

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-информационной систем (РИС)

Кафедра информационных технологий и автоматизированных систем (ИТАС)

УТВЕРЖДАЮ

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

Р.А. Файзрахманов

« \_\_\_\_\_ »

2025 г.

**КУРСОВАЯ РАБОТА**  
по дисциплине  
**«Организация ЭВМ и систем»**  
на тему  
**«Структурно-алгоритмическое проектирование ЭВМ»**

Студент: \_\_\_\_\_ Серебряков Егор Константинович  
(подпись, дата)

Группа: РИС-23-26

Оценка \_\_\_\_\_

Руководитель КР:  
\_\_\_\_\_ к.т.н., доц. каф. ИТАС Погудин А.Л.  
(подпись, дата)

**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-информационной систем (РИС)

Кафедра информационных технологий и автоматизированных систем (ИТАС)

УТВЕРЖДАЮ

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

Р.А. Файзрахманов

«\_\_\_\_\_» 2025 г.

## ЗАДАНИЕ на выполнение курсовой работы

Фамилия, имя, отчество: Серебряков Егор Константинович  
Факультет Электротехнический Группа РИС-23-26

Начало выполнения работы: 16.10.2025

Контрольные сроки просмотра работы: 17.10, 24.11, 1.12

Защита работы: 16.12.2025

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) – Разработка АЛУ, реализующее заданный набор операций с учетом на код выполнения операций и способ построения выполняющего автомата. Тип управляющего автомата – автомат Мура, выполняющий две команды (деление с восстановлением остатка, конъюнкция). Флаги OV, C. Прямой код.

3. Содержание:

3.1 Исследование предметной области

3.2 Основы анализа кода программного языка

3.3 Спецификация устройства на уровне «черного ящика»

3.4 Представление устройства в виде операционной и управляющей частей

3.5 Разработка структуры операционной части 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 учебной модели АЛУ

Руководитель КР:

к.т.н., доц. каф. ИТАС Погудин А.Л.

\_\_\_\_\_  
(подпись, дата)

Задание получил:

Е.К. Серебряков

\_\_\_\_\_  
(подпись, дата)

**КАЛЕНДАРНЫЙ ГРАФИК ВЫПОЛНЕНИЯ  
КУРСОВОЙ РАБОТЫ**

№ пп	Этапы работы	Объём этапа, %	Сроки выполнения	
			Начало	Конец
1.	Исследование предметной области	5	16.10.25	18.10.25
2.	Устройство управления	2	19.10.25	20.10.25
3.	Сложение	2	21.10.25	22.10.25
4.	Вычитание	2	23.10.25	24.10.25
5.	Сравнение	2	25.10.25	26.10.25
6.	Эквивалентность	2	27.10.25	28.10.25
7.	Адресация	2	29.10.25	30.10.25
8.	Разработка устройства	3	31.10.25	02.11.25
9.	Анализ исходных данных задания на курсовую работу	4	03.11.25	05.11.25
10.	Спецификация устройства на уровне «черного ящика»	4	06.11.25	08.11.25
11.	Представление черного ящика в виде операционной и управляющей частей	6	09.11.25	12.11.25
12.	Разработка структуры операционной части устройства	11	13.11.25	19.11.25
13.	Составление схемы алгоритма работы устройства и его микропрограммы	16	20.11.25	28.11.25
14.	Разработка схемы алгоритма работы	5	29.11.25	01.12.25
15.	Составление полной спецификации устройства	3	02.12.25	03.12.25
16.	Разработка фрагмента функциональной схемы вычитателя	5	04.12.25	06.12.25
17.	Контрольный пример	8	07.12.25	09.12.25
18.	Временная диаграмма работы УУ	10	10.12.25	12.12.25
19.	Оформление курсовой работы	8	13.12.25	15.12.25
20.	Защита курсовой работы	0	16.12.25	

Руководитель КР: \_\_\_\_\_ к.т.н., доц. каф. ИТАС Погудин А.Л.  
 (подпись, дата)

Задание получил: \_\_\_\_\_ Серебряков Егор Константинович  
 (подпись, дата)

## РЕФЕРАТ

Отчет 35 с., 16 рис., 2 табл., 4 источн., 1 прил.

# АРИФМЕТИКО-ЛОГИЧЕСКОЕ УСТРОЙСТВО, УПРАВЛЯЮЩЕЕ УСТРОЙСТВО, ДЕЛЕНИЕ С ВОССТАНОВЛЕНИЕМ ОСТАТКА, КОНЬЮНКЦИЯ

Цель работы – разработка арифметико-логического устройства (АЛУ), выполняющего операции деления с восстановлением остатка и конъюнкцию. Операндами являются двоичные числа, представленные в прямом коде.

При разработке АЛУ применялась концепция «черного ящика»: первоначальное определение общих функций устройства и системы входных и выходных сигналов. Дальнейшая работа с «черным ящиком» построена на принципе декомпозиции. При декомпозиции сложные функции разбиваются на более простые подфункции до получения описания функции на элементарном уровне.

В ходе выполнения курсовой работы достигнуты следующие результаты: разработано арифметико-логическое устройство на основе учебной программной модели АЛУ-R, приведен контрольный пример в числовой форме

## СОДЕРЖАНИЕ

СОДЕРЖАНИЕ .....	5
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ .....	6
1 Исследование предметной области курсовой работы .....	8
1.1 Прямой код .....	8
1.2 Деление двоичных чисел с восстановлением остатка в прямом коде .....	8
1.3 Логическая конъюнкция .....	9
1.4 Флаги OV и C .....	10
1.5 Проектирование управляющего автомата Мили на основе графических схем .....	11
2 Разработка устройства .....	13
2.1 Анализ исходных данных задания на курсовую работу .....	13
2.2 Спецификация устройства на уровне «черного ящика» .....	13
2.3 Представление устройства в виде операционной и управляющей частей .....	15
2.4 Разработка структуры операционной части устройства .....	16
2.5 Разработка схемы алгоритма работы .....	17
2.6 Составление полной спецификации устройства .....	21
2.7 Разработка временной диаграммы устройства .....	24
2.8 Контрольный пример .....	26
ЗАКЛЮЧЕНИЕ .....	30
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ .....	31
ПРИЛОЖЕНИЕ А .....	32
ПРИЛОЖЕНИЕ Б .....	34

## **ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

АЛУ – Арифметико-логическое устройство

МО – Микрооперация

МПР – Микропрограмма

ЧЯ – Черный ящик

ОЧ – Операционная часть устройства

УЧ – Управляющая часть устройства

ГТИ – Генератор тактовых импульсов

ГСА – Граф-схема алгоритма

## ВВЕДЕНИЕ

Арифметико-логическое устройство (АЛУ) является центральным узлом процессора, обеспечивающим реализацию вычислительных процедур и логических преобразований данных.

