

## Лабораторная работа 1

### ПРЕДСТАВЛЕНИЕ ДАННЫХ В ЭВМ ТИПА VAX-11. ФОРМАТЫ КОМАНД. АРИФМЕТИКО-ЛОГИЧЕСКИЕ ОПЕРАЦИИ С ЦЕЛОЧИСЛЕННЫМИ ДАННЫМИ

*Цель работы:* знакомство с архитектурой процессора VAX-11, изучение форматов команд и данных, типов адресации, системы арифметико-логических операций процессора VAX-11. Освоение симулятора процессора VAX-11 и выполнение программы арифметико-логической обработки целочисленных данных с использованием прямой адресации и различных способов косвенной адресации.

#### 1. Методические указания

В данной лабораторной работе изучаются представление данных и форматы основных команд в ЭВМ типа VAX-11, а также осуществляется выполнение простой программы арифметико-логической обработки данных с помощью программы-симулятора.

Данные в программе-симуляторе могут храниться или в 32-разрядных регистрах общего назначения (РОН) R0 – R15 (R0-RF), или в 8-битных ячейках памяти по адресам 0 – 2047 (0 - 7FF). Команды в программе могут быть размещены только в памяти по вышеприведенным адресам. При этом между командами и данными в хранении, записи и чтении нет принципиальной разницы. Доступный объем памяти для хранения команд и данных в симуляторе по умолчанию составляет 2 килобайта и при необходимости может быть увеличен, используя соответствующие настройки программы симулятора.

В лабораторной работе необходимо обратить особое внимание на то, что в ЭВМ типа VAX-11 машинные операции выполняются со словами различной длины: от одного до шестнадцати байт, а также на порядок построения команд. Здесь необходимо учитывать особенности размещения операндов в памяти и организации адресации операндов.

При выполнении команд форматы операндов определяются исключительно кодом операции. Так если указана операция со словами двойной длины (типа L), а операнды были определены размером в один байт, то операция будет выполнена со словами двойной длины. При этом старшие разряды операндов определяются текущими значениями старших трех байтов длинных слов по адресам указан-



ных операндов. Если указана операция с байтами, а операнды были определены как слова, то в операции будут участвовать только указанные байты. Если для указанных в заданиях арифметико-логических операций существует несколько машинных операций с данными различной длины, то необходимо выбирать тот код операции, который соответствует формату операнда-результата операции.

При составлении программы и трассировке ее алгоритма также необходимо учитывать порядок выполнения команд. В процессе выполнения команды сначала выбирается и анализируется код операции, затем вычисляется адрес первого операнда (со всеми изменениями регистров) и выбирается первый операнд, потом вычисляется адрес второго операнда и т. д.

## 2. Практическая часть

Практическая часть работы включает выполнение следующих действий:

а) формирование числовых значений операндов в соответствии с индивидуальным заданием, перевод их в шестнадцатеричную систему счисления и определение минимального формата представления исходных данных;

б) определение номеров РОН и адресов в памяти для размещения операндов;

в) составление и выполнение программы работы с данными, хранящимися в РОН или в памяти с использованием прямой и различных способов косвенной адресации, по заданному алгоритму;

г) тестирование программы путем ручной трассировки заданного алгоритма с последующим сравнением результатов работы программы с результатами ручной трассировки.

## 3. Варианты заданий

Значения исходных данных в десятичной системе счисления определяются следующими выражениями:

$$X1 = (-1)^{NV} [(NV + NGP) * 3];$$

$$X2 = (-1)^{(NV+1)} [(NV + NGL) * 2];$$

$$X3 = (-1)^{(NV+NGL)} [NV + NGP + NGL + 21];$$

$$X4 = (-1)^{NV} [NV + NGP + 29]^2;$$

$$X5 = (-1)^{(NV+1)} [NV + NGL + 23]^2;$$

$$X6 = (-1)^{(NV+NGL)} [NV + NGP + NGL + 79]^2;$$



$$X7 = X4^2;$$

$$X8 = -X5^2;$$

$$X9 = (-1)^{(NV+NGL)} X6^2.$$

Здесь и далее: NV – номер варианта индивидуального задания, NGL – последняя цифра номера группы, NGP – предпоследняя цифра номера группы.

