистр R1.

При данном способе адресации операнд содержится в регистре, т.е. в данном случае в регистре R1. Содержимое регистра R1 до выполнения операции равно 011000 в восьмеричном коде. Команда DEC уменьшает значение операнда на единицу, т.е. после выполнения команды содержимое регистра R1 станет равным 010777.

При данном способе адресации работа с ячейками памяти не производится.

### 1.5.2 Прямая адресация с автоувеличением

В данном случае в формуле команды на языке Ассемблер указывается номер регистра в скобках и ставится знак «+» справа:

A: (R<sub>N</sub>)+

Код адресации в этом случае равен: 2

Регистр R содержит адрес операнда и после выполнения инструкции увеличивается на 2 для операций над словами и на 1 для операций над байтами.

Команда: **DECB (R2) +**

Данная команда в машинном коде:

	адрес	Инструкция 1 слово	Инструкция 2 слово	Инструкция 3 слово
СчК :	010000	105322		
СчК+2:	010002			
СчК+4:				

**Количество регистров, изменивших состояние?**

1

**Количество ячеек памяти, изменивших состояние?**

1

Регистр	До	После
R0	030111	030111
R1	050200	050200
R2	020002	020003
R3	040200	040200
R4	030200	030200
R5	071170	071170
R6	016666	016666

Ячейка	До	После
020000	050000	050000
020002	030010	030007
020004	017777	017777
030000	013333	013333
030006	020001	020001
030010	101111	101111

Поскольку рассматриваемая в данном примере команда одноадресная, то код операции представляет собой четыре восьмеричных цифры.

В оставшиеся две позиции команды вставляем код адресации (приложение 3), который в данном примере имеет значение **2**, и номер регистра общего назначения (РОН), который имеет значение **2**, т.к. в формуле задан регистр R2.

При данном способе адресации в регистре содержится адрес операнда, т.е. в данном случае адрес операнда указан в регистре R2. Содержимое регистра R2 до выполнения операции равно 020002 в восьмеричном коде, после выполнения команды его содержимое автоматически увеличивается на единицу (байтовая операция) и становится равным: 020003.

Искомый операнд (по содержимому регистра R2) находится в ячейке с адресом 020002 и имеет значение 030010.

Команда DEC уменьшает значение операнда на единицу, т.е. после выполнения команды содержимое ячейки 020002 станет равным 030007.

### 1.5.3 Прямая адресация с автоуменьшением

В данном случае в формуле команды на языке Ассемблер указывается номер регистра в скобках и слева ставится знак «-»:

A: -(R<sub>N</sub>)

Код адресации в этом случае равен: 4

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

Команда: **DECB – (R6)**

Данная команда в машинном коде:

	адрес	Инструкция 1 слово	Инструкция 2 слово	Инструкция 3 слово
СчК :	001100	105346		
СчК+2:	001102			
СчК+4:				

**Количество регистров, изменивших состояние?**

1

**Количество ячеек памяти, изменивших состояние?**

1

Регистр	До	После
R0	012000	012000
R1	014000	014000
R2	054000	054000
R3	017700	017700
R4	024000	024000
R5	003100	003100
R6	003200	003177

Ячейка	До	После
003100	040000	040000
003176	001234	000634
000320	017777	017777
003202	020400	020400
040000	044444	044444
040002	033333	033333

Поскольку рассматриваемая в данном примере команда одноадресная, то код операции представляет собой четыре восьмеричных цифры.

В оставшиеся две позиции команды вставляем код адресации (приложение 3), который в данном примере имеет значение **4**, и номер регистра общего назначения (РОН), который имеет значение **6**, т.к. в формуле задан регистр R6.

При данном способе адресации в регистре содержится адрес операнда, т.е. в данном случае адрес операнда указан в регистре R6.

Содержимое регистра R6 до выполнения операции содержимое автоматически увеличивается на единицу (байтовая операция) и становится равным: 003177.

Поскольку адрес нечетный, то операндом является старший байт содержимого ячейки с меньшим четным адресом 003176: 001234.

Находим старший байт числа 001234 и вычитаем из этого байта единицу в соответствии с операцией **DECB**

0	0	1	2		3	4
0	000	001	0	10	011	100



-			1			
0	000	000	1			
0	000	000	110		011	100
0	0	0	6		3	4

После выполнения команды содержимое ячейки 003176 станет равным 000634.

### **1.6 Представление результатов работы**

По окончании лабораторной работы студент представляет результаты преподавателю в устной форме

### **Выводы**

В результате выполнения данной лабораторной работы приобретены навыки выполнения различных команд с прямой адресацией через регистр, с прямой адресацией с автоувеличением и с прямой адресацией с автоуменьшением

## Список литературы

1. Павлов, А.В. Архитектура вычислительных систем [Электронный ресурс] : учеб. пособие — Электрон. дан. — Санкт-Петербург : НИУ ИТМО, 2016. — 86 с. — Режим доступа: <https://e.lanbook.com/book/91328>. — Загл. с экрана.
2. Душкин, А. В. Вычислительная техника: Учебное пособие / Душкин А.В., Ланкин О.В., Чекризов Р.В. - Воронеж:Воронежский институт ФСИН России, 2015. - 325 с. ISBN 978-5-4446-0731-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/924589> (дата обращения: 06.11.2020). – Режим доступа: по подписке.
3. Архитектура ЭВМ : учеб. пособие: Учебное пособие / Жмакин А.П., - 2-е изд., перераб. и доп. - СПб:БХВ-Петербург, 2010. - 347 с. ISBN 978-5-9775-0550-5 ([www.znanium.com](http://www.znanium.com))
4. Бройдо В.Л., Ильина О.П. Вычислительные системы, сети и телекоммуникации. 3-е изд. - СПб.: Питер, 2008. -766 с.
5. Жмакин А.П. Архитектура ЭВМ.- СПб.: БХВ-Петербург, 2006. – 320 с.
6. Белош В.В. «Организация ЭВМ и систем» [Электронный ресурс]: Методические материалы по курсу дистанционного обучения по направлению подготовки бакалавров 09.03.01 «Информатика и ВТ» / КНИТУ-КАИ, Казань, 2016 – Доступ по логину и паролю. URL: [https://bb.kai.ru:8443/webapps/blackboard/content/listContentEditable.jsp?content\\_id=\\_216395\\_1&course\\_id=\\_11998\\_1](https://bb.kai.ru:8443/webapps/blackboard/content/listContentEditable.jsp?content_id=_216395_1&course_id=_11998_1)