Целью данной работы выступает проектирование АЛУ, ориентированного на выполнение арифметической операции деления по методу с восстановлением остатка и логической операции конъюнкцией. Обработке подлежат 8-разрядные двоичные числа, представленные в формате прямого кода. Синтез управляющего устройства базируется на модели конечного автомата Мили. В функционал проектируемого модуля заложена генерация сигналов состояния: признака переполнения разрядной сетки (OV) и признака переноса из старшего разряда (C).

Объектом исследования является структурная организация арифметико-логических устройств, предметом — алгоритмы функционирования и аппаратная реализация управляющего автомата.

В ходе исследования проведет анализ исходных требований, formalизована спецификация устройства на уровне модели «черного ящика», а также спроектированы структурная схема и граф-схема алгоритма управления. Корректность проектных решений верифицирована посредством построения временных диаграмм и расчетного моделирования на контрольном числовом примере.

# **1 Исследование предметной области курсовой работы**

## **1.1 Прямой код**

Прямой код представляет собой один из основных форматов представления целых чисел в двоичной арифметике. В этом формате N-разрядное двоичное число разделяется на две логические части. Старший разряд выделяется под бит знака, где логический нуль (0) интерпретируется как положительный знак, а логическая единица (1) — как отрицательный. Оставшиеся N-1 младших разрядов формируют информационную часть, содержащую численное значение (модуль) операнда в стандартной двоичной форме.

## **1.2 Деление двоичных чисел с восстановлением остатка в прямом коде**

Алгебраическое деление двоичных чисел в прямом коде с восстановлением остатка предполагает раздельное выполнение операций над знаком и модулем операндов. Значение итогового частного определяется согласно математическому правилу знаков: результат положителен, если знаки делимого и делителя совпадают, и отрицателен, если они различны. После определения знака выполняется итеративное деление абсолютных величин. Знак частного вносится в старший разряд финального результата.

Алгоритм включает следующие ключевые этапы:

- 1) Определение знака частного: Знак частного вычисляется как результат операции исключающего ИЛИ (XOR) между знаковыми битами делимого и делителя;
- 2) Определение знака остатка: Знак остатка всегда совпадает со знаком исходного делимого, что соответствует стандартному определению операции деления;
- 3) Выполнение деления модулей: Основной вычислительный процесс базируется на алгоритме деления с восстановлением остатка, который

оперирует исключительно положительными модулями чисел. Суть алгоритма — N разрядных итераций (где N — разрядность операндов), каждая из которых включает: общий сдвиг составного регистра (остаток и частное) влево, пробное вычитание делителя из остатка, Анализ остатка: Если остаток остался положительным, в младший разряд частного записывается 1. Если остаток стал отрицательным, он восстанавливается (возвращается к значению до вычитания), и в частное записывается 0;

4) Формирование результата: Финальный модуль частного (полученный в ходе итераций) дополняется вычисленным знаковым битом.

При работе в прямом коде возможны четыре комбинации знаков операндов, которые влияют только на финальную коррекцию знака:

1) Случай  $A/B > 0$  ( $A>0, B>0$  или  $A<0, B<0$ ) не требуют коррекции знака частного;

2) Случай  $A/B < 0$  ( $A>0, B<0$  или  $A<0, B>0$ ) требуют присвоения отрицательного знака модулю частного, полученному в результате итераций.

### **1.3 Логическая конъюнкция**

Конъюнкция (логическое И, операция AND) является бинарной логической операцией, которая возвращает 1, если оба операнда имеют значение 1. Результат равен 0 в том случае, если один из operandов равен 0. При работе с многоразрядными двоичными числами эта операция, как и большинство логических операций в вычислительной технике, выполняется поразрядно.

Рассмотрим операцию конъюнкции на примере двух двоичных чисел с разрядностью равной 4:

$$A = 1101_2 \quad B = 1010_2$$

Выполним операцию конъюнкции для каждого разряда чисел:

$$1 \ \& \ 1 = 1$$

$1 \& 0 = 0$

$0 \& 1 = 0$

$1 \& 0 = 0$

Результатом операции конъюнкции над числами А и В будет значение равное  $1000_2$ .

#### **1.4 Флаги OV и C**

Флаги OV (арифметическое переполнение) и C (перенос/заем) являются ключевыми диагностическими признаками, которые используются арифметико-логическим устройством для контроля хода и результата операций.

Флаг OV служит прямым индикатором критической ошибки, сигнализируя о том, что результат арифметической операции вышел за пределы разрядной сетки, предусмотренной архитектурой.

При выполнении операции деления флаг OV устанавливается исключительно в случае попытки деления на ноль ( $B=0$ ). В контексте алгебраического деления ненулевых операндов, арифметическое переполнение частного невозможно, поскольку его абсолютная величина всегда ограничена.

Для чисто логических операций, таких как конъюнкция, флаг OV всегда сбрасывается в 0. Это обусловлено тем, что логические преобразования поразрядны, не зависят от знака и не могут физически генерировать ошибку переполнения.

Флаг C является вспомогательным признаком, который отражает состояние переноса или заема из самого старшего разряда при выполнении арифметических операций.

