

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
Пермский национальный исследовательский политехнический университет  
(ПНИПУ)

Факультет: Электротехнический (ЭТФ)  
Направление: 09.03.04 – Программная инженерия (ПИ)  
Профиль: Разработка программно-информационных систем (РИС)  
Кафедра информационных технологий и автоматизированных систем (ИТАС)

УТВЕРЖДАЮ

Зав. кафедрой ИТАС: д-р экон. наук, проф.

Р.А. Файзрахманов  
« 17 » 12 2025 г.

**КУРСОВАЯ РАБОТА**

по дисциплине

«Организация ЭВМ и систем»

на тему

**«Структурно-алгоритмическое проектирование ЭВМ»**

Студент 17.12.25 Гордеев Василий Андреевич  
(подпись, дата)

Группа: РИС-23-36

Оценка отлично

Руководитель КР:

17.12.25  
(подпись, дата)

к.т.н., доц. каф. ИТАС Погудин А.Л.

Пермь 2025

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
Пермский национальный исследовательский политехнический университет  
(ПНИПУ)

Факультет: Электротехнический (ЭТФ)  
Направление: 09.03.04 – Программная инженерия (ПИ)  
Профиль: Разработка программно-информационных систем (РИС)  
Кафедра информационных технологий и автоматизированных систем (ИТАС)

УТВЕРЖДАЮ

Зав. кафедрой ИТАС д-р экон. наук, проф.

Р.А. Файзрахманов

« 06 » 2025 г.

**ЗАДАНИЕ**  
**на выполнение курсовой работы**

Фамилия, имя, отчество: Гордеев Василий Андреевич

Факультет Электротехнический Группа РИС-23-36

Начало выполнения работы: 06.10.2025

Контрольные сроки просмотра работы: 27.10, 11.11, 02.12

Защита работы: 17.12.2025

1. Наименование темы: «Структурно-алгоритмическое проектирование ЭВМ».

2. Исходные данные к работе (проекта):

Объект исследования – Арифметико-логическое устройство (АЛУ).

Предмет исследования – Алгоритм работы и структура АЛУ.

Цель работы (проекта) – Разработать АЛУ реализующее, заданный набор операций с учётом на код выполнения операций и способ построения выполняющего автомата. Тип управляющего автомата – автомат Мили, выполняющий три команды (сложения, вычитания, конъюнкции). Флаги OV, Z; КОД ДК.

3. Содержание:


- 3.1 Исследование предметной области курсовой работы
- 3.2 Анализ исходных данных задания на курсовую работу
- 3.3 Спецификация устройства на уровне «черного ящика»
- 3.4 Представление устройства в виде операционной и управляющей частей
- 3.5 Разработка структуры устройства
- 3.6 Разработка граф-схемы работы устройства (на уровне автомата Мили)
- 3.7 Составление алгоритма работы устройства (на уровне команд и микрокоманд).
- 3.8 Составление полной спецификации устройства
- 3.9 Разработка микропрограммы работы устройства
- 3.10 Разработка временной диаграммы работы устройства
- 3.11 Контрольный пример в модели АЛУ (учебной ЭВМ)

Руководитель КР:

  
(подпись, дата)

к.т.н., доц. каф. ИТАС Погудин А.Л.

Задание получил:

06.10.25   
(подпись, дата)

В. А. Гордеев



КАЛЕНДАРНЫЙ ГРАФИК ВЫПОЛНЕНИЯ  
КУРСОВОЙ РАБОТЫ

№ пп	Этапы работы	Объём этапа, %	Сроки выполнения	
			Начало	Конец
1.	Исследование предметной области	10	06.10.25	07.10.25
2.	Устройство управления	5	08.10.25	11.10.25
3.	Сложение	5	12.10.25	13.10.25
4.	Вычитание	5	14.10.25	15.10.25
5.	Сравнение	5	16.10.25	18.10.25
6.	Логическое «И»	5	19.10.25	21.10.25
7.	Адресация	5	22.10.25	25.10.25
8.	Разработка устройства.	5	26.10.25	27.10.25
9.	Анализ исходных данных задания на курсовую работу	5	28.10.25	30.10.25
10.	Спецификация устройства на уровне «черного ящика»	5	31.10.25	03.11.25
11.	Представление черного ящика в виде операционной и управляющей частей	5	04.11.25	05.11.25
12.	Разработка структуры операционной части устройства	5	06.11.25	09.11.25
13.	Составление схемы алгоритма работы устройства и его микропрограммы	5	10.11.25	11.11.25
14.	Разработка схемы алгоритма работы	5	12.11.25	15.11.25
15.	Составление полной спецификации устройства	5	16.11.25	18.11.25
16.	Разработка фрагмента функциональной схемы вычитателя	5	19.11.25	22.11.25
17.	Контрольный пример	5	23.11.25	28.11.25
18.	Временная диаграмма работы УУ	5	29.11.25	02.12.25
19.	Оформление курсовой работы	5	03.12.25	16.12.25
20.	Защита курсовой работы		17.12.25	17.12.25

Руководитель КР:

06.10.25  
(подпись, дата)

к.т.н., доц. каф. ИТАС Погудин А.Л..

Задание получил:

06.10.25  
(подпись, дата)

Гордеев Василий Андреевич

## РЕФЕРАТ

Отчет 22 с., 7 рис., 11 табл., 3 источника.

АРИФМЕТИКО-ЛОГИЧЕСКОЕ УСТРОЙСТВО (АЛУ),  
МИКРОПРОГРАММНОЕ УПРАВЛЕНИЕ, АВТОМАТ МИЛИ, ПРЯМОЙ  
КОД, ДОПОЛНИТЕЛЬНЫЙ КОД, СЛОЖЕНИЕ, ВЫЧИТАНИЕ.