## **2 Лабораторная работа №2 «Косвенная адресация»**

### **2.1 Цель работы**

Целью настоящей лабораторной работы является изучение выполнения различных команд с косвенной адресацией через регистр, с косвенной адресацией с автоувеличением и с косвенной адресацией с автоуменьшением.

### **2.2 Общие положения**

В лабораторной работе №2 задаются различные команды выполнения операций в виде формулы на языке Ассемблер с использованием способа косвенной адресации операндов.

Пример 1 «Косвенная адресация через регистр»

Пример 2 «Косвенная адресация с автоувеличением»

Пример 3 «Косвенная адресация с автоуменьшением»

В соответствии с заданной формулой команды необходимо перевести команду ассемблера в формат машинной команды и определить все этапы выполнения каждой команды

### **2.3 Задание на лабораторную работу №2**

1. Ознакомиться с построением адресации и форматом команд ЭВМ (Приложение 2).
2. Перевести команду ассемблера в формат машинной команды.
3. Определить адрес следующей команды.
4. Определить содержимое регистров и ячеек памяти в соответствии с выполняемой командой.
5. Зафиксировать результаты выполнения команды в виде копии экрана компьютера.

## 2.4 Ход работы

В процессе работы выполнить следующие действия:

- 1) включить компьютер;
- 2) загрузить программную модель выполнения команд ADR SM;
- 3) выбрать соответствующий способ косвенной адресации;
- 4) выполнить команды с выбранным способом адресации;
- 5) зафиксировать результаты выполнения команд в виде копий экрана компьютера.

## 2.5 Выполнение команд с косвенным способом адресации

### 2.5.1 Косвенная адресация через регистр

В данном случае в формуле команды на языке Ассемблер перед номером регистра указывается значок @ или номер регистра указывается в скобках:

А: @R<sub>N</sub> или (R<sub>N</sub>)

Код адресации в этом случае равен: 1

При данном способе адресации регистр R<sub>N</sub> содержит адрес операнда.

Команда: **CLR @R1**

Данная команда в машинном коде:

	адрес	Инструкция 1 слово	Инструкция 2 слово	Инструкция 3 слово
СчК :	010000	005011		
СчК+2:	010002			
СчК+4:				

**Количество регистров, изменивших состояние?**

0

**Количество ячеек памяти, изменивших состояние?**

1

Регистр	До	После
R0	022000	
R1	020000	
R2	000220	
R3	000064	
R4	000546	
R5	000078	
R6	000053	

Ячейка	До	После
020000	034653	000000
022000	000004	000004
020002	000045	000045
020004	000005	000005
030000	000003	000003
033000	000005	000005

Содержимое регистра **R1** означает, что операнд находится в ячейке памяти с адресом 020000. До выполнения операции операнд имеет значение 034653, команда **CLR** обнуляет его значение, т.е. после выполнения операции содержимое этой ячейки равно 000000.

### 2.5.2 Косвенная адресация с автоувеличением

В данном случае в формуле команды на языке Ассемблер перед номером регистра указывается значок @, затем номер регистра в скобках и знак «+» справа:

A: @(R<sub>N</sub>)+

Код адресации в этом случае равен: 3

При данном способе адресации регистр R<sub>N</sub> содержит указатель (адрес) адреса операнда. После выполнения инструкции содержимое регистра R<sub>N</sub> увеличивается на 2.

Команда: **CLR @(R3)+**

Данная команда в машинном коде:

	адрес	Инструкция 1 слово	Инструкция 2 слово	Инструкция 3 слово
СчК :	010000	005033		
СчК+2:	010002			
СчК+4:				

Количество регистров, изменивших состояние?

**1**

Количество ячеек памяти, изменивших состояние?

**1**

Регистр	До	После
R0	002000	002000
R1	002002	002002
R2	002004	002004
R3	020006	020010
R4	002010	002010
R5	002012	002012
R6	002014	002014

Ячейка	До	После
010000	000001	000001
010002	000002	000002
010004	000003	000000
020006	010004	010004
020010	010002	010002
020012	010000	010000

Содержимое регистра **R3** означает, что в ячейке памяти с адресом 020006 находится адрес операнда, т.е. адрес ячейки 010004. До выполнения операции операнд имеет значение 000003, команда **CLR** обнуляет его значение, т.е. после выполнения операции содержимое этой ячейки равно 000000.

### 2.5.3 Косвенная адресация с автоуменьшением

В данном случае в формуле команды на языке Ассемблер перед номером регистра указывается значок @, затем знак «-» и номер регистра в скобках:

A: @-(R<sub>N</sub>)

Код адресации в этом случае равен: 5

При данном способе адресации содержимое регистра R<sub>N</sub> уменьшается на 2 и рассматривается как указатель (адрес) адреса операнда.

Команда: **DECB @-(R2)**

Данная команда в машинном коде:

	адрес	Инструкция 1 слово	Инструкция 2 слово	Инструкция 3 слово
СчК :	001002	105352		
СчК+2:	001004			
СчК+4:				

Количество регистров, изменивших состояние?

1

Количество ячеек памяти, изменивших состояние?