В алгоритме деления с восстановлением остатка флаг C критически важен на каждом промежуточном шаге. Он используется для анализа знака

остатка после выполнения пробного вычитания делителя. Наличие переноса из старшего разряда (или отсутствие заема, если используется дополнительный код) указывает на положительный промежуточный остаток. Эта информация необходима управляющему автомatu для принятия решения о записи единицы в частное. После завершения всего цикла деления значение С, как правило, не сохраняется.

При выполнении логических операций (например, конъюнкции), флаг С сбрасывается в 0. Это происходит потому, что логические операции не имеют отношения к арифметике и, следовательно, не используют и не генерируют сигналы переноса или заема.

Таким образом, флаг OV информирует о невозможности получения результата (ошибка), тогда как флаг С является ключевым элементом, который управляет пошаговой последовательностью сложных арифметических вычислений.

## **1.5 Проектирование управляющего автомата Мили на основе графических схем**

Переход от алгоритмического описания, представленного Графической Схемой Алгоритма (ГСА), к управляющему автомatu типа Мили является необходимым этапом формализации, обеспечивающим синтез логической структуры управляющего устройства. ГСА служит исходной моделью, а конечный автомат Мили — её аппаратной реализацией.

Процесс преобразования ГСА в модель конечного автомата включает следующие ключевые этапы:

- 1) Каждому операторному блоку ГСА ставится в соответствие уникальное внутреннее состояние  $Y_i$  автомата. Эти состояния формируют множество возможных позиций управляющего устройства в ходе выполнения алгоритма;

2) Логические условия (ветви алгоритма), представленные в ГСА, преобразуются в осведомительные входные сигналы  $X_j$ , которые поступают на управляющий автомат от операционного блока;

3) Определяется логика смены состояний. Для каждой комбинации (текущее состояние  $Y_i$ , входной сигнал  $X_j$ ) однозначно определяется следующее состояние  $Y_k$ ;

4) В соответствии с принципами автомата Мили, выходные управляющие сигналы (микрооперации)  $Z_k$  формируются как функция текущего состояния  $Y_i$  и текущего входного сигнала  $X_j$ .

Математическая модель, описывающая функционирование конечного автомата, представляется системой двух функций:

$$Y(t+1) = \delta(Y(t), X(t)) \text{ — функция переходов;}$$

$$Z(t) = \lambda(Y(t), X(t)) \text{ — функция выходов;}$$

где  $Y$  — состояние автомата,  $X$  — входные сигналы, а  $Z$  — выходные управляющие сигналы.

Результатом данного этапа проектирования является таблица переходов, которая содержит полную формализованную информацию о реакциях управляющего устройства на все возможные комбинации входных воздействий в каждом состоянии системы, являясь непосредственной основой для последующего синтеза логических схем.

## **2 Разработка устройства**

### **2.1 Анализ исходных данных задания на курсовую работу**

В соответствии с техническим заданием, разрабатываемое арифметико-логическое устройство (АЛУ) должно обеспечивать выполнение двух основных операций: конъюнкции (логическое И) и деления с восстановлением остатка. Обработке подлежат 8-разрядные двоичные числа, представленные в прямом коде.

Устройство функционирует под управлением автомата типа Мили с жёсткой логикой, который реализует заданный алгоритм выполнения обеих операций.

Входными данными для АЛУ являются два 8-разрядных операнда (делимое A и делитель B для арифметики; операнды A и B для логики).

Выходными данными являются 8-разрядный результат операции (частное или логический результат) и значения флагов состояния.

Особенностью проектирования является необходимость обеспечения корректной обработки знаковых чисел в прямом коде, что требует раздельного анализа знаков перед началом деления и последующего присвоения знака результату.

Также критически важным является формирование соответствующих признаков результата для обеих операций, в частности флагов OV (переполнение) и флага C (перенос/заем) для управления итерационным процессом деления.

### **2.2 Спецификация устройства на уровне «чёрного ящика»**

На уровне «чёрного ящика» проектируемое арифметико-логическое устройство (АЛУ) рассматривается как функциональный блок, определяемый исключительно своими внешними интерфейсами: входными operandами, управляющими сигналами и выходными результатами со статусами.

АЛУ принимает два 8-разрядных операнда, обозначенных A и B, которые служат исходными данными для выполнения как арифметических (деление), так и логических (конъюнкция) операций. Кроме того, имеется управляющий вход U, который задает код выполняемой операции.

Устройство генерирует 8-разрядный выходной результат C, несущий значение частного (для деления) или результат поразрядной конъюнкции. Помимо основного результата, формируются одноразрядные флаги состояния: OV (Overflow — Переполнение), сигнализирующий о критических ошибках, таких как деление на ноль и C (Carry/Borrow — Перенос/Заем), который содержит признак переноса или заема из старшего разряда, важный для контроля итерационного алгоритма деления.

На рисунке 1 изображена спецификация устройства на уровне «черного ящика».

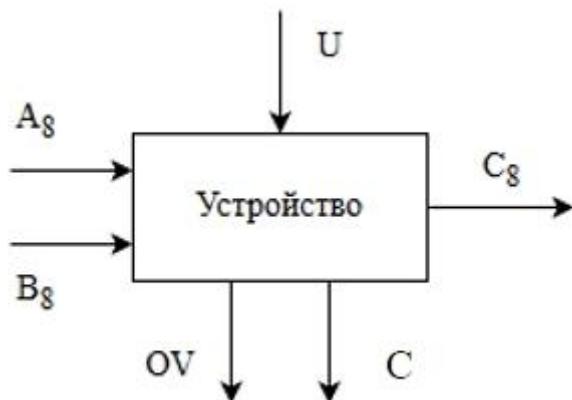


Рисунок 1 – спецификация устройства на уровне «черного ящика».

Таким образом, АЛУ представляет собой блок, который преобразует 8-разрядные входные данные в результат, подчиняясь команде, заданной входом U, и предоставляя диагностическую информацию о статусе операции через выходы OV и C.

## 2.3 Представление устройства в виде операционной и управляющей частей

Архитектура АЛУ декомпозируется на две взаимодействующие части для чёткого распределения функций: Операционную Часть (ОЧ) и Управляющую Часть (УЧ).