Цель работы – спроектировать и реализовать арифметико-логическое устройство для выполнения операций сложения, вычитания и конъюнкции над восьмиразрядными числами. Особенностью разработки является реализация алгоритма обработки знаковых чисел: входные данные поступают в прямой коде, вычисления производятся в дополнительном коде, а результат преобразуется обратно в прямой код.

При разработке устройства использовалась концепция «черного ящика» для определения внешних интерфейсов. Синтез управляющего устройства выполнен на основе модели автомата Мили с жесткой логикой. Для реализации алгоритма применена стратегия микропрограммного управления, позволяющая гибко настраивать последовательность управляющих сигналов.

В результате работы разработаны функциональная схема, граф переходов управляющего автомата и временные диаграммы выполнения операций. Составлены таблицы прошивки памяти микрокоманд.

Работоспособность устройства и корректность алгоритмов подтверждена результатами моделирования в среде ALU-R.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	7
1 Исследовательская часть .....	8
1.1 Исследование предметной области .....	8
1.1.1 Прямой, обратный и дополнительный коды .....	8
1.1.2 Особенности алгоритма обработки данных .....	9
1.1.3 Флаги и признаки результата.....	9
1.2 Анализ исходных данных задания .....	9
1.3 Представление устройства в виде операционной и управляющей частей .....	10
1.4 Структура устройства АЛУ-R.....	10
1.4.1 Управление входными шинами (Поля 1a и 1b).....	11
1.4.2 Арифметико-Логический Блок (АЛБ) .....	11
1.4.3 Управление сдвигом и регистром результата .....	11
1.4.4 Управление счетчиком .....	12
1.4.5 Логические условия .....	12
1.5 Краткие сведения об управляющих автоматах .....	12
2 Проектная часть.....	13
2.1 Разработка модели «Черного ящика» АЛУ .....	13
2.2 Разработка алгоритмов функционирования .....	14
2.2.1 Алгоритм функционирования на уровне команд .....	15
2.2.2 Синтез графа УА Мили на микрокомандном уровне.....	15
2.3 Временные диаграммы выполнения микроопераций .....	18
2.3.1 Диаграмма операции Сложения .....	18
2.3.2 Диаграмма операции Вычитания.....	18
2.3.3 Диаграмма операции Конъюнкции .....	19
3 Экспериментальная часть.....	20
3.1 Распределение памяти микрокоманд .....	20
3.2 Программная реализация АЛУ .....	21
3.2.1 Блок операции Сложения .....	21
3.2.2 Блок операции Вычитания .....	21
3.2.3 Блок операции Конъюнкции .....	22
3.3 Контрольные примеры в модели АЛУ .....	23

ЗАКЛЮЧЕНИЕ .....	24
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	25

## ВВЕДЕНИЕ

Арифметико-логическое устройство (АЛУ) является центральным узлом любого процессора, отвечающим за преобразование информации. Именно в АЛУ выполняются все арифметические вычисления (сложение, вычитание) и логические операции над данными. Понимание принципов построения АЛУ, методов кодирования чисел и организации управляющих автоматов является базисом для проектирования вычислительных систем.

Актуальность работы обусловлена необходимостью глубокого понимания процессов, происходящих в ЭВМ на уровне регистровых передач. В современных системах операции часто выполняются над числами в дополнительном коде для унификации схмотехники, однако интерфейс с внешним миром обычно требует представления данных в прямом коде.

Целью курсовой работы является проектирование арифметико-логического устройства, выполняющего заданный набор операций, и его программная реализация в среде моделирования ALU-R.

Для достижения цели решаются следующие задачи:

- 1) Исследовательский этап: Изучить теоретические основы машинной арифметики (прямой, обратный, дополнительный коды) и функциональные возможности модели ALU-R согласно учебному пособию А.П. Жмакина.
- 2) Проектный этап: Разработать модель «черного ящика», составить графы-схемы алгоритмов (ГСА) на уровне команд и микрокоманд, построить временную диаграмму выполнения операций.
- 3) Экспериментальный этап: Реализовать микропрограммный автомат в модели ALU-R, строго соблюдая адресацию микрокоманд, и протестировать работоспособность устройства.

## 1 Исследовательская часть

### 1.1 Исследование предметной области

#### 1.1.1 Прямой, обратный и дополнительный коды

В соответствии с классической теорией организации ЭВМ [1], выделяют следующие формы представления целых чисел со знаком.

##### Прямой код (ПК)

Прямой код является формой представления числа, в которой старший разряд ( $a_n$ ) кодирует знак числа (0 — для положительных, 1 — для отрицательных), а оставшиеся разрядов ( $a_{n-1}...a_0$ ) содержат абсолютное значение (модуль) числа.

Аналитическое выражение для прямого кода:

$$[A]_{\text{пр}} = 0, |A|, \text{ при } A \geq 0$$

$$[A]_{\text{пр}} = 1, |A|, \text{ при } A < 0$$

К недостаткам прямого кода следует отнести сложность аппаратной реализации арифметических операций, требующую анализа знаковых разрядов и выполнения операций вычитания модулей. Кроме того, в данной системе кодирования существует двойственность представления нуля («+0» и «-0»).

##### Обратный код (ОК)

Обратный код применяется преимущественно как промежуточный этап при формировании дополнительного кода.

Для положительных чисел обратный код идентичен прямому.