1

Регистр	До	После
R0	003000	003000
R1	003002	003002
R2	003004	003002
R3	000100	000100
R4	000200	000200
R5	000300	000300
R6	000400	000400

Ячейка	До	После
003002	005600	005600
003004	004400	004400
004400	000001	000001
005200	000045	000045
005600	017576	017575
006000	013455	013455

Перед выполнением операции содержимое регистра **R2** уменьшается на 2 и становится равным 003002. Это означает, что в ячейке памяти с адресом 003002 находится адрес операнда, т.е. адрес ячейки 005600. До выполнения операции операнд имеет значение 017576. Поскольку адрес операнда 005600 является четным, то байтовая операция **DECB** выполняется с младшим байтом, поэтому просто вычитаем единицу из значения операнда.

После выполнения операции содержимое ячейки 005600 равно 017575.

## **2.6 Представление результатов работы**

По окончании лабораторной работы студент представляет результаты преподавателю в устной форме

### **Выводы**

В результате выполнения данной лабораторной работы приобретены навыки выполнения различных команд с косвенной адресацией через регистр, с косвенной адресацией с автоувеличением и с косвенной адресацией с автоуменьшением.



## Список литературы

1. Павлов, А.В. Архитектура вычислительных систем [Электронный ресурс] : учеб. пособие — Электрон. дан. — Санкт-Петербург : НИУ ИТМО, 2016. — 86 с. — Режим доступа: <https://e.lanbook.com/book/91328>. — Загл. с экрана.
2. Душкин, А. В. Вычислительная техника: Учебное пособие / Душкин А.В., Ланкин О.В., Чекризов Р.В. - Воронеж:Воронежский институт ФСИИ России, 2015. - 325 с. ISBN 978-5-4446-0731-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/924589> (дата обращения: 06.11.2020). – Режим доступа: по подписке.
3. Архитектура ЭВМ : учеб. пособие: Учебное пособие / Жмакин А.П., - 2-е изд., перераб. и доп. - СПб:БХВ-Петербург, 2010. - 347 с. ISBN 978-5-9775-0550-5 ([www.znanium.com](http://www.znanium.com))
4. Бройдо В.Л., Ильина О.П. Вычислительные системы, сети и телекоммуникации. 3-е изд. - СПб.: Питер, 2008. -766 с.
5. Жмакин А.П. Архитектура ЭВМ.- СПб.: БХВ-Петербург, 2006. – 320 с.
6. Белош В.В. «Организация ЭВМ и систем» [Электронный ресурс]: Методические материалы по курсу дистанционного обучения по направлению подготовки бакалавров 09.03.01 «Информатика и ВТ» / КНИТУ-КАИ, Казань, 2016 – Доступ по логину и паролю. URL: [https://bb.kai.ru:8443/webapps/blackboard/content/listContentEditable.jsp?content\\_id=\\_216395\\_1&course\\_id=\\_11998\\_1](https://bb.kai.ru:8443/webapps/blackboard/content/listContentEditable.jsp?content_id=_216395_1&course_id=_11998_1)

### **3 Лабораторная работа №3 «Индексная адресация»**

#### **3.1 Цель работы**

Целью настоящей лабораторной работы является изучение выполнения различных команд с прямой адресацией с индексированием и команд с косвенной индексной адресацией.

#### **3.2 Общие положения**

В лабораторной работе №3 задаются различные команды выполнения операций в виде формулы на языке Ассемблер с использованием способа прямой адресации с индексированием и способа с косвенной индексной адресацией.

В соответствии с заданной формулой команды необходимо перевести команду ассемблера в формат машинной команды и определить все этапы выполнения каждой команды

#### **3.3 Задание на лабораторную работу №3**

1. Ознакомиться с построением адресации и форматом команд ЭВМ (Приложение 2).
2. Перевести команду ассемблера в формат машинной команды.
3. Определить адрес следующей команды.
4. Определить содержимое регистров и ячеек памяти в соответствии с выполняемой командой.
5. Зафиксировать результаты выполнения команды в виде копии экрана компьютера.

### 3.4 Ход работы

В процессе работы выполнить следующие действия:

- 1) включить компьютер;
- 2) загрузить программную модель выполнения команд ADR SM;
- 3) выбрать соответствующий способ индексной адресации;
- 4) выполнить команды с данным способом адресации;
- 5) зафиксировать результаты выполнения команд в виде копий экрана компьютера.

### 3.5 Выполнение команд с индексной адресацией

#### 3.5.1 Прямая адресация с индексированием

В данном случае в формуле команды на языке Ассемблер указывается величина смещения **E** и номер регистра в скобках:

$$A: E(R_N)$$

Код адресации в этом случае равен: **6**

Значение выражение **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, содержимого **R** – база.

Команда: **DEC 30(R2)**

Данная команда в машинном коде:

	адрес	Инструкция 1 слово	Инструкция 2 слово	Инструкция 3 слово
СчК :	033000	005362	000030	
СчК+2:	001102			
СчК+4:	033004			

**Количество регистров, изменивших состояние?**

0

**Количество ячеек памяти, изменивших состояние?**

1

Регистр	До	После
R0	023453	
R1	026545	
R2	001770	
R3	014444	
R4	024444	
R5	000035	
R6	000065	

Ячейка	До	После
001770	000032	000032
002000	005453	005453
002010	000004	000004
002020	000010	000007
002040	000065	000065
002060	000654	000654

Поскольку рассматриваемая в данном примере команда одноадресная, то код операции представляет собой четыре восьмеричных цифры.

В оставшиеся две позиции команды вставляем код адресации (приложение 3), который в данном примере имеет значение **6**, и номер регистра общего назначения (РОН), который имеет значение **2**, т.к. в формуле задан регистр R2.

Во второе слово инструкции, которое начинается с адреса СчК+2, вставляем смещение 000030.

Поскольку длина команды составляет два слова, то адрес следующей команды будет СчК+4, т.е. 033004.