Операционная Часть (ОЧ) выполняет все арифметико-логические преобразования (деление с восстановлением остатка и конъюнкция). В процессе работы ОЧ генерирует осведомительные флаги состояния ( $OV$  — переполнение и  $C$  — перенос/заем), отражающие результат последней микрооперации.

Управляющая Часть (УЧ), реализованная как автомат Мили, принимает флаги  $OV$  и  $C$  (обратная связь) и код команды  $U$ . Анализируя эту информацию, УЧ генерирует сигналы управления, которые определяют последовательность микроопераций и синхронизацию работы ОЧ. Такое разделение является основополагающим принципом построения управляемых вычислительных устройств.

На рисунке 2 изображена декомпозиция «чёрного ящика» в ОЧ и УЧ.

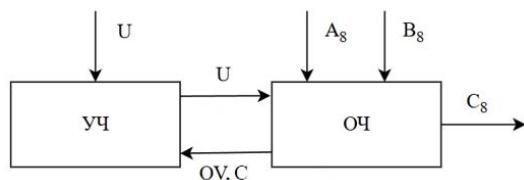


Рисунок 2 – декомпозиция «чёрного ящика» в ОЧ и УЧ

Таким образом, УЧ функционирует как логический контроллер, определяя следующий шаг вычислительного процесса, тогда как ОЧ выступает в роли исполнительного модуля, реализующего эти шаги на уровне данных.

## 2.4 Разработка структуры операционной части устройства

Структура операционной части (ОЧ) спроектирована для обеспечения хранения данных и выполнения всех необходимых вычислительных и логических преобразований. Она включает в себя два 8-разрядных регистра для приема операндов А и В, а также 8-разрядный регистр вывода С для сохранения результата выполненной операции.

Функциональность ОЧ обеспечивается специализированными вычислительными модулями. В соответствии с заданным функционалом, предусмотрены два основных блока: блок конъюнкции, выполняющий поразрядную операцию AND над соответствующими битами операндов, и блок деления с восстановлением остатка, который реализует итерационный алгоритм последовательного вычитания и сдвига.

На рисунке 3 изображена структура операционной части устройства.

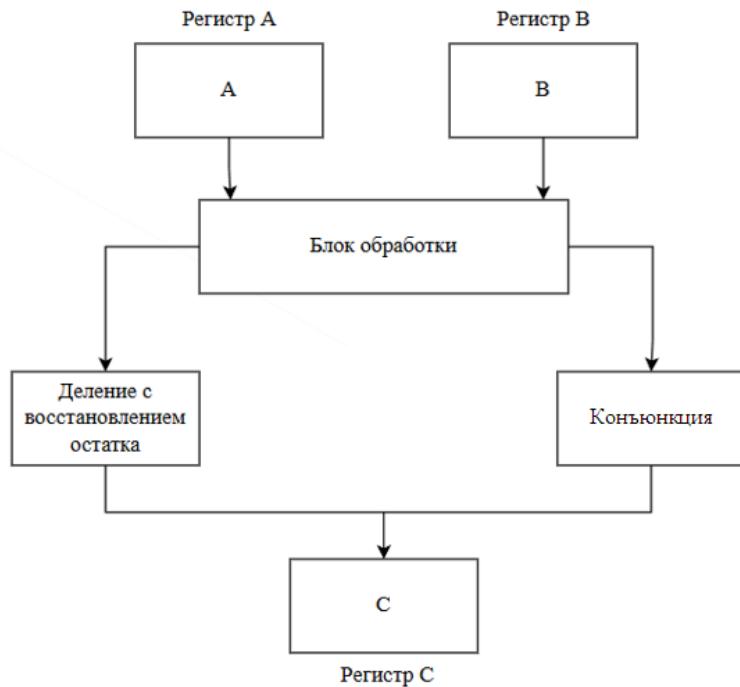


Рисунок 3 – структура операционной части устройства

Последовательность микроопераций в операционной части задается управляющей частью, которая генерирует соответствующие сигналы управления на основе кода выполняемой команды.

## 2.5 Разработка схемы алгоритма работы

Алгоритм работы представлен последовательностью микроопераций, выполняющих операции деления с восстановлением остатка и конъюнкции.

Конъюнкция представлена итерационным циклом, проходящим по каждым битам двух соответствующих разрядов operandов и выполняющим конъюнкцию между соответствующими разрядами соответствующих operandов.

Деление с восстановлением остатка представлено двумя итерационными циклами, производящими требуемые действия для сдвига operandов, вычисления разности между ними и нахождения значений для младших разрядов сдвигаемых значений.

На рисунках 4-7 представлена разработанная блок-схема алгоритма.

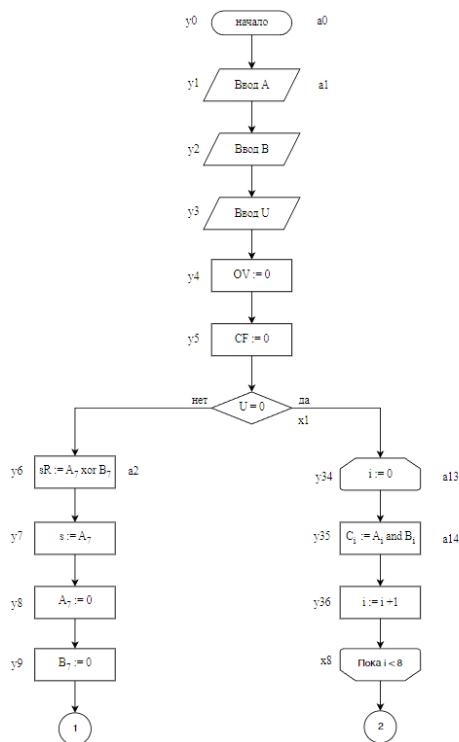


Рисунок 4 – блок-схема алгоритма (начало)