Для отрицательных чисел обратный код формируется путем поразрядной инверсии цифровой части прямого кода при сохранении значения знакового разряда равным единице.

$$[A]_{\text{обр}} = [A]_{\text{пр}}, \text{ при } A \geq 0$$

$$[A]_{\text{обр}} = 1, (\text{инверсия } a_{n-1}...a_0), \text{ при } A < 0$$

Спецификой арифметики в обратном коде является необходимость циклического переноса единицы переполнения из старшего разряда в младший, что усложняет схемотехнику сумматора и увеличивает время выполнения операции.

##### Дополнительный код (ДК)

Дополнительный код является основным стандартом для выполнения целочисленных операций в современных вычислительных системах.

Для положительных чисел дополнительный код совпадает с прямым.



Для отрицательных чисел дополнительный код образуется из обратного кода путем добавления единицы к младшему разряду.

Ключевым преимуществом дополнительного кода является возможность сведения операции вычитания к операции сложения с отрицательным числом. При этом знаковый разряд участвует в операции наравне с цифровыми, а перенос из старшего разряда игнорируется.

### **1.1.2 Особенности алгоритма обработки данных**

В рамках проектируемого устройства реализуется специфический алгоритм обработки данных. На вход АЛУ поступают операнды исключительно в прямом коде (положительные числа). Для унификации вычислительного процесса применяется метод предварительного преобразования операндов в отрицательное представление в дополнительном коде.

Преобразование входных данных: Исходные положительные операнды А и В преобразуются в отрицательные числа в дополнительном коде по формуле:  $-X = (\text{инверсия } X) + 1$ .

Выполнение арифметической операции: Производится сложение или вычитание полученных отрицательных значений.

Восстановление результата: Полученный результат (находящийся в дополнительном коде) преобразуется обратно в прямой код положительного числа путем инверсии и инкремента.

### **1.1.3 Флаги и признаки результата**

В процессе функционирования АЛУ формируются статусные сигналы (флаги), характеризующие результат операции:

- 1) Z (Zero): Признак нулевого результата. Принимает значение логической единицы, если все биты результата равны нулю.
- 2) OV (Overflow): Признак арифметического переполнения. Сигнализирует о выходе результата операции за пределы допустимого диапазона представления чисел.
- 3) P (Carry/Borrow): Признак переноса или заема. Используется при выполнении операций вычитания для определения знака разности и необходимости коррекции результата.

## **1.2 Анализ исходных данных задания**

В соответствии с заданием на курсовую работу, объектом проектирования является Арифметико-Логическое Устройство (АЛУ). Основные характеристики проектируемого устройства:

Разрядность: 8 бит.

Формат данных: Целые числа со знаком. Входные и выходные данные представлены в прямом коде. Внутренняя обработка осуществляется в дополнительном коде, что требует реализации алгоритмов преобразования (инверсия + инкремент).

Набор операций:

- Алгебраическое сложение;
- Алгебраическое вычитание;
- Логическая конъюнкция.

Управляющий автомат: Автомат Мили с жесткой логикой.

Контроль состояния: Формирование флагов переполнения (OV) и нуля (Z).

### **1.3 Представление устройства в виде операционной и управляющей частей**

Согласно принципам структурно-алгоритмического проектирования, АЛУ декомпозируется на два взаимодействующих автомата:

Операционный автомат (ОА): Структурная часть, выполняющая хранение и преобразование информации. Включает в себя:

- 1) Регистры приема операндов (R, S) и результата (F);
- 2) Комбинационную схему сумматора (АЛБ);
- 3) Схему сдвигателя;

Мультиплексоры входной логики. ОА выполняет элементарные действия (микрооперации)  $u_i$  и вырабатывает логические условия  $x_j$  (признаки результата).

Управляющий автомат (УА): Логическая схема, координирующая работу ОА. На основе кода операции и значений логических условий  $x_j$ , УА вырабатывает во времени последовательность управляющих сигналов  $Y_t$ , обеспечивающую выполнение заданного алгоритма. В данной работе УА реализуется микропрограммным способом.

### **1.4 Структура устройства АЛУ-R**

Для проектирования микропрограммного автомата необходимо определить полный набор микроопераций, поддерживаемых моделью ALU-R. Ниже приведен детальный перечень управляющих сигналов и соответствующих им функциональных действий.

### 1.4.1 Управление входными шинами (Поля 1a и 1b)

Блок входной логики управляет мультиплексорами, подающими данные на входы сумматора R и S.

Поле 1a (Управление входом R):

00:  $R := A$  (Прямая передача операнда A)

01:  $R := \neg A$  (Передача инверсного значения A)

10:  $R := 0$  (Обнуление входа)

11:  $R := a_7, \neg a_6 \dots \neg a_0$  (Передача специфической маски)

Поле 1b (Управление входом S):

00:  $S := B$  (Прямая передача операнда B)

01:  $S := \neg B$  (Передача инверсного значения B)

10:  $S := 0$  (Обнуление входа)

11:  $S := b_7, \neg b_6 \dots \neg b_0$  (Передача специфической маски)

### 1.4.2 Арифметико-Логический Блок (АЛБ)

Код микрооперации АЛБ определяет функцию, выполняемую комбинационной схемой сумматора. Результат операции обозначается как Q.

000:  $Q := R + S + P_0$  (Сложение с учетом входного переноса P0)

001:  $Q := R - S - P_0$  (Вычитание с учетом заема)

010:  $Q := S - R - P_0$  (Обратное вычитание)

011:  $Q := R \vee S$  (Логическое ИЛИ / Дизъюнкция)

100:  $Q := R \& S$  (Логическое И / Конъюнкция)

101:  $Q := R \oplus S$  (Сложение по модулю 2)

110:  $Q := R$  (Транзит значения R на выход)

111:  $Q := S$  (Транзит значения S на выход)

### 1.4.3 Управление сдвигом и регистром результата

Результат с выхода АЛБ (Q) проходит через схему сдвигателя перед записью в регистр результата (F).