Адрес операнда находим как сумму содержимого регистра R2 и смещения E:

$$001770 + 000030 = 002020$$

Содержимое ячейки 002020 (операнд) равно 000010, после выполнения команды **DEC** станет равным 000007

### 3.5.2 Косвенная индексная адресация

В данном случае в формуле команды на языке Ассемблер перед номером регистра указывается значок @, затем смещение **E** и номер регистра в скобках:

A: @ E (R<sub>N</sub>)

Код адресации в этом случае равен: 7

При данном способе адресации смещение **E** складывается с содержимым регистра R<sub>N</sub> и рассматривается как указатель (адрес) адреса операнда.

Команда: **INC @40 (R0)**

Данная команда в машинном коде:

	адрес	Инструкция 1 слово	Инструкция 2 слово	Инструкция 3 слово
СчК :	010000	005270	000040	
СчК+2:				
СчК+4:	010004			

Количество регистров, изменивших состояние?

**0**

Количество ячеек памяти, изменивших состояние?

**1**

Регистр	До	После
R0	001000	
R1	001002	
R2	001010	
R3	001012	
R4	001020	
R5	002000	
R6	000300	

Ячейка	До	После
001000	000001	000001
001020	004000	004000
000100	000077	000077
001040	030000	030000
030000	000177	000200
004000	002777	002777

Поскольку рассматриваемая в данном примере команда одноадресная, то код операции представляет собой четыре восьмеричных цифры.

В оставшиеся две позиции команды вставляем код адресации (приложение 3), который в данном примере имеет значение 7, и номер регистра общего назначения (РОН), который имеет значение 0, т.к. в формуле задан регистр R0.

Во второе слово инструкции, которое начинается с адреса СчК+2, вставляем смещение 000040.

Поскольку длина команды составляет два слова, то адрес следующей команды будет СчК+4, т.е. 010004

Указатель (адрес) адреса операнда находим как сумму содержимого регистра R0 и смещения E:

$$001000 + 000040 = 001040$$

Содержимое ячейки 001040 (адрес операнда) равно 030000, т.е. операнд находится в этой ячейке и равен 000177. Команда INC увеличивает значение операнда на единицу и после выполнения команды операнд имеет значение 000200.

### **3.6 Представление результатов работы**

По окончании лабораторной работы студент представляет результаты преподавателю в устной форме

### **Выводы**

В результате выполнения данной лабораторной работы приобретены навыки выполнения различных команд с использованием способа прямой адресации с индексированием и способа с косвенной индексной адресацией.

## Список литературы

1. Павлов, А.В. Архитектура вычислительных систем [Электронный ресурс] : учеб. пособие — Электрон. дан. — Санкт-Петербург : НИУ ИТМО, 2016. — 86 с. — Режим доступа: <https://e.lanbook.com/book/91328>. — Загл. с экрана.
2. Душкин, А. В. Вычислительная техника: Учебное пособие / Душкин А.В., Ланкин О.В., Чекризов Р.В. - Воронеж:Воронежский институт ФСИН России, 2015. - 325 с. ISBN 978-5-4446-0731-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/924589> (дата обращения: 06.11.2020). – Режим доступа: по подписке.
3. Архитектура ЭВМ : учеб. пособие: Учебное пособие / Жмакин А.П., - 2-е изд., перераб. и доп. - СПб:БХВ-Петербург, 2010. - 347 с. ISBN 978-5-9775-0550-5 ([www.znanium.com](http://www.znanium.com))
4. Бройдо В.Л., Ильина О.П. Вычислительные системы, сети и телекоммуникации. 3-е изд. - СПб.: Питер, 2008. -766 с.
5. Жмакин А.П. Архитектура ЭВМ.- СПб.: БХВ-Петербург, 2006. – 320 с.
6. Белош В.В. «Организация ЭВМ и систем» [Электронный ресурс]: Методические материалы по курсу дистанционного обучения по направлению подготовки бакалавров 09.03.01 «Информатика и ВТ» / КНИТУ-КАИ, Казань, 2016 – Доступ по логину и паролю. URL: [https://bb.kai.ru:8443/webapps/blackboard/content/listContentEditable.jsp?content\\_id=\\_216395\\_1&course\\_id=\\_11998\\_1](https://bb.kai.ru:8443/webapps/blackboard/content/listContentEditable.jsp?content_id=_216395_1&course_id=_11998_1)

## **4 Лабораторная работа №4 «Непосредственная адресация»**

### **4.1 Цель работы**

Целью настоящей лабораторной работы является изучение выполнения различных команд с непосредственной адресацией.

### **4.2 Общие положения**

В лабораторной работе №4 задаются различные команды выполнения операций в виде формулы на языке Ассемблер с использованием способа непосредственной адресации.

В соответствии с заданной формулой команды необходимо перевести команду ассемблера в формат машинной команды и определить все этапы выполнения каждой команды

### **4.3 Задание на лабораторную работу №4**

1. Ознакомиться с построением адресации и форматом команд ЭВМ (Приложение 2).
2. Перевести команду ассемблера в формат машинной команды.
3. Определить адрес следующей команды.
4. Определить содержимое регистров и ячеек памяти в соответствии с выполняемой командой.
5. Зафиксировать результаты выполнения команды в виде копии экрана компьютера.

### **4.4 Ход работы**

В процессе работы выполнить следующие действия:

- 1) включить компьютер;
- 2) загрузить программную модель выполнения команд ADR SM;
- 3) выбрать способ адресации «непосредственная адресация»;
- 4) выполнить команду с данным способом адресации;



5) зафиксировать результаты выполнения команд в виде копий экрана компьютера.

#### 4.5 Выполнение команд с непосредственным способом адресации

В данном случае в формуле команды на языке Ассемблер перед номером регистра указывается значок #, затем операнд **E**:

A: # E

Код адресации в этом случае равен: 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 **R7** (счетчика команд).