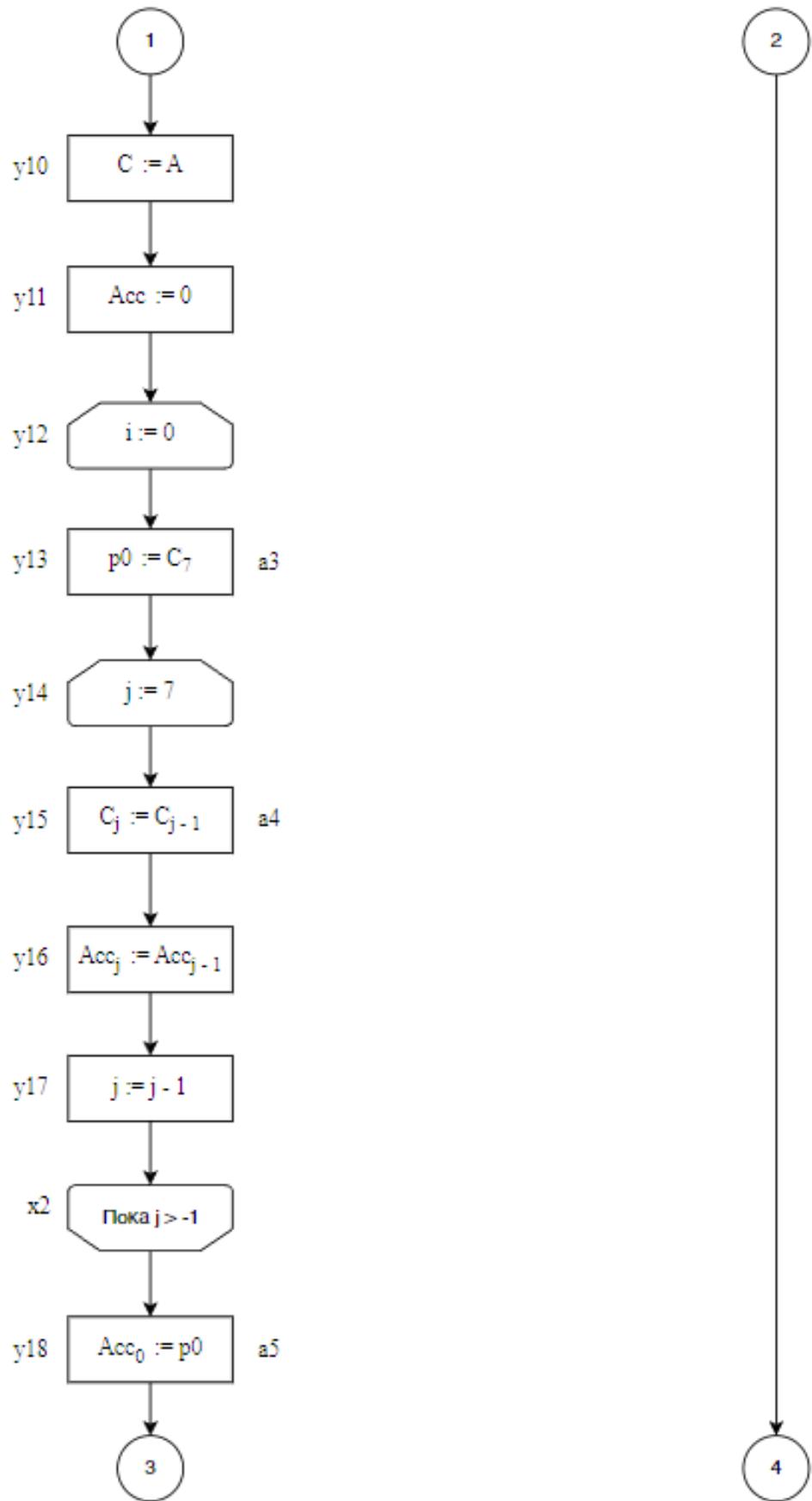


Рисунок 5 – блок-схема алгоритма (продолжение)