В каждом варианте три операнда из девяти хранятся в РОН, а остальные – в памяти. В табл. 1.1 операнды, размещаемые в регистрах, помечены символами 'р'. Пустые клетки в табл. 1.1 соответствуют операндам в памяти. В табл. 1.2 указаны конкретные номера РОН для операндов, расположенных в регистрах. Например, для первого варианта в РОН размещаются операнды X1, X4 и X9. Присвоим операндам порядковые номера для поиска в табл. 1.2: G1 для X1, G2 для X4 и G3 для X9. Далее, согласно табл. 1.2 определяем, что операнд X1(G1) расположен в регистре R0, операнд X4(G2) – в R1, а операнд X9(G3) – в R2. Аналогично для операндов, расположенных в памяти введем порядковые номера M1, M2, ..., M6. Например, для первого варианта M1 соответствует операнду X2, M2 – операнду X3 и т. д.

Адреса размещения операндов в памяти в десятичной системе счисления определяются следующими выражениями:

$$\text{Адр}(M1) = NV * NGL + NGP,$$

$$\text{Адр}(M2) = NV * NGL + NGP + 10,$$

$$\text{Адр}(M3) = NV * NGL + NGP + 20,$$

$$\text{Адр}(M4) = NV * NGL + NGP + 30,$$

Таблица 1.1

Размещение операндов по вариантам

Данные	Варианты																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
X1	р						р			р					р		р		р	р
X2		р		р				р		р			р					р		
X3		р			р		р				р			р			р			
X4	р			р		р				р		р						р	р	
X5			р		р	р			р				р		р		р			
X6			р			р		р			р		р			р			р	
X7			р				р		р			р			р			р		р
X8		р			р			р			р			р		р				р
X9	р			р					р			р		р		р				



Таблица 1.2

## Размещение данных в регистрах по вариантам

Данные	Варианты																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
G1	0	B	1	7	4	5	2	E	6	9	A	8	3	C	D	9	3	4	5	7
G2	1	C	3	8	E	6	9	0	2	D	7	4	2	B	5	A	E	7	D	9
G3	2	6	A	D	9	7	B	6	3	1	0	E	C	4	8	5	B	A	1	0

$$\text{Адр}(M5) = NV * NGL + NGP + 40,$$

$$\text{Адр}(M6) = NV * NGL + NGP + 50.$$

Начальный адрес размещения программы в десятичной системе счисления определяется выражением:

$$\text{Адр}(\text{PROGRAM}) = NV * 20 + NGL * 100.$$

Варианты алгоритмов программ приведены на рис. 1.1. Номер реализуемого алгоритма определяется как один плюс остаток от деления суммы NV, NGL и NGP на 9. Операнд  $X_i'$  в алгоритмах используется в команде арифметического сдвига, всегда представ-

$\begin{aligned} X4 &= X5 + X6 \\ X8 &= -X8 \\ X7 &= X7 * 2^{X1'} \\ X3 &= X3 \vee X2 \\ X2 &= X2 + 1 \\ X9 &= X9 - X8 - C \end{aligned}$	$\begin{aligned} X9 &= X9 * 2^{X2'} \\ X6 &= X5 \oplus X4 \\ X7 &= X7 \& \overline{X8} \\ X1 &= \overline{X3} \\ X6 &= X6 + 1 \\ X9 &= X7 + X8 \end{aligned}$	$\begin{aligned} X7 &= X7 - X9 - C \\ X3 &= X1 \& \overline{X2} \\ X6 &= X6 + X5 \\ X9 &= X9 + 1 \\ X4 &= -X4 \\ X8 &= X8 * 2^{X3'} \end{aligned}$
Алгоритм 1	Алгоритм 2	Алгоритм 3
$\begin{aligned} X4 &= X4 \& \overline{X6} \\ X8 &= X8 + X9 + C \\ X5 &= 0 \\ X7 &= X8 * 2^{X3'} \\ X1 &= X1 \oplus X2 \\ X9 &= X8 - X7 \end{aligned}$	$\begin{aligned} X3 &= X1 + X2 \\ X9 &= \overline{X9} \\ X6 &= X6 - 1 \\ X4 &= X4 - X5 \\ X7 &= X8 \vee X9 \\ X8 &= X9 * 2^{X7'} \end{aligned}$	$\begin{aligned} X8 &= X8 - 1 \\ X1 &= X2 - X3 \\ X5 &= \overline{X6} \\ X9 &= X7 * 2^{X4'} \\ X9 &= X9 + X8 + C \\ X7 &= X7 - X8 \end{aligned}$
Алгоритм 4	Алгоритм 5	Алгоритм 6
$\begin{aligned} X1 &= -X2 \\ X8 &= X8 + X7 \\ X6 &= X5 + X4 \\ X3 &= X3 - 1 \\ X8 &= X8 * 2^{X9'} \\ X7 &= X7 + X8 + C \end{aligned}$	$\begin{aligned} X4 &= 0 \\ X9 &= X7 * 2^{X3'} \\ X7 &= X8 - X9 \\ X2 &= X2 - X1 \\ X6 &= -X5 \\ X1 &= X1 - 1 \end{aligned}$	$\begin{aligned} X4 &= X4 - X5 \\ X5 &= X6 \vee X4 \\ X8 &= X9 * 2^{X3'} \\ X1 &= X1 + X2 \\ X8 &= X8 \oplus X7 \\ X7 &= 0 \end{aligned}$
Алгоритм 7	Алгоритм 8	Алгоритм 9