Команда: **ADD #30, @R3**

Данная команда в машинном коде:

	адрес	Инструкция 1 слово	Инструкция 2 слово	Инструкция 3 слово
СчК :	013002	062713	000030	
СчК+2:				
СчК+4:	013006			

Количество регистров, изменивших состояние?

**0**

Количество ячеек памяти, изменивших состояние?

**1**

Регистр	До	После
R0	030000	
R1	000030	
R2	000005	
R3	030000	
R4	010000	
R5	070400	
R6	000001	

Ячейка	До	После
030000	050000	050030
030030	000055	000055
040000	070000	070000
040030	000040	000040
050000	000020	000020
050030	000177	000177

Рассматриваемая в данном примере команда использует два способа адресации: источник адресуется с помощью непосредственной адресации, а приемник с помощью косвенной адресации через регистр.

Непосредственная адресация использует регистр R7.

В машинном коде команда будет иметь вид: 062713.

Непосредственный операнд 000030 вставляем во второе слово команды.

Поскольку длина команды составляет два слова, то адрес следующей команды будет СчК+4, т.е. 013006

Первый операнд 000030 складывается со вторым операндом (содержимое ячейки 030000) 050000 и результат записывается по адресу второго операнда, т.е. в ячейку 030000:

$$000030 + 050000 = 050030$$

Содержимое ячейки 030000 после выполнения команды имеет значение 050030.

#### **4.6 Представление результатов работы**

По окончании лабораторной работы студент представляет результаты преподавателю в устной форме

#### **Выводы**

В результате выполнения данной лабораторной работы приобретены навыки выполнения различных команд с непосредственной адресацией.

## Список литературы

1. Павлов, А.В. Архитектура вычислительных систем [Электронный ресурс] : учеб. пособие — Электрон. дан. — Санкт-Петербург : НИУ ИТМО, 2016. — 86 с. — Режим доступа: <https://e.lanbook.com/book/91328>. — Загл. с экрана.
2. Душкин, А. В. Вычислительная техника: Учебное пособие / Душкин А.В., Ланкин О.В., Чекризов Р.В. - Воронеж:Воронежский институт ФСИН России, 2015. - 325 с. ISBN 978-5-4446-0731-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/924589> (дата обращения: 06.11.2020). – Режим доступа: по подписке.
3. Архитектура ЭВМ : учеб. пособие: Учебное пособие / Жмакин А.П., - 2-е изд., перераб. и доп. - СПб:БХВ-Петербург, 2010. - 347 с. ISBN 978-5-9775-0550-5 ([www.znanium.com](http://www.znanium.com))
4. Бройдо В.Л., Ильина О.П. Вычислительные системы, сети и телекоммуникации. 3-е изд. - СПб.: Питер, 2008. -766 с.
5. Жмакин А.П. Архитектура ЭВМ.- СПб.: БХВ-Петербург, 2006. – 320 с.
6. Белош В.В. «Организация ЭВМ и систем» [Электронный ресурс]: Методические материалы по курсу дистанционного обучения по направлению подготовки бакалавров 09.03.01 «Информатика и ВТ» / КНИТУ-КАИ, Казань, 2016 – Доступ по логину и паролю. URL: [https://bb.kai.ru:8443/webapps/blackboard/content/listContentEditable.jsp?content\\_id=\\_216395\\_1&course\\_id=\\_11998\\_1](https://bb.kai.ru:8443/webapps/blackboard/content/listContentEditable.jsp?content_id=_216395_1&course_id=_11998_1)

## **5 Лабораторная работа №5 «Абсолютная адресация»**

### **5.1 Цель работы**

Целью настоящей лабораторной работы является изучение выполнения различных команд с абсолютной адресацией.

### **5.2 Общие положения**

В лабораторной работе №5 задаются различные команды выполнения операций в виде формулы на языке Ассемблер с использованием способа абсолютной адресации.

В соответствии с заданной формулой команды необходимо перевести команду ассемблера в формат машинной команды и определить все этапы выполнения каждой команды

### **5.3 Задание на лабораторную работу №5**

1. Ознакомиться с построением адресации и форматом команд ЭВМ (Приложение 2).
2. Перевести команду ассемблера в формат машинной команды.
3. Определить адрес следующей команды.
4. Определить содержимое регистров и ячеек памяти в соответствии с выполняемой командой.
5. Зафиксировать результаты выполнения команды в виде копии экрана компьютера.

### **5.4 Ход работы**

В процессе работы выполнить следующие действия:

- 1) включить компьютер;
- 2) загрузить программную модель выполнения команд ADR SM;
- 3) выбрать способ адресации «абсолютная адресация»;
- 4) выполнить команду с данным способом адресации;

5) зафиксировать результаты выполнения команд в виде копий экрана компьютера.

### 5.5 Выполнение команд с абсолютным способом адресации

В данном случае в формуле команды на языке Ассемблер перед абсолютным адресом **E** указываются значки **@#**. Абсолютный адрес **E** хранится во втором или третьем слове инструкции.

А: **@#E**

Код адресации в этом случае равен: 3

Команда: **DEC @#20000**

Данная команда в машинном коде:

	адрес	Инструкция 1 слово	Инструкция 2 слово	Инструкция 3 слово
СчК:	010000	005337	020000	
СчК+2:				
СчК+4:	010004			

Количество регистров, изменивших состояние?

**0**

Количество ячеек памяти, изменивших состояние?

**1**

Регистр	До	После
R0	000001	
R1	000012	
R2	120000	
R3	020000	
R4	030000	
R5	004444	
R6	044017	

Ячейка	До	После
010000	050000	050000
020000	010000	007777
030000	000007	000007
040000	000077	000077
050000	017770	017770
060000	020000	020000

Рассматриваемая в данном примере команда использует два способа адресации: источник адресуется с помощью непосредственной адресации, а приемник с помощью косвенной адресации через регистр.