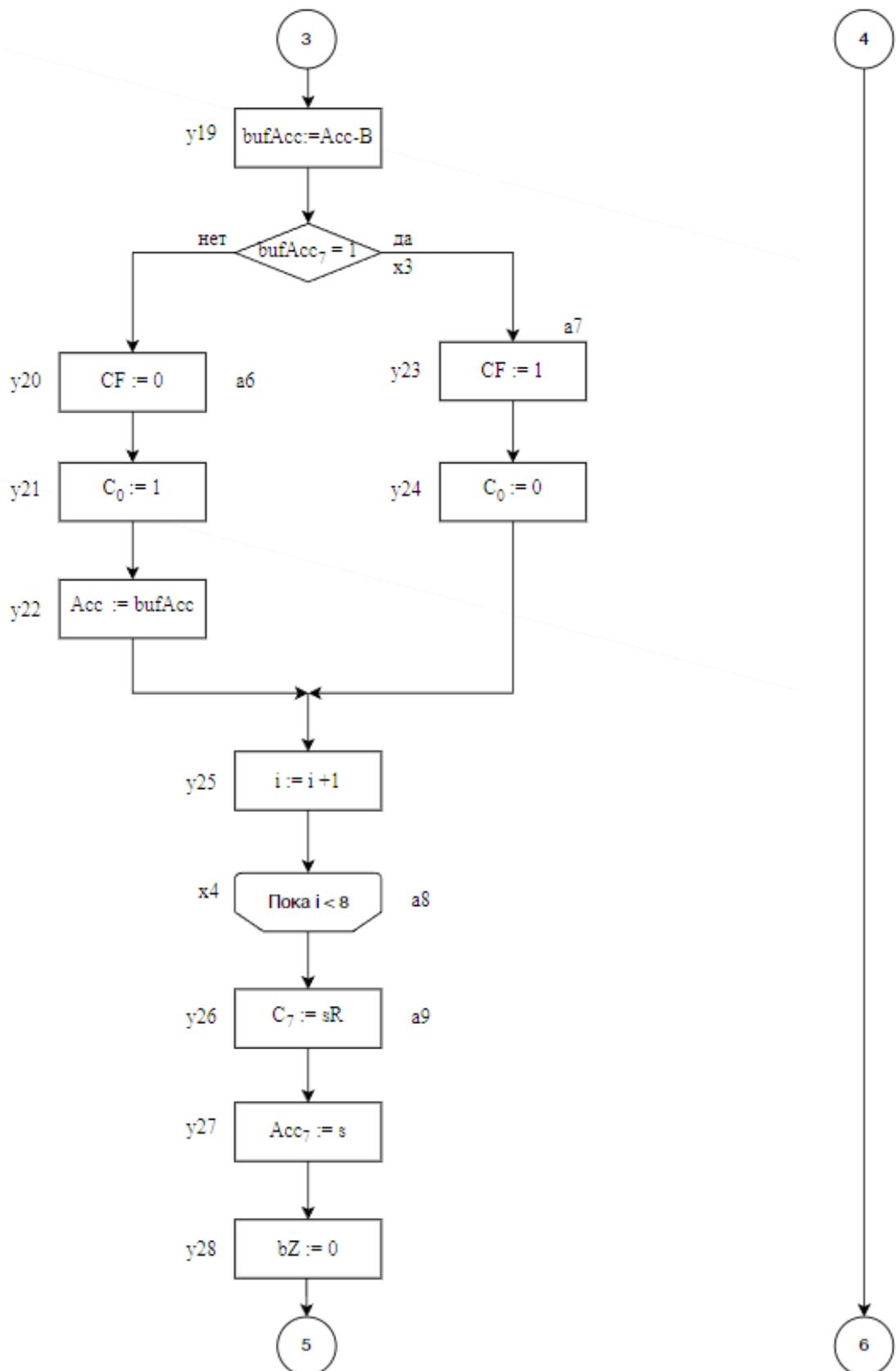


Рисунок 6 – блок-схема алгоритма (продолжение)

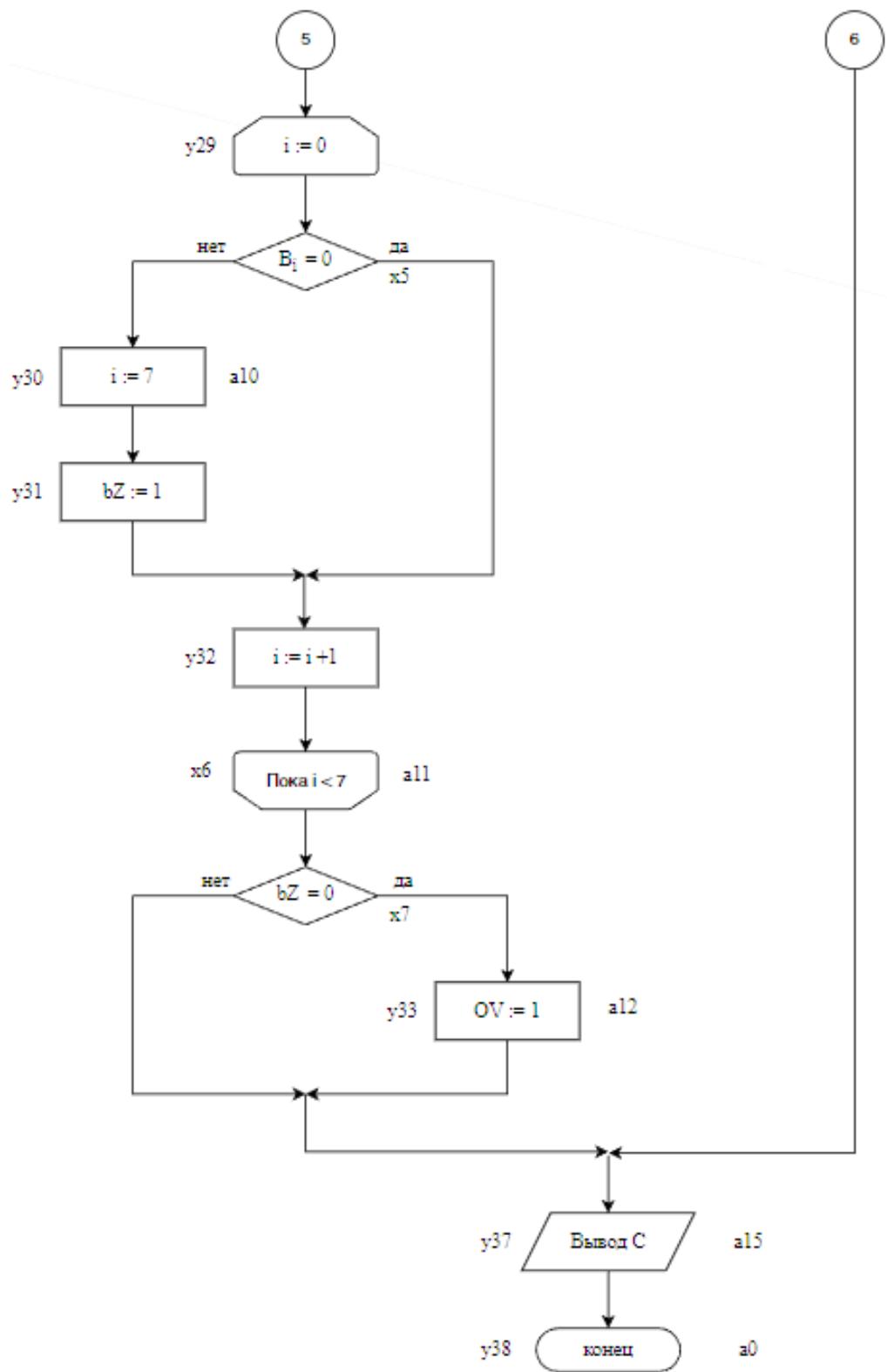


Рисунок 7 – блок-схема алгоритма (конец)

По составленной блок-схеме алгоритма построен граф переходов Мили.

На рисунке 8 изображен автомат Мили, составленный на основе блок-схемы алгоритма.