000:  $F := Q$  (Запись результата без сдвига)

001: L1:  $DL.F[7:0] := Q[7:0].DR$  (Сдвиг влево на 1 разряд, прием бита из DR)

010: R1:  $F[7:0].DR := DL.Q[7:0]$  (Сдвиг вправо на 1 разряд, прием бита из DL)

011:  $F.DR := DR.Q$  (Циклический сдвиг)

100:  $F := Q[6:0].z$  (где  $z = P8 \text{ xor } DL$ ) — Сдвиг с учетом флага переноса

111:  $F := C$  (Запись в регистр F содержимого счетчика)

#### 1.4.4 Управление счетчиком

Счетчик используется для организации циклов в микропрограммах.

00:  $сч := сч$  (Режим хранения, значение не меняется)

01:  $сч := сч + 1$  (Инкремент)

10:  $сч := сч - 1$  (Декремент)

11:  $сч := 0$  (Сброс счетчика)

#### 1.4.5 Логические условия

Для реализации условных переходов в микропрограмме используются следующие признаки (флаги):

X1: Перенос P8 (Перенос из старшего разряда сумматора)

X2: DL (Значение входного бита слева)

X3: DR (Значение входного бита справа)

X4:  $сч > 7$  (Проверка значения счетчика)

X5:  $сч < 0$  (Переполнение счетчика)

### 1.5 Краткие сведения об управляющих автоматах

Управляющий автомат (УА) представляет собой дискретное устройство, предназначенное для координации работы операционных узлов ЭВМ путем выдачи распределенной во времени последовательности управляющих сигналов.

В теории цифровых автоматов выделяют два основных класса УА [1]:

- 1) Автомат с жесткой логикой: Закон функционирования определяется структурой комбинационной схемы и связями между триггерами памяти. Изменение алгоритма работы такого автомата требует физического изменения схемы.
- 2) Автомат с программируемой логикой (микропрограммный): Алгоритм управления хранится в запоминающем устройстве в виде микропрограммы. Каждая ячейка памяти содержит код микрокоманды, определяющий набор активных управляющих сигналов и адрес следующей микрокоманды. Такой подход обеспечивает гибкость и универсальность управления, позволяя модифицировать алгоритмы без изменения аппаратуры.

## 2 Проектная часть

Целью проектного этапа является разработка функциональной и алгоритмической структуры арифметико-логического устройства. В данном разделе определяются интерфейсы взаимодействия АЛУ с внешней средой, разрабатываются алгоритмы выполнения операций на уровне микрокоманд и строится временная диаграмма функционирования устройства.

### 2.1 Разработка модели «Черного ящика» АЛУ

На этапе структурного проектирования арифметико-логическое устройство рассматривается как «черный ящик» — функционально законченный модуль, имеющий определенный набор входных и выходных портов. Внутренняя реализация на данном этапе скрыта, внимание уделяется интерфейсу взаимодействия с внешней средой (шиной данных и устройством управления).

Схема модели «Черного ящика» представлена на рисунке 2.1.

Входные информационные сигналы:

- Шина А [7:0]: 8-разрядный вход для первого операнда. Данные поступают в прямом коде.
- Шина В [7:0]: 8-разрядный вход для второго операнда. Данные поступают в прямом коде.

Входные управляющие сигналы:

- Шина управления (Y): Совокупность сигналов микропрограммного управления, определяющих действие АЛУ в текущем такте. Включает поля управления мультиплексорами (1a, 1b), код операции АЛБ, сигналы сдвига и управления счетчиком.
- CLK (Clock): Сигнал синхронизации (тактовый импульс), инициирующий выполнение микроопераций.

Выходные сигналы:

- Шина F [7:0]: 8-разрядный выход результата операции. Результат выдается в прямом коде.

Статусные сигналы (Флаги):

- Z (Zero): Признак равенства результата нулю.
- P8 (Carry/Overflow): Флаг переноса из старшего разряда, используемый также для диагностики переполнения и управления ветвлением при вычитании.

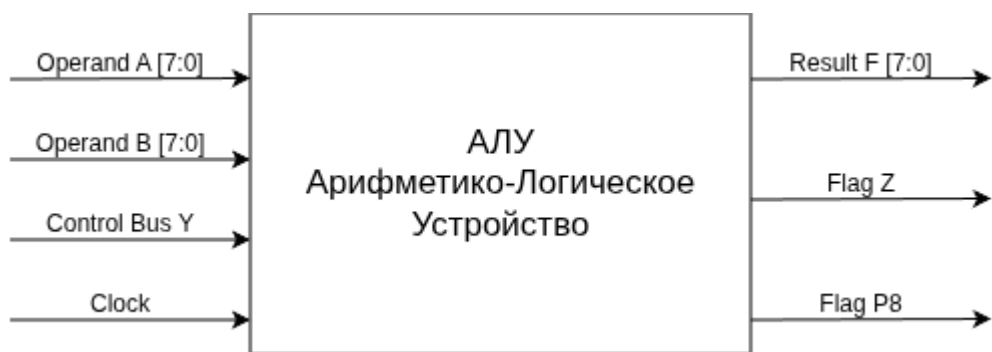


Рисунок 2.1 — Модель «Черного ящика» АЛУ

Таблица 2.1 — Описание интерфейсных сигналов

Обозначение	Разрядность	Тип	Назначение
A	8 бит	Вход	Первый операнд (Прямой код)
B	8 бит	Вход	Второй операнд (Прямой код)
Y	12 бит	Вход	Вектор управления (1a, 1b, ALB, Shift, Count)
Clock	1 бит	Вход	Синхронизация
F	8 бит	Выход	Результат операции (Прямой код)
Z	1 бит	Выход	Признак нуля (F=0)
P8	1 бит	Выход	Перенос из 7-го разряда / Признак заема

## 2.2 Разработка алгоритмов функционирования

Основной задачей проектирования является синтез алгоритмов, позволяющих реализовать сложные арифметические операции (сложение и вычитание в дополнительном коде) на базе ограниченного набора микроопераций модели ALU-R.