Абсолютная адресация использует регистр R7.

В машинном коде команда будет иметь вид: 005337.

Абсолютный адрес операнда 020000 указываем во втором слове команды.

Поскольку длина команды составляет два слова, то адрес следующей команды будет СчК+4, т.е. 010004

Команда **DEC** уменьшает значение операнда 010000 (содержимого ячейки с абсолютным адресом 020000) на единицу, т.е. содержимое ячейки 020000 после выполнения команды имеет значение 007777.

## **5.6 Представление результатов работы**

По окончании лабораторной работы студент представляет результаты преподавателю в устной форме

### **Выводы**

В результате выполнения данной лабораторной работы приобретены навыки выполнения различных команд с абсолютной адресацией.

## Список литературы

1. Павлов, А.В. Архитектура вычислительных систем [Электронный ресурс] : учеб. пособие — Электрон. дан. — Санкт-Петербург : НИУ ИТМО, 2016. — 86 с. — Режим доступа: <https://e.lanbook.com/book/91328>. — Загл. с экрана.
2. Душкин, А. В. Вычислительная техника: Учебное пособие / Душкин А.В., Ланкин О.В., Чекризов Р.В. - Воронеж:Воронежский институт ФСИИ России, 2015. - 325 с. ISBN 978-5-4446-0731-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/924589> (дата обращения: 06.11.2020). – Режим доступа: по подписке.
3. Архитектура ЭВМ : учеб. пособие: Учебное пособие / Жмакин А.П., - 2-е изд., перераб. и доп. - СПб:БХВ-Петербург, 2010. - 347 с. ISBN 978-5-9775-0550-5 ([www.znanium.com](http://www.znanium.com))
4. Бройдо В.Л., Ильина О.П. Вычислительные системы, сети и телекоммуникации. 3-е изд. - СПб.: Питер, 2008. -766 с.
5. Жмакин А.П. Архитектура ЭВМ.- СПб.: БХВ-Петербург, 2006. – 320 с.
6. Белош В.В. «Организация ЭВМ и систем» [Электронный ресурс]: Методические материалы по курсу дистанционного обучения по направлению подготовки бакалавров 09.03.01 «Информатика и ВТ» / КНИТУ-КАИ, Казань, 2016 – Доступ по логину и паролю. URL: [https://bb.kai.ru:8443/webapps/blackboard/content/listContentEditable.jsp?content\\_id=\\_216395\\_1&course\\_id=\\_11998\\_1](https://bb.kai.ru:8443/webapps/blackboard/content/listContentEditable.jsp?content_id=_216395_1&course_id=_11998_1)



## **6 Лабораторная работа №6 «Относительная адресация»**

### **6.1 Цель работы**

Целью настоящей лабораторной работы является изучение выполнения различных команд с относительной адресацией и относительно-косвенной адресацией.

### **6.2 Общие положения**

В лабораторной работе №6 задаются различные команды выполнения операций в виде формулы на языке Ассемблер с использованием способа относительной адресации и относительно-косвенной адресации.

В соответствии с заданной формулой команды необходимо перевести команду ассемблера в формат машинной команды и определить все этапы выполнения каждой команды

### **6.3 Задание на лабораторную работу №6**

1. Ознакомиться с построением адресации и форматом команд ЭВМ (Приложение 2).
2. Перевести команду ассемблера в формат машинной команды.
3. Определить адрес следующей команды.
4. Определить содержимое регистров и ячеек памяти в соответствии с выполняемой командой.
5. Зафиксировать результаты выполнения команды в виде копии экрана компьютера.

## 6.4 Ход работы

В процессе работы выполнить следующие действия:

- 1) включить компьютер;
- 2) загрузить программную модель выполнения команд ADR SM;
- 3) выбрать соответствующий способ относительной адресации;
- 4) выполнить команду с выбранным способом адресации;
- 5) зафиксировать результаты выполнения команд в виде копий экрана компьютера.

## 6.5 Выполнение команд с относительным способом адресации

### 6.5.1 Относительная адресация

В данном случае в формуле команды на языке Ассемблер указывается адрес операнда **E**.

Этот вид адресации интерпретируется как индексный режим с использованием регистра R7 (PC), т.е. счетчика команд, в качестве индексного регистра.

Смещение для вычисления адреса хранится во втором или третьем слове инструкции и задается как число, которое прибавляется к содержимому PC и дает адрес операнда.

Таким образом, смещение есть (**E** - PC), где **E** – адрес операнда

**A: E**

Код адресации в этом случае равен: 6

Команда: **CLR 1000**

Данная команда в машинном коде:

	адрес	Инструкция 1 слово	Инструкция 2 слово	Инструкция 3 слово
СчК:	000014	005067	000760	
СчК+2:				
СчК+4:	000020			

Количество регистров, изменивших состояние?

0

Количество ячеек памяти, изменивших состояние?

1

Регистр	До	После
R0	000700	
R1	000147	
R2	002000	
R3	001000	
R4	004200	
R5	001777	
R6	000470	

Ячейка	До	После
001000	077777	000000
001002	002222	002222
001004	047333	047333
010200	020666	020666
010400	017001	017001
010600	003204	003204

После выполнения текущей команды содержимое счетчика команд станет равным 000020. Эту величину необходимо отнять от абсолютного адреса 1000, чтобы получить искомое смещение для записи его в команду:

$$001000 - 000020 = 000760$$

При выполнении команды смещение, указанное в команде сложится с содержимым счетчика команд и даст искомый адрес операнда:

$$000760 + 000020 = 001000$$

Находим ячейку с адресом 01000, в которой содержится операнд 07777, и, в соответствии с кодом операции, обнуляем ее содержимое.