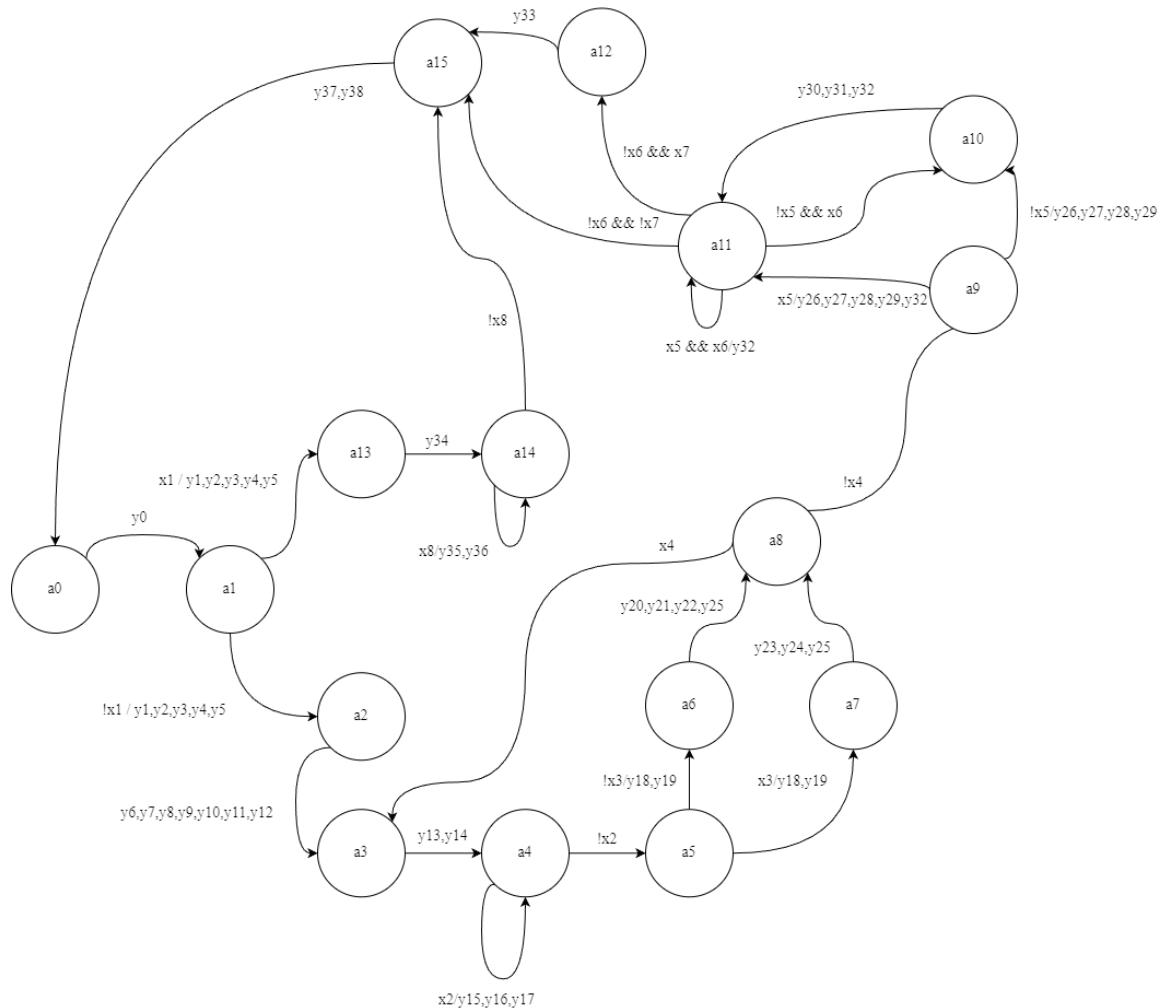


Рисунок 8 – автомат Мили

Вершины автомата Мили соответствуют состояниям автомата, дуги – отражают переходы между состояниями на основании логических условий.

## 2.6 Составление полной спецификации устройства

Анализ блок-схемы алгоритма и графа переходов автомата Мили позволяет составить полную спецификацию проектируемого устройства.

В таблице 1 приведено описание всех линий и сигналов, полученных в процессе разработки алгоритма работы АЛУ.

Таблица 1 – сигналы устройства

Имя сигнала/шины, размерность	Тип (In/Out)	Назначение сигнала
y0	In для ОЧ	Сигнал начала программы
y1	In для ОЧ	Сигнал записи числа А в регистр
y2	In для ОЧ	Сигнал записи числа В в регистр
y3	In для ОЧ	Сигнал получения управляющего сигнала U
y4	In для ОЧ	Сигнал установки флага OV = 0
y5	In для ОЧ	Сигнал установки флага CF (флаг C) = 0
y6	In для ОЧ	Сигнал вычисления знака результата и запись знака в переменную sR
y7	In для ОЧ	Сигнал сохранения знака остатка в переменную s
y8	In для ОЧ	Сигнал записи в старший разряд числа А нуля
y9	In для ОЧ	Сигнал записи в старший разряд числа В нуля
y10	In для ОЧ	Сигнал записи в переменную С (частное) числа А
y11	In для ОЧ	Сигнал записи во все разряды Acc (остаток) нулей
y12	In для ОЧ	Сигнал записи в счетчик i числа 0
y13	In для ОЧ	Сигнал записи в p0 старшего бита числа С
y14	In для ОЧ	Сигнал записи в счетчик j числа 7
y15	In для ОЧ	Сигнал записи в разряд j числа С числа в разряде j-1 числа С
y16	In для ОЧ	Сигнал записи в разряд j числа Acc числа в разряде j-1 числа Acc
y17	In для ОЧ	Сигнал записи в счетчик j значения j-1
y18	In для ОЧ	Сигнал записи в младший бит остатка (Acc) значения старшего бита числа С, сохраненного в p0
y19	In для ОЧ	Сигнал вычисления разницы между остатком (Acc) и делителем (B) с сохранением в buffAcc
y20	In для ОЧ	Сигнал записи во флаг С (CF – заем) нуля