Рис. 1.1. Алгоритмы программ



ляется в формате байта и определяется как остаток от деления  $X_i$  на 10. Обозначение  $C$  в алгоритмах соответствует признаку переноса, сформированного предыдущей командой алгоритма.

В табл. 1.3 указаны типы используемой адресации для каждого операнда, где 6 – косвенная регистровая адресация, 8 – автоинкрементная (косвенная с автоувеличением), 7 – автодекрементная (косвенная с автоуменьшением) и 9 – косвенная автоинкрементная адресация (двойная косвенная с автоувеличением).

Вариант комбинации способов адресации определяется как один плюс остаток от деления  $NV$  на 20.

В табл. 1.3 Оп1 – операнд в алгоритме по порядку (нумерация выполняется справа налево и сверху вниз), размещенный в памяти. Операнды, размещенные в регистрах, в приведенной нумерации не участвуют. Например, алгоритм 1 для первого варианта (табл. 1.1), с учетом новой нумерации операндов, приведен на рис. 1.2.

Промежуточные ячейки (ПЯ), используемые при реализации косвенной адресации, должны быть расположены, начиная с адреса:  $\text{Адр(ТЕМР)} = NV * NGP + 1000$  в десятичной системе счисления.

РОН для хранения адресов при косвенной адресации данных, выбираются произвольно из числа регистров, не используемых для хранения операндов.

Следует отметить, что операнды  $X_1$ ,  $X_4$  и  $X_9$  не пронумерованы, так как размещены в регистрах, вторая команда в алгоритме 1 – двухадресная, а третья команда – трехадресная, согласно системе команд.

Таблица 1.3

Способы адресации по вариантам

Данные	Варианты																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Оп1	7	6	6	8	9	7	8	9	6	7	8	9	9	8	7	6	7	8	9	6
Оп2	7	7	6	9	8	9	6	7	8	8	7	6	9	8	6	9	8	7	9	7
Оп3	8	8	6	6	7	9	7	6	6	8	9	7	6	7	6	8	8	7	8	9
Оп4	8	9	7	7	6	8	9	7	8	6	9	8	7	7	8	7	8	7	8	8
Оп5	6	9	9	9	6	7	8	9	7	6	6	9	9	9	9	8	6	9	7	7
Оп6	6	8	9	7	9	7	8	6	9	7	6	8	7	9	8	7	6	6	8	8
Оп7	9	7	7	8	8	6	9	8	7	8	7	7	8	6	9	9	9	6	6	9
Оп8	9	6	8	6	7	8	9	7	9	9	8	6	8	6	9	6	9	9	9	9
Оп9	7	9	8	7	7	8	6	8	7	9	8	9	6	7	6	9	7	8	6	6
Оп10	6	7	8	9	8	6	7	8	6	9	7	7	8	8	7	6	6	8	7	6



$$\begin{aligned}
 X4 &= \text{Оп2} + \text{Оп1} \\
 \text{Оп4} &= -\text{Оп3} \\
 \text{Оп6} &= \text{Оп5} * 2^{X1'} \\
 \text{Оп8} &= \text{Оп8} \vee \text{Оп7} \\
 \text{Оп9} &= \text{Оп9} + 1 \\
 X9 &= X9 - \text{Оп10} - C
 \end{aligned}$$