## 6.5.2 Относительно-косвенная адресация

В данном случае в формуле команды на языке Ассемблер указывается значок @ перед величиной **E**:

А: @E

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

Команда: **CLR @1000**

Данная команда в машинном коде:

	адрес	Инструкция 1 слово	Инструкция 2 слово	Инструкция 3 слово
СчК:	000070	005067	000760	
СчК+2:				
СчК+4:	000074			

Количество регистров, изменивших состояние?

**0**

Количество ячеек памяти, изменивших состояние?

**1**

Регистр	До	После
R0	000012	
R1	000007	
R2	000016	
R3	001777	
R4	002000	
R5	001000	
R6	001701	

Ячейка	До	После
001000	004000	004000
002000	006000	006000
003000	000700	000700
004000	017000	000000
005000	000177	000177
006000	002700	002700

После выполнения текущей команды содержимое счетчика команд станет равным 000074

Эту величину необходимо отнять от абсолютного адреса 1000, чтобы получить искомое смещение для записи его в команду:

$$001000 - 000074 = 000704$$

При выполнении команды смещение, указанное в команде сложится с содержимым счетчика команд и даст указатель адреса операнда:

$$000704 + 000074 = 001000$$

Находим ячейку с адресом 01000, в которой содержится адрес операнда 004000.

В ячейке с адресом 004000 находится операнд 017000, и, в соответствии с кодом операции, обнуляем ее содержимое.

## **6.6 Представление результатов работы**

По окончании лабораторной работы студент представляет результаты преподавателю в устной форме

### **Выводы**

В результате выполнения данной лабораторной работы приобретены навыки выполнения различных команд с относительной и с относительно-косвенной адресацией

## Список литературы

1. Павлов, А.В. Архитектура вычислительных систем [Электронный ресурс] : учеб. пособие — Электрон. дан. — Санкт-Петербург : НИУ ИТМО, 2016. — 86 с. — Режим доступа: <https://e.lanbook.com/book/91328>. — Загл. с экрана.
2. Душкин, А. В. Вычислительная техника: Учебное пособие / Душкин А.В., Ланкин О.В., Чекризов Р.В. - Воронеж:Воронежский институт ФСИИ России, 2015. - 325 с. ISBN 978-5-4446-0731-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/924589> (дата обращения: 06.11.2020). – Режим доступа: по подписке.
3. Архитектура ЭВМ : учеб. пособие: Учебное пособие / Жмакин А.П., - 2-е изд., перераб. и доп. - СПб:БХВ-Петербург, 2010. - 347 с. ISBN 978-5-9775-0550-5 ([www.znanium.com](http://www.znanium.com))
4. Бройдо В.Л., Ильина О.П. Вычислительные системы, сети и телекоммуникации. 3-е изд. - СПб.: Питер, 2008. -766 с.
5. Жмакин А.П. Архитектура ЭВМ.- СПб.: БХВ-Петербург, 2006. – 320 с.
6. Белош В.В. «Организация ЭВМ и систем» [Электронный ресурс]: Методические материалы по курсу дистанционного обучения по направлению подготовки бакалавров 09.03.01 «Информатика и ВТ» / КНИТУ-КАИ, Казань, 2016 – Доступ по логину и паролю. URL: [https://bb.kai.ru:8443/webapps/blackboard/content/listContentEditable.jsp?content\\_id=\\_216395\\_1&course\\_id=\\_11998\\_1](https://bb.kai.ru:8443/webapps/blackboard/content/listContentEditable.jsp?content_id=_216395_1&course_id=_11998_1)

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ»  
(КНИТУ-КАИ)  
Чистопольский филиал «Восток»

---

Кафедра компьютерных и телекоммуникационных систем

## **Отчет**

**по лабораторной работе № 1**

по дисциплине «Организация ЭВМ и систем»

**Прямая адресация**

Выполнил

ст. группы 21202 Петров И.И.

Проверил

к. т. н., доцент Белош В.В.

г. Чистополь

2023 г.



## **Отчет**

**по лабораторной работе № 2**

по дисциплине «Организация ЭВМ и систем»

**Косвенная адресация**

## **Отчет**

**по лабораторной работе № 3**

по дисциплине «Организация ЭВМ и систем»

**Индексная адресация**

## **Отчет**

**по лабораторной работе № 4**

по дисциплине «Организация ЭВМ и систем»

**Непосредственная адресация**

## **Отчет**

**по лабораторной работе № 5**

по дисциплине «Организация ЭВМ и систем»

**Абсолютная адресация**

## **Отчет**

**по лабораторной работе № 6**

по дисциплине «Организация ЭВМ и систем»