Продолжение таблицы 1

y21	In для ОЧ	Сигнал записи в младший разряд С (частного) числа 1
y22	In для ОЧ	Сигнал записи в остаток (Acc) значения, хранимого в bufAcc
y23	In для ОЧ	Сигнал записи во флаг С (CF – заем) единицы
y24	In для ОЧ	Сигнал записи в младший разряд С (частного) числа 0
y25	In для ОЧ	Сигнал увеличения счетчика i на +1
y26	In для ОЧ	Сигнал восстановления знака частного (C) с помощью присваивания sR
y27	In для ОЧ	Сигнал восстановления знака остатка (Acc) с помощью присваивания s
y28	In для ОЧ	Сигнал записи 0 во флаг bZ являющийся флагом равенства делителя нулю
y29	In для ОЧ	Сигнал записи в счетчик i числа 0
y30	In для ОЧ	Сигнал записи в счетчик i числа 7
y31	In для ОЧ	Сигнал записи 1 во флаг bZ
y32	In для ОЧ	Сигнал увеличения счетчика i на +1
y33	In для ОЧ	Сигнал записи в флаг OV числа 1
y34	In для ОЧ	Сигнал записи в счетчик i числа 0
y35	In для ОЧ	Сигнал вычисления конъюнкции между i-разрядами числа А и В и присваивание результата в i-разряд числа С
y36	In для ОЧ	Сигнал увеличения счетчика i на +1
y37	In для ОЧ	Сигнал вывода числа С в регистр
y38	In для ОЧ	Сигнал конца программы
x1	Out для ОЧ	Сигнал проверки числа U на равенство нулю
x2	Out для ОЧ	Сигнал проверки счетчика j > -1
x3	Out для ОЧ	Сигнал проверки старшего разряда bufAcc на равенство 1
x4	Out для ОЧ	Сигнал проверки счетчика i < 8
x5	Out для ОЧ	Сигнал проверки i-разряда числа В на равенство 0
x6	Out для ОЧ	Сигнал проверки счетчика i < 7
x7	Out для ОЧ	Сигнал проверки флага Bz на равенство нулю
x8	Out для ОЧ	Сигнал проверки счетчика i < 8

На рисунке 9 приведена структура устройства на микрокомандном уровне управления.



Рисунок 9 – структура устройства на микрокомандном уровне управления

## 2.7 Разработка временной диаграммы устройства

Временные диаграммы служат инструментом проектирования цифровых устройств. Они обеспечивают визуализацию временных соотношений между всеми сигналами в системе. На этих диаграммах отображается, как изменяются логические уровни управляющих и информационных сигналов на протяжении выполнения вычислительного алгоритма.

Для сложных операций, которые требуют нескольких тактов, временные диаграммы строятся для ключевых фаз алгоритма. Это позволяет наглядно показать, как взаимодействуют управляющая и операционная части устройства в наиболее характерных режимах работы.

На рисунке 10 представлена временная диаграмма работы автомата при одной итерации контрольного примера конъюнкции.

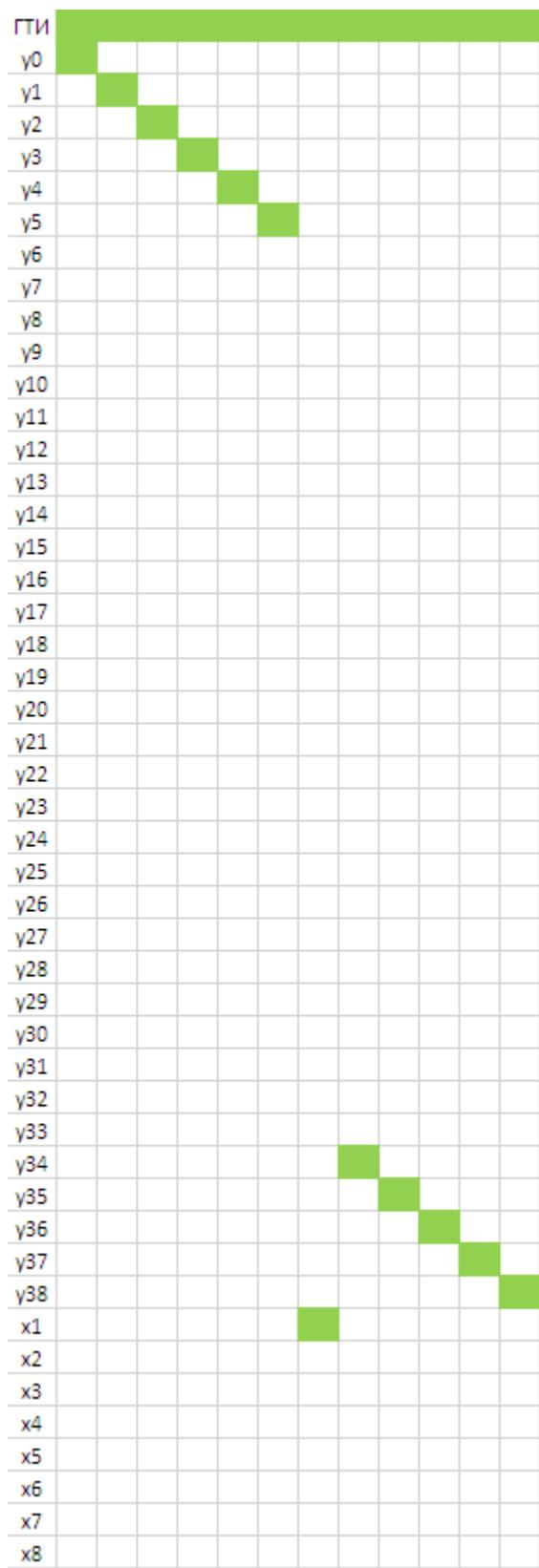


Рисунок 10 – времененная диаграмма итерации конъюнкции

На рисунке 11 представлена временная диаграмма одной итерации при выполнении деления с восстановлением остатка.

