

ГУАП

КАФЕДРА № 42

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ \_\_\_\_\_  
ПРЕПОДАВАТЕЛЬ

ассистент		И.Д. Свеженин
_____ должность, уч. степень, звание	_____ подпись, дата	_____ инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

Представление данных в ЭВМ типа VAX-11.

по курсу: Архитектура ЭВМ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4329		Д.С. Шаповалова
		_____ подпись, дата	_____ инициалы, фамилия

Санкт-Петербург 2025

## 1. 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.

## 2. Задание:

1. Определить исходные данные в соответствии с вариантом
2. Перевести значения величин X1-X9 в 16 СС и выбрать минимальный формат для их представления
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.
4. Выбрать алгоритм и реализовать его.

### 3. Ход работы:

#### 3.1. Определение номера алгоритма:

Вычисляется как  $1 + (NV + NGL + NGP) \% 9$

Номер варианта =  $NV = 16$

Последняя цифра номера группы =  $NGL = 9$

Предпоследняя цифра номера группы =  $NGP = 2$

$$1 + (16 + 9 + 2) \% 9 = 1 + (27) \% 9 = 1 + 0 = 1$$

На рисунке 1 вариант алгоритма для варианта 16 =

$$\begin{aligned} X4 &= X5 + X6 \\ X8 &= -X8 \\ X7 &= X7 * 2^{X1'} \end{aligned}$$

Рисунок 1.1 – Алгоритм 1

$$X1' = X1 \% 10 \text{ (байт)}$$

#### 3.2. Определение исходных данных:

$$X1 = (-1)^{16} [(16+2)*3] = +(18*3) = +54 = 0x36 \text{ (16) (байт)}$$

$$X2 = (-1)^{(16+1)} [(16+9)*2] = -(25*2) = -50 = 0xCE \text{ (16) (байт)}$$

$$X3 = (-1)^{(16+9)} [16+2+9+21] = -48 = 0xD0 \text{ (16) (байт)}$$

$$X4 = (-1)^{(16)} [16+2+29]^2 = +2209 = 0x08A1 \text{ (16) (слово)}$$

$$X5 = (-1)^{(16+1)} [16+9+23]^2 = -2304 = 0xF700 \text{ (16) (слово)}$$

$$X6 = (-1)^{(16+9)} [16+2+9+79]^2 = -11236 = 0xD41C \text{ (16) (слово)}$$

$$X7 = (X4)^2 = +4879681 = 0x4A7541 \text{ (16) (длинное слово)}$$

$$X8 = (-X5)^2 = +5308416 = 0x510000 \text{ (16) (длинное слово)}$$

$$X9 = (-1)^{(16+9)} X6^2 = -126247696 = 0xF875E4F0 \text{ (16) (длинное слово)}$$

#### 3.3 Определение адресов размещения операндов

Данные	Варианты		
	16	2	3
G1	9	B	1
G2	A	C	3
G3	5	6	A

Рисунок 1.2 – Размещение данных в регистрах по вариантам

X6, X8, X9 размещаются в регистрах:

X6 в R9

X8 в RA

X9 в R5

Адреса в памяти:

$$\text{Адр}(M1) = NV * NGL + NGP = 16 * 9 + 2 = 146 = 92 \text{ (16)}$$

$$\text{Адр}(M2) = 146 + 10 = 156 = 9C \text{ (16)}$$

$$\text{Адр}(M3) = 166 = A6 \text{ (16)}$$

$$\text{Адр}(M4) = 176 = B0 \text{ (16)}$$

$$\text{Адр}(M5) = 186 = BA \text{ (16)}$$

$$\text{Адр}(M6) = 196 = C4 \text{ (16)}$$

Начальный адрес размещения:

$$\text{Адр}(\text{PROGRAM}) = NV * 20 + NGL * 100 = 16 * 20 + 9 * 100 = 320 + 900 = 1220 = 4C4 \text{ (16)}$$

### 3.4 Нумерация операндов и определение адресации

X5 – Оп1 – 6 – косвенная адресация

X4 – Оп2 – 9 – косвенная автоинкрементная адресация (двойная косвенная с автоувелечением)

X6 – в регистре, не нумеруется.

X8 – в регистре, не нумеруется.

X1 – Оп3 – 8 – автоинкрементная (косвенная с автоувелечением)

X7 – Оп4 – 7 – автодекрементная (косвенная с автоуменьшением)

Промежуточные Ячейки, для реализации косвенной адресации расположены, начиная с адреса  $\text{Адр}(\text{TEMP}) = NV * NGP + 1000 = 16 * 2 + 1000 = 32 + 1000 = 1032 = 408 \text{ (16)}$

### 3.5 Алгоритм в новой нумерации

$$\text{Оп2} = \text{Оп1} + X6$$

$$X8 = -X8$$

$$\text{Оп4} = \text{Оп4} * 2^{(\text{Оп3})}$$

В таблице 1 приведены способы адресации операндов в операторах.