Рис. 1.2. Алгоритм 1 в новой нумерации операндов

Таблица 1.4

**«Каркас» текста программы**

Оператор	Адрес	Шестнадцатеричный код	Мнемонический код	Комментарий
1	Q	5R 7R 6R A1	ADDW3 (R), -(R), R	X4 = X5 + X6
2	Q+4	9R 8R CE	MNEGL (R)+, @(R)+	X8 = -X8
3	Q+7	8R 9R 5R 78	ASHL R, @(R)+, (R)+	X7 = X7 2 <sup>X1'</sup>
4	Q+B	6R 7R 88	BISB2 -(R), (R)	X3 = X3 ∨ X2
5	Q+E	9R 96	INCB @(R)+	X2 = X2 + 1
6	Q+10	5R 7R D9	SBWC -(R), R	X9 = X9 - X8 - C
	Q+13	00	HALT	Останов

Используя рис. 1.1 и рис. 1.2, табл. 1.3, таблицу с кодами команд и начальный адрес программы Q можно составить «каркас» текста программы. Для алгоритма 1 такой «каркас» представлен в табл. 1.4. Из табл. 1.3 для первого варианта выбран второй столбец со способами адресации. При выборе кодов команд предполагалось, что операнды X1, X2, X3 имеют формат В, X4, X5, X6 – формат W, X7, X8, X9 – формат L. Далее необходимо выбрать регистры R для спецификаций операндов в табл. 1.4, реализовав заданные способы адресации.

#### 4. Порядок выполнения работы

*А. В процессе самостоятельной работы.*

1). Выбрать исходные данные в соответствии с номером варианта.

Например, для варианта (NV = 33, NGP = 7, NGL = 3) исходные данные имеют следующий вид:



$$\begin{aligned}
X1 &= (-1)^{NV} [(NV + NGP) * 3] = (-1)^{33} [(33 + 7) * 3] = -120; \\
X2 &= (-1)^{(NV+1)} [(NV + NGL) * 2] = (-1)^{(33+1)} [(33 + 3) * 2] = 72; \\
X3 &= (-1)^{(NV+NGL)} [NV + NGP + NGL + 21] = (-1)^{(33+3)} [33 + 7 + 3 + 21] = 64; \\
X4 &= (-1)^{NV} [NV + NGP + 29]^2 = (-1)^{33} [33 + 7 + 29]^2 = -4761; \\
X5 &= (-1)^{(NV+1)} [NV + NGL + 23]^2 = (-1)^{(33+1)} [33 + 3 + 23]^2 = 3481; \\
X6 &= (-1)^{(NV+NGL)} [NV + NGP + NGL + 79]^2 = (-1)^{(33+3)} [33 + 7 + 3 + 79]^2 = \\
&= 14884;
\end{aligned}$$

$$X7 = X4^2 = (-4761)^2 = 22667121;$$

$$X8 = -X5^2 = -3481^2 = -12117361;$$

$$X9 = (-1)^{(NV+NGL)} X6^2 = (-1)^{(33+3)} 14884^2 = 221533456.$$

2). Перевести значения величин X1-X9 в шестнадцатеричную систему счисления и выбрать минимальный формат для их представления.

Например, для варианта (NV = 33, NGP = 7, NGL = 3):

$$X1 = -120 = 88_{(16)};$$

$$X2 = 72 = 48_{(16)};$$

$$X3 = 64 = 40_{(16)};$$

$$X4 = -4761 = ED67_{(16)};$$

$$X5 = 3481 = D99_{(16)};$$

$$X6 = 14884 = 3A24_{(16)};$$

$$X7 = 22667121 = 159DF71_{(16)};$$

$$X8 = -12117361 = FF471A8F_{(16)};$$

$$X9 = 221533456 = D345510_{(16)}.$$

Учитывая диапазоны для форматов данных процессора VAX11 можно сказать, что для данного варианта, операнды X1-X3 должны быть представлены в формате байта, X4-X6 – в формате слова, а X7-X9 – в формате длинного слова.

3). Определить регистры и адреса ячеек памяти для размещения операндов, начальный адрес программы и адреса для размещения промежуточных ячеек. Все адреса должны быть переведены в шестнадцатеричную систему счисления.