### 2.2.1 Алгоритм функционирования на уровне команд

На макроуровне алгоритм работы АЛУ состоит из трех независимых ветвей, выбор которых определяется пользователем (через адрес начала микропрограммы).

Ветвь СЛОЖЕНИЕ:

- 1) Прием операндов А и В.
- 2) Преобразование А в отрицательное число в дополнительном коде (-А).
- 3) Преобразование В в отрицательное число в дополнительном коде (-В).
- 4) Суммирование отрицательных операндов.
- 5) Преобразование результата из дополнительного кода обратно в прямой (восстановление модуля и знака).

Ветвь ВЫЧИТАНИЕ:

- 1) Формирование отрицательных операндов -А и -В.
- 2) Выполнение пробного вычитания (-А) - (-В).
- 3) Анализ флага переноса Р8.
- 4) Если результат положителен (в терминах дополнительного кода), выполняется переход к завершению и конвертации.
- 5) Если результат отрицателен (возник заем), выполняется переход к альтернативной ветви для коррекции (обратное вычитание S - R).
- 6) Преобразование результата в прямой код.

Ветвь КОНЪЮНКЦИЯ:

- 1) Поразрядное логическое умножение А & В.

### 2.2.2 Синтез графа УА Мили на микрокомандном уровне

Для реализации разработанного алгоритма функционирования АЛУ выполнен синтез управляющего автомата (УА) по модели Мили. Функционирование автомата описывается ориентированным связным графом, где вершины соответствуют состояниям автомата, а дуги — переходам между ними.

В проектируемой системе принято взаимно однозначное соответствие между состоянием автомата  $a_i$  и адресом ячейки памяти микрокоманд. Кодирование состояний выполнено прямым двоичным кодом (указан внутри вершин графа).

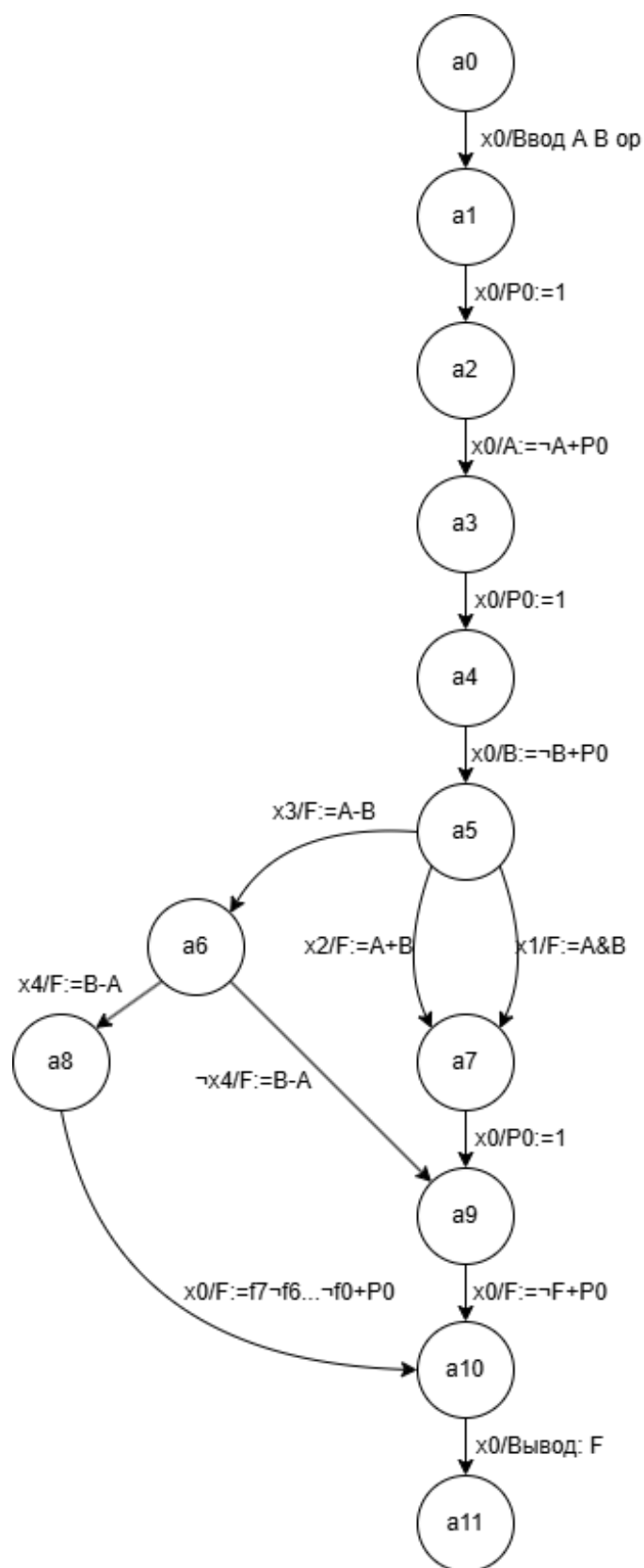


Рисунок 2.2 — Граф переходов управляющего автомата Мили

На дугах графа используется нотация  $X / Y$ , где:

$X$  — логическое условие перехода

$Y$  — выполняемая последовательность операций

Таблица 2.2 — Условия переходов

x0	1
x1	$op = 0$
x2	$op = 1$
x3	$op = 2$
x4	$F < 0$

Для перевода автомата мили в граф-схему, разделим составные условия.