Таблица 1 – Способы адресации операндов

Оператор	Способы адресации операндов			Комментарий
	Третий операнд	Второй операнд	Первый операнд	
1	-	9	6	Оп2 = Оп1+Х6
2	-	-	-	Х8 = -Х8
3	-	7	8	Оп4 = Оп4*2^(Оп3)'

В таблице 2 представлена карта распределения памяти.

Таблица 2 – Карта распределения памяти

Объект	Формат	Шестнадцатеричный код	Адрес загрузки
X1`	B	4	92
X2	B	CE	9C
X3	B	D0	A6
X4	W	08A1	B0
X5	W	F700	BA
X6	W	D41C	R9
X7	L	004A7541	C4
X8	L	00510000	RA
X9	L	F875E4F0	R5
Адр.(X5) 6	L	000000BA	R0
Адр.(X4) 9	L	000000B0	408
Адр.(X1) 8	L	00000092	R3
Адр.(X7) 7	L	000000C4	R4
Текст прог.		Текст программы	4C4-4CF
Адр. ПЯ1	L	00000408	R1
ПЯ1	L	000000B0	408

### 3.6 Текст программы

Таблица 3 – Текст программы

Оператор	Адрес	16-ричный код	Мнемонический код	Комментарий
1	4C4	A1 60 59 91	ADDW3 (R0), R9, @(R1)+	X4 = X5 +X6
2	4C8	CE 5A 5A	MNEGL RA, RA	X8 = -X8
3	4CB	78 83 74 74	ASHL -(R3), (R4)+, (R4)+	X7 = X7*2^(X1)
4	4CF	00	HALT	Останов

```
Описание выполнения команды:
Адрес:      000004C4
Код команды:  A1
Мнемоника:   ADDW3
Число операндов: 3
Описание операнда №1:
Адрес:      000000BA
Размер:     слово (2 байта)
Значение:   F700
Описание операнда №2:
Адрес:      в регистре №9
Размер:     слово (2 байта)
Значение:   D41C
Описание операнда №3:
Адрес:      000000B0
--- До операции ---
Размер:     слово (2 байта)
Значение:   08A1
--- После операции ---
Размер:     слово (2 байта)
Значение:   CB1C
```

Рисунок 2.1 – Первый оператор

```
Описание выполнения команды:
Адрес:      000004C8
Код команды:  CE
Мнемоника:   MNEGL
Число операндов: 2
Описание операнда №1:
Адрес:      в регистре №10
Размер:     длинное слово (4 байта)
Значение:   00510000
Описание операнда №2:
Адрес:      в регистре №10
--- До операции ---
Размер:     длинное слово (4 байта)
Значение:   00510000
--- После операции ---
Размер:     длинное слово (4 байта)
Значение:   FFAF0000
```

Рисунок 2.2 – Второй оператор

```
Описание выполнения команды:
Адрес:      000004CB
Код команды: 78
Мнемоника:   ASHL
Число операндов: 3
Описание операнда №1:
Адрес:      00000092
Размер:     байт
Значение:   04
Описание операнда №2:
Адрес:      000000C4
Размер:     длинное слово (4 байта)
Значение:   004A7541
Описание операнда №3:
Адрес:      000000C0
--- До операции ---
Размер:     длинное слово (4 байта)
Значение:   00000000
--- После операции ---
Размер:     длинное слово (4 байта)
Значение:   04A75410
```

Рисунок 2.3 – Третий оператор

#### 4. Таблица трассировки программы

Таблица 4 – Трассировка программы

Операнд		Оператор		
		1	2	3
1	Объект	R0 000000BA	RA	R3 00000092
	До операции	F700	00510000	4
	После операции	F700	FFAF0000	4
2	Объект	R9	-	R4 000000C4
	До операции	D41C	-	04A75410
	После операции	D41C	-	04A75410
3	Объект	R1 00000408 000000B0	-	000000C0
	До операции	08A1	-	00000000
	После операции	CB1C	-	04A75410
Результат		X5+X6=X4= CB1C	-(00510000)=FAF0000	04A75410



## **ВЫВОД**

В ходе выполнения лабораторной работы №5 были освоены ключевые принципы программирования на ассемблере ЭВМ VAX-11. Особое внимание уделено корректному применению различных способов адресации (6 — косвенная регистровая, 7 — автоинкрементная, 8 — автодекрементная, 9 — косвенная автоинкрементная), что позволило гибко организовать доступ к операндам, размещённым как в регистрах, так и в памяти.

Были выполнены все этапы подготовки программы: преобразование исходных данных с учётом форматов (байт, слово, длинное слово) и представления отрицательных чисел в дополнительном коде, распределение переменных по памяти и регистрам, а также создание промежуточных ячеек для реализации косвенной адресации. На основе этого составлены мнемонические команды, которые затем переведены в машинные коды в соответствии с форматами VAX-11.

Программа, реализующая заданный алгоритм, была успешно отлажена в симуляторе, что подтвердило правильность понимания архитектуры процессора и особенностей адресации в VAX-11.