Например, для варианта (NV = 33, NGP = 7, NGL = 3). Так как параметр NV превышает 20 (максимальное возможное число вариантов) в табл. 1.1 и 1.2, то выбираем из этих таблиц данные для варианта (33-20) = 13. По табл. 1.1 и 1.2 определяем, что операнд X2 расположен в регистре R3, операнд X5 – в регистре R2, а операнд X6 – в регистре RC. Остальные операнды размещены в памяти по следующим адресам:



$$\begin{aligned}\text{Адр}(\text{M1}) &= \text{Адр}(\text{X1}) = \text{NV} * \text{NGL} + \text{NGP} = 33 * 3 + 7 = 106 = 6\text{A}_{(16)}; \\ \text{Адр}(\text{M2}) &= \text{Адр}(\text{X3}) = \text{NV} * \text{NGL} + \text{NGP} + 10 = 33 * 3 + 7 + 10 = 116 = \\ &= 74_{(16)}; \\ \text{Адр}(\text{M3}) &= \text{Адр}(\text{X4}) = \text{NV} * \text{NGL} + \text{NGP} + 20 = 33 * 3 + 7 + 20 = 126 = \\ &= 7\text{E}_{(16)}; \\ \text{Адр}(\text{M4}) &= \text{Адр}(\text{X7}) = \text{NV} * \text{NGL} + \text{NGP} + 30 = 33 * 3 + 7 + 30 = 136 = \\ &= 88_{(16)}; \\ \text{Адр}(\text{M5}) &= \text{Адр}(\text{X8}) = \text{NV} * \text{NGL} + \text{NGP} + 40 = 33 * 3 + 7 + 40 = 146 = \\ &= 92_{(16)}; \\ \text{Адр}(\text{M6}) &= \text{Адр}(\text{X9}) = \text{NV} * \text{NGL} + \text{NGP} + 50 = 33 * 3 + 7 + 50 = 156 = \\ &= 9\text{C}_{(16)}.\end{aligned}$$

Начальный адрес размещения программы:

$$\text{Адр}(\text{PROGRAM}) = \text{NV} * 20 + \text{NGL} * 100 = 33 * 20 + 3 * 100 = 960 = 3\text{C0}_{(16)}.$$

Промежуточные ячейки должны быть размещены с адреса:

$$\text{Адр}(\text{TEMP}) = \text{NV} * \text{NGP} + 1000 = 33 * 7 + 1000 = 1231 = 4\text{CF}_{(16)}.$$

4). Выбрать алгоритм для реализации. Для варианта ( $\text{NV} = 33$ ,  $\text{NGP} = 7$ ,  $\text{NGL} = 3$ ) согласно заданию выбирается алгоритм 8 (остаток от деления 44 на 9). Пронумеруем в алгоритме операнды, размещенные в памяти, по порядку (справа налево и сверху вниз) (табл. 1.5). Операнды, размещенные в регистрах, не нумеруются. Определим способы адресации для операндов согласно табл. 1.3. Способы адресации операндов приведены в табл. 1.5 (по варианту выбран столбец 14 (остаток от деления 34 на 20) из табл. 1.3). Способ адресации для Оп10 не используется, в виду отсутствия этого операнда в данном алгоритме.

Таблица 1.5

Способы адресации операндов

Оператор	Способы адресации (СА) операндов			Комментарий
	Третий операнд	Второй операнд	Первый операнд	
1	--	--	8	Оп1 = 0
2	7	7	8	Оп4 = Оп3 * 2 Оп2
3	6	9	9	Оп7 = Оп6 - Оп5
4	--	5	6	X2 = X2 - Оп8
5	--	5	5	X6 = -X5
6	--	--	7	Оп9 = Оп9 - 1



Таблица 1.6

## Карта распределения памяти

Объект	Формат	Шестнадцатеричный код	Адрес загрузки
Число X1	B	88	6A
Число X2	B	48	R3
Число X3'	B	04	74
Число X4	W	ED67	7E
Число X5	W	0D99	R2
Число X6	W	3A24	RC
Число X7	L	0159DF71	88
Число X8	L	FF471A8F	92
Число X9	L	0D345510	9C
Адрес X1(CA=6)	L	0000006A	R8
Адрес X1(CA=7)	L	0000006B	R9
Адрес X4(CA=8)	L	0000007E	R0
Адрес X3(CA=8)	L	00000074	R1
Адрес X7(CA=7,6)	L	0000008C	R4
Адрес X9(CA=7)	L	000000A0	R5
Текст программы		Текст программы	3C0-3D2
Адрес ПЯ1(CA=9)	L	000004CF	R6
ПЯ1	L	0000009C	4CF
Адрес ПЯ2(CA=9)	L	000004D3	R7
ПЯ2	L	00000092	4D3