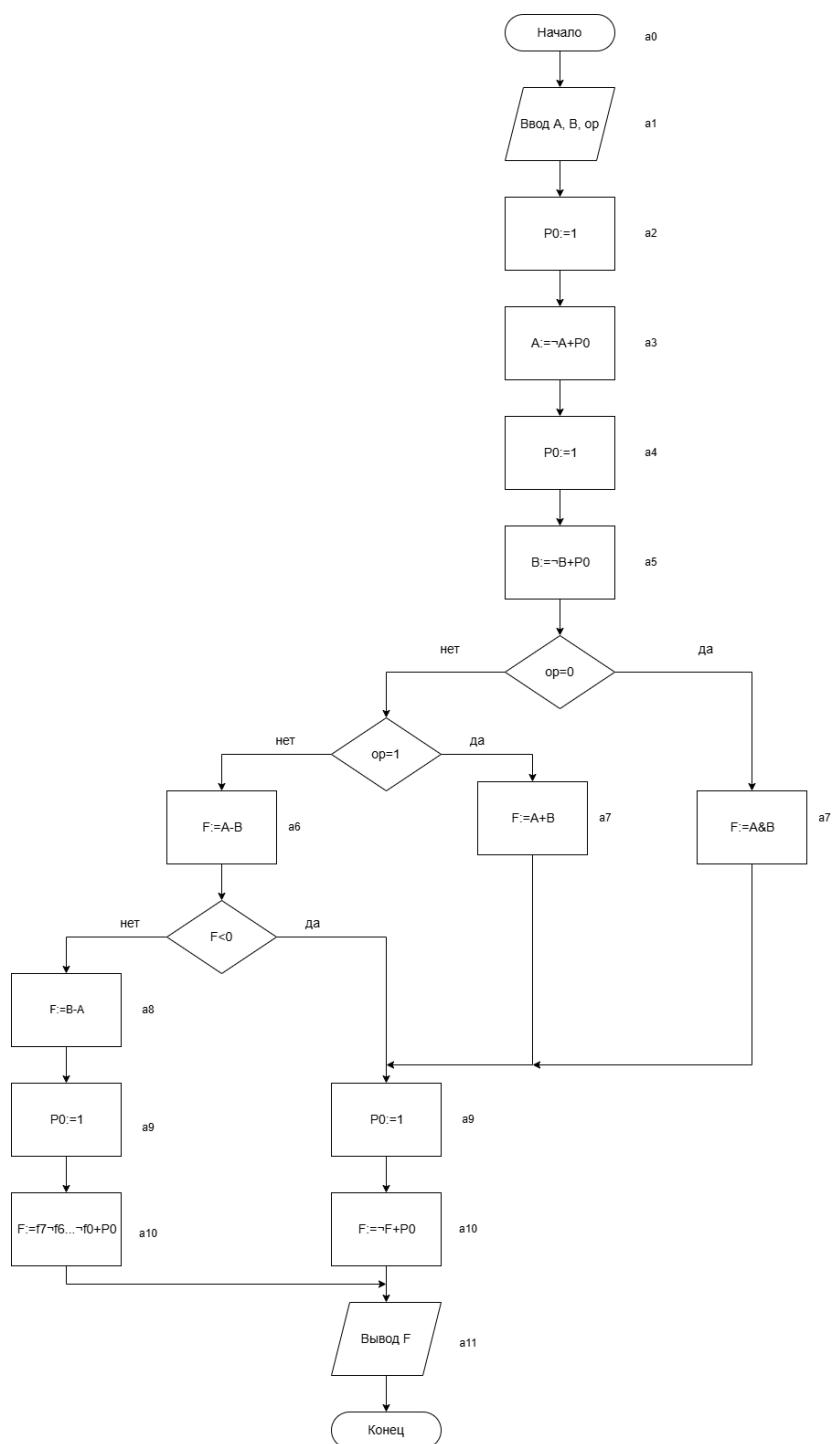


Рисунок 2.3 — Блок схема на уровне микрокоманд

## 2.3 Временные диаграммы выполнения микроопераций

Для анализа быстродействия и порядка работы устройства построены временные диаграммы (циклограммы) для всех реализованных операций. По оси ординат отложены выполняемые микрооперации, по оси абсцисс — такты машинного времени

### 2.3.1 Диаграмма операции Сложения

Операция является линейной и выполняется за фиксированное время — 7 тактов.

Таблица 2.3 — Временная диаграмма операции Сложения

Микрооперация \ Такт	T1	T2	T3	T4	T5	T6	T7
$P0 := 1$	■		■			■	
$R := \neg A$		■					
$S := \neg B$				■			
$Q := R + S$					■		
$R := \neg F$ (Восст.)							■

### 2.3.2 Диаграмма операции Вычитания

Операция содержит условный переход, поэтому время выполнения зависит от входных данных. Диаграмма совмещает две ветви алгоритма.