**Относительная адресация**

## Теоретическая часть

### Программная модель процессоров семейства PDP-11

#### 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. 1.

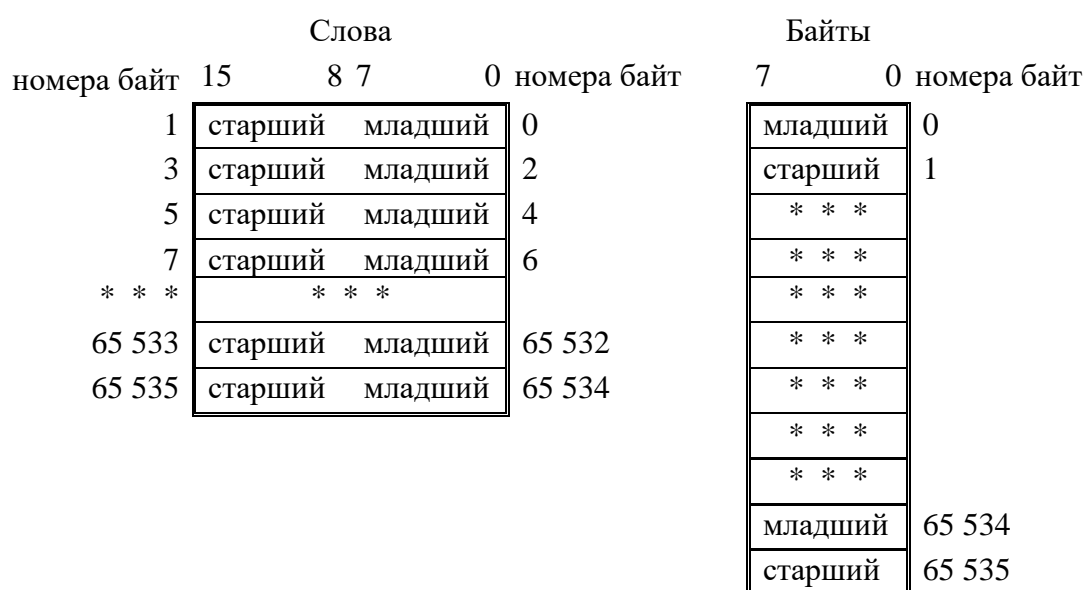


Рис. 1. Структура и адресация бит в байте и слов в памяти

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 пространство памяти.

### **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) 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. Структура регистра слова состояния процессора

### 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.

### Структура команд

Существенной новацией структуры команд ЭВМ PDP-11 является все-сторонне 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.3). Они содержат поля:

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



Рис.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. 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.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 "c" – 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 "e" – 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. (команды с суффиксом В – байтвые

Таблица 1. Коды операций двухместных команд.

№	Код	№	Код	Описание
0000	–	1000		В двухместных полно-адресных командах не используется
0001	MOV	1001	MOVВ	Пересылка содержимого источника в приемник
0010	СМР	1010	СМРВ	Сравнение источника с приемником, установка кода усло-
0011	ВИТ	1011	ВИТВ*	Тестирование битов (по маске**), установка кода условий
0100	ВИС	1100	ВИСВ*	Сброс битов (по маске**).

0101	BIS	1101	BISB*	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
Примечания: *– команды с суффиксом В – байтовые, **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 A<sub>1</sub> A<sub>2</sub> ;переслать слово,
- MOVB A<sub>1</sub> A<sub>2</sub> ;переслать байт,

где: A<sub>1</sub> A<sub>2</sub> – адреса памяти или РОН, заданные в команде.

### 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 операнда
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 операнда
5	@ –(R)	Автоуменьшение косвенная	Содержимое регистра уменьшается на 2, результат используется как адрес адреса операнда
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. 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 в направлении их увеличения;

б) 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 засылка 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. 6.

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. Для выявления особенностей адресаций операндов для каждого типа адресации с использованием и без использования счетчика команд в качестве регистра, их рассмотрение производится попарно: с использованием обычного РОНа и с использованием счетчика команд/

### ***Режим автоувеличения (2).***

Режим автоувеличения с использование обычного РОНа

**$\beta+0$  MOV (R0)+, R1** ;содержимого ячейки памяти, адрес которой (например, 1000) хранится в R0 переслать в регистр R1.

В этом примере  $\beta+0$  –является адресом выполняемой команды. После выполнения этой команды содержимое счетчика команд – R7 увеличивается на длину команды (в байтах) и становится равным  $\beta+2$ ; это адрес следующей команда (X),

$\beta+2$ ) X–следующая команда.

Режим автоувеличения с использование счетчика команд.

$\beta+0$ ) **MOV (R7)+, R1** ;та же команда, но в качестве регистра используется счетчик команд (R7); После выборки команды содержимое R7 увеличивается на длину команды (в байтах) и становится равным  $\beta+2$ . 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  $\beta+4$ .

Таким образом, при использовании в качестве РОНа счетчика команд, команда увеличивается на одно слово, содержимое которого 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 удлинится на слово X, где X используется в качестве абсолютного адреса операнда:

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

$\beta+2$ )X

$\beta+4$ )следующая команда.

**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

β+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. не привязанные 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).***

β+0) *MOV @X(R7)+, R1*

β+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*.

## ВИДЫ АДРЕСАЦИИ СМ ЭВМ

### 1. ПРЯМАЯ АДРЕСАЦИЯ

1.1. *П.а. ЧЕРЕЗ РЕГИСТР* А: RN код адресации :0  
Операнд задается непосредственно в регистре, определяемом выражением R.

Пример:  
CLR R0 ; очищение регистра 0

1.2. *П.а. С АВТОУВЕЛИЧЕНИЕМ* А: (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

1.3. *П.а. С АВТОУМЕНЬШЕНИЕМ* А: -(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

1.4. *ИНДЕКСНАЯ* А: 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

### 2. КОСВЕННАЯ АДРЕСАЦИЯ

2.1. *К.а. ЧЕРЕЗ РЕГИСТР* А: @RN или (RN) код адресации :1  
Регистр R содержит адрес операнда.

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

2.2. *К.а. С АВТОУВЕЛИЧЕНИЕМ* А: @(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

2.3. *К.а. 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.

2.4. *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. АДРЕСАЦИЯ С ИСПОЛЬЗОВАНИЕМ РС (счетчика команд)

3.1. *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.

3.2. *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.

### 3.3. 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 PC и дает адрес операнда.

Таким образом, смещение есть (E - PC), где E- 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. PC=20.

После чтения инструкции MOV PC 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 (PC).

Значение смещения процессор считывает по адресу, указанному в PC (в нашем примере содержимое PC = 22), после чего значение PC становится равным 24.

Для вычисления адреса операнда-источника значение смещения прибавляется к содержимому указанного регистра, т.е: адрес операнда вычисляется как смещение + PC = 54+24=100

Пример:

CLR 100 ; очистить ячейку 100

### 3.4. ОТНОСИТЕЛЬНО-КОСВЕННАЯ A: @E код адресации :7

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

Пример:

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



## Список используемых инструкций

Мнемоника	Восьмеричный 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	Содержимое адреса, определенного первым операндом, пересылается по адресу, заданному вторым операндом. Содержимое адреса, заданного первым операндом не меняется
MOVB	11	слова (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	(байт)	