Таблица 1.7

## Текст программы

Оператор	Адрес	Шестнадцатеричный код	Мнемонический код	Комментарий
1	3C0	80 B4	CLRW (R0)+	X4=0
2	3C2	75 74 81 78	ASHL (R1)+, -(R4), -(R5)	X9=X7*2 <sup>X3'</sup>
3	3C6	64 97 96 C3	SUBL3 @(R6)+, @(R7)+, (R4)	X7=X8-X9
4	3CA	53 68 82	SUBB2 (R8), R3	X2=X2-X1
5	3CD	5C 52 AE	MNEG W R2, RC	X6=-X5
6	3D0	79 97	DECB -(R9)	X1=X1-1
7	3D2	00	HALT	Останов



5). Составить карту распределения памяти под команды и данные, сведя информацию п. 2 и 3 в специальную таблицу. Например, для варианта ( $NV=33$ ,  $NGP=7$ ,  $NGL=3$ ) карта распределения памяти приведена в табл. 1.6.

В карте распределения памяти представлено содержимое регистров R0, R1, R4-R9 и ПЯ, выбранных для хранения адресов при косвенных адресациях.

6). Составить программу заданного алгоритма в мнемонических и машинных кодах. Например, для варианта ( $NV=33$ ,  $NGP=7$ ,  $NGL=3$ ), текст программы приведен в табл. 1.7. Программа реализует алгоритм 8.

7). Произвести трассировку заданного алгоритма с использованием заданных исходных данных, при этом в таблице трассировки должны быть отражены также и значения информации в ячейках памяти и регистрах, используемых для реализации различных способов адресации. Результаты трассировки для варианта ( $NV=33$ ,  $NGP=7$ ,  $NGL=3$ ) представлены в табл. 1.8. Столбец «Результат в лаборатории» заполняется на лабораторных занятиях при выполнении программы студентом в присутствии преподавателя. В остальные столбцы заносятся результаты ручного счета. Результаты выполнения операций выделены в таблице трассировки жирным начертанием. Отметим, что в команде `SUBL3 @(R6)+,@(R7)+,(R4)` вместо регистра R7 можно было повторно использовать регистр R6, так как после выборки первого операнда в команде регистр R6 будет содержать адрес следующей ПЯ – 000004D3(16). В командах `SUBB2 (R8),R3` и `DECB -(R9)` для адресации операнда X1 используются разные регистры R8 и R9, так как при первом обращении к операнду необходимо использовать косвенную адресацию, а при втором – адресацию с автоуменьшением. Также следует обратить внимание, что в команде `ASHL (R1)+,-(R4),-(R5)` после выборки первого операнда содержимое R1 увеличивается только на 1, так как в команде сдвига первый операнд представлен в формате байта.

8). Оформить отчет по лабораторной работе. Содержание отчета представлено далее в тексте.

#### *Б. В учебной лаборатории.*

1). Осуществить запись операндов в регистры или в память в соответствии с заданием.

2). Выполнить запись адресов в регистры и в ячейки памяти для косвенной адресации.

3). Записать программу в память.



Таблица 1.8

## Результаты трассировки

Оператор	Операнд 1			Операнд 2			Операнд 3			Результат в лаборатории
	Объект	До операции	После операции	Объект	До операции	После операции	Объект	До операции	После операции	
1	7E R0	ED67 0000007E	0000 00000080							
2	74 R1	04 00000074	04 00000075	88 R4	0159DF71 0000008C	0159DF71 00000088	9C R5	0D345510 000000A0	159DF710 0000009C	
3	9C R6 4CF	159DF710 000004CF 0000009C	159DF710 000004D3 0000009C	92 R7 4D3	FF471A8F 000004D3 00000092	FF471A8F 000004D7 00000092	88 R4	0159DF71 00000088	E9A9237F 00000088	
4	6A R8	88 6A	88 6A	R3	48	C0				
5	R2	0D99	0D99	RC	3A24	F267				
6	6A R9	88 6B	87 6A							



- 4). Сохранить программу в файле на диске с расширением .v11.
- 5). Выполнить программу по шагам.
- 6). Проверить результаты выполнения программы, сравнивая их с результатами ручной трассировки алгоритма.

## 5. Содержание отчета

- 1). Титульный лист.
- 2). Текст задания.
- 3). Перевод исходных данных в шестнадцатеричную систему счисления.
- 4). Карта распределения памяти под команды и данные.
- 5). Алгоритм программы.
- 6). Текст программы в мнемонических и машинных кодах.
- 7). Таблица трассировки программы.
- 8). Выводы по лабораторной работе.