Таблица 2.4 — Временная диаграмма операции Вычитания

Микрооперация \ Такт	T1	T2	T3	T4	T5	T6	T7	T8
$P0 := 1$	■		■			■	■	
$R := \neg A$		■						
$S := \neg B$				■				

$Q := R - S$								
$Q := S - R$								
$R := \neg F$								
$S := f7 \neg f6 \dots \neg f1$								

Условные обозначения:

- Общая часть (Такты 1-5). (Цвет: Черный)
- Ветвь 1: результат положительный в операции  $Q := R - S$  (Такты 6-7). (Цвет: Зеленый)
- Ветвь 2: результат отрицательный в операции  $Q := R - S$ , был перенос (Такты 6-8). (Цвет: Синий)

### 2.3.3 Диаграмма операции Конъюнкции

Логическая операция выполняется за один машинный такт.

Таблица 2.5 — Временная диаграмма операции Конъюнкции

Микрооперация \ Такт	T1
$Q := R \& S$	

### 3 Экспериментальная часть

Целью экспериментального этапа является верификация разработанных алгоритмов в среде моделирования ALU-R. В данном разделе приведена карта распределения памяти микрокоманд и листинги программ, адресация которых строго соответствует синтезированному графу автомата Мили.

#### 3.1 Распределение памяти микрокоманд

На основе синтезированного графа автомата составлена карта памяти. Адреса ячеек памяти однозначно соответствуют двоичным кодам состояний, указанным в вершинах графа.

Таблица 3.1 — Карта памяти микрокоманд

Адреса (DEC)	Адреса (BIN)	Назначение блока	Примечание
0 — 6	0000000 — 0000110	СЛОЖЕНИЕ	Полный цикл операции
7 — 8	0000111 — 0001000	—	Резерв (Пустые ячейки)
9 — 13	0001001 — 0001101	ВЫЧИТАНИЕ (Начало)	Подготовка и сравнение
14 — 15	0001110 — 0001111	ВЫЧИТАНИЕ (Ветвь 1)	Результат положительный (Нет переноса)
16	0010000	—	Резерв
17 — 19	0010001 — 0010011	ВЫЧИТАНИЕ (Ветвь 2)	Результат отрицательный (Есть перенос)
20 — 21	0010100 — 0010101	—	Резерв
22	0010110	КОНЪЮНКЦИЯ	Логическая операция



## 3.2 Программная реализация АЛУ

### 3.2.1 Блок операции Сложения

Используются адреса с 0 по 6. Алгоритм: перевод операндов в дополнительный код, сложение, восстановление результата.

Таблица 3.2 — Микропрограмма операции Сложения

Адрес	A	B	F	1a	1b	АЛБ	Сдв	Сч	ЛУ	i	Адрес перехода	Комментарий
0000000	00	01	00	00	-	111	000	00	000	0	0000000	P0 := 1
0000001	000	001	010	01	10	000	000	00	000	0	0000000	R := ¬A
0000010	00	01	00	00	-	111	000	00	000	0	0000000	P0 := 1
0000011	000	001	011	10	01	000	000	00	000	0	0000000	S := ¬B
0000100	010	011	100	00	00	000	000	00	000	0	0000000	Q := R + S
0000101	00	01	00	00	-	111	000	00	000	0	0000000	P0 := 1
0000110	100	100	101	01	10	000	000	00	000	0	0000000	R := ¬F (Восст.)

### 3.2.2 Блок операции Вычитания

Начало программы расположено по адресу 0001001 (9). В ячейке 0001101 (13) происходит проверка флага переноса P8. Если P8=1, выполняется переход на адрес 0010001 (17).

Таблица 3.3 — Микропрограмма Вычитания (Основной ствол)

Адрес	A	B	F	1a	1b	АЛБ	Сдв	Сч	ЛУ	i	Адрес перехода	Комментарий
0001001	00	01	00	00	-	111	000	00	000	0	0000000	P0 := 1
0001010	000	001	010	01	10	000	000	00	000	0	0000000	R := ¬A
0001011	00	01	00	00	-	111	000	00	000	0	0000000	P0 := 1

0001100	000	001	011	10	01	000	000	00	000	0	0000000	$S := \neg B$
0001101	010	011	100	00	00	001	000	00	001	1	0010001	$Q := R - S$ (Если $P8=1 \Rightarrow \text{Jump } 0010001$ )

Таблица 3.4 — Ветвь 1: Нет переноса (Продолжение по порядку)

Адрес	A	B	F	1a	1b	АЛБ	Сдв	Сч	ЛУ	i	Адрес перехода	Комментарий
0001110	00	01	00	00	-	111	000	00	000	0	0000000	$P0 := 1$
0001111	100	100	101	01	10	000	000	00	000	0	0000000	$R := \neg F$

Таблица 3.5 — Ветвь 2: Есть перенос (Переход на адрес 17)

Адрес	A	B	F	1a	1b	АЛБ	Сдв	Сч	ЛУ	i	Адрес перехода	Комментарий
0010001	010	011	100	00	00	010	000	00	000	0	0000000	$Q := S - R$ (Обратное выч.)
0010010	00	01	00	00	-	111	000	00	000	0	0000000	$P0 := 1$
0010011	100	100	101	10	11	000	000	00	000	0	0000000	$S := f7 \neg f6 \dots \neg f1$

### 3.2.3 Блок операции Конъюнкции

Расположен по адресу 22 (0010110).

Таблица 3.6 — Микропрограмма операции Конъюнкции

Адрес	A	B	F	1a	1b	АЛБ	Сдв	Сч	ЛУ	i	Адрес перехода	Комментарий
0010110	000	001	010	00	00	100	000	00	000	0	0000000	$Q := R \& S$

### 3.3 Контрольные примеры в модели АЛУ

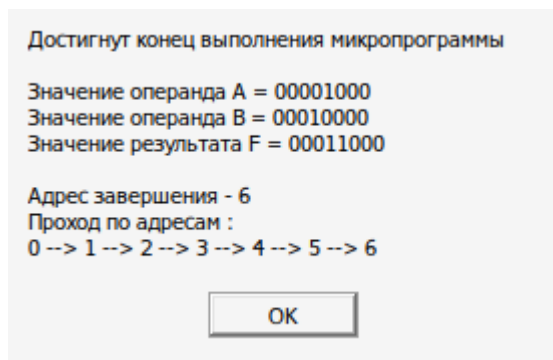


Рисунок 3.1 —  $8+16=24$

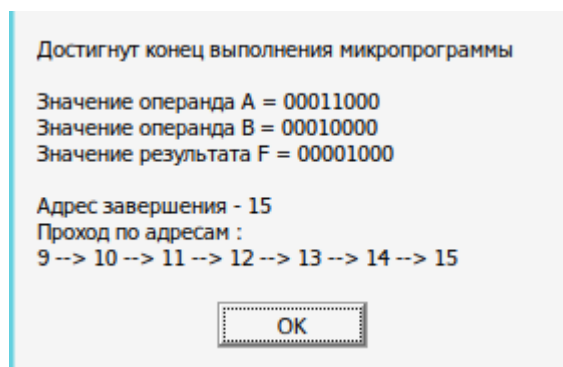


Рисунок 3.2 —  $24-16=8$

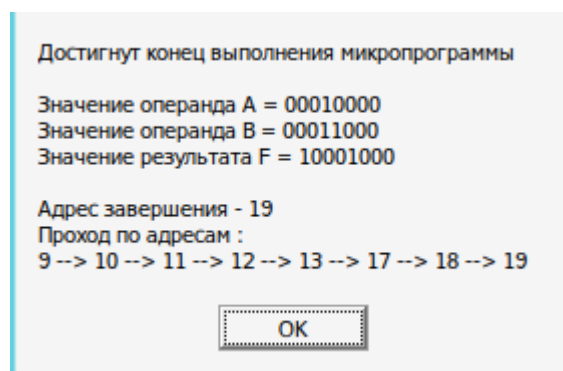


Рисунок 3.3 —  $16-24=-8$

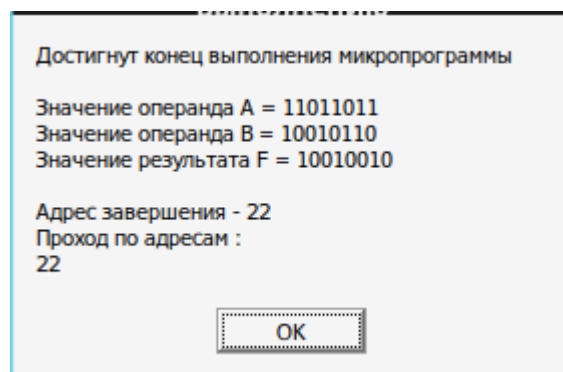


Рисунок 3.4 — Конъюнкция

## ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы было проведено проектирование и программная реализация арифметико-логического устройства на базе учебной модели ALU-R. Основной целью работы являлась разработка микропрограммного управления для выполнения операций сложения, вычитания и конъюнкции над числами в прямом коде.

В процессе исследования теоретических основ и функциональных возможностей модели были решены следующие задачи:

- 1) Анализ алгоритмов обработки данных: Установлено, что для корректного выполнения арифметических операций над знаковыми числами в модели ALU-R необходимо применять алгоритм предварительного преобразования. Входные положительные операнды (прямой код) преобразуются в отрицательные значения в дополнительном коде путем инверсии и добавления единицы ( $P0=1$ ). Результат операции подвергается обратному преобразованию для восстановления прямого кода.
- 2) Синтез управляющего автомата: На этапе проектирования разработан граф переходов автомата Мили, описывающий логику работы устройства. Особое внимание уделено операции вычитания, которая содержит условное ветвление. Реализован механизм анализа флага переноса ( $P8$ ), управляющий переходом к альтернативной ветви алгоритма (обратное вычитание) для корректного вычисления модуля разности. Построены временные диаграммы, демонстрирующие потактовое выполнение операций.
- 3) Программная реализация: Разработанные алгоритмы были транслированы в машинный код модели ALU-R. Карта распределения памяти микрокоманд составлена с учетом топологии графа автомата.

Результаты экспериментального моделирования подтвердили работоспособность спроектированного устройства. АЛУ корректно выполняет арифметические действия с учетом знаков и формирует необходимые признаки результата. Поставленная цель курсовой работы достигнута в полном объеме.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Павловская, Т.А. С/С++. Программирование на языке высокого уровня: Учеб. пособие. – СПб.: Питер, 2007. – 461 с.
- 2) Жмакин, А. П. Архитектура ЭВМ: 2-е изд., перераб. и доп.: учеб. пособие. — СПб.: БХВ-Петербург, 2010. — 352 с.
- 3) Потапов, В.И., Шафеева, О.П., Червенчук, И.В. Основы компьютерной арифметики и логики: Учеб. пособие. – Омск: Изд-во ОмГТУ, 2004. – 172 с
- 4) Потапов, И. В. Элементы прикладной теории цифровых автоматов: учеб. пособие / И. В. Потапов. – Омск: Изд-во ОмГТУ, 2011. – 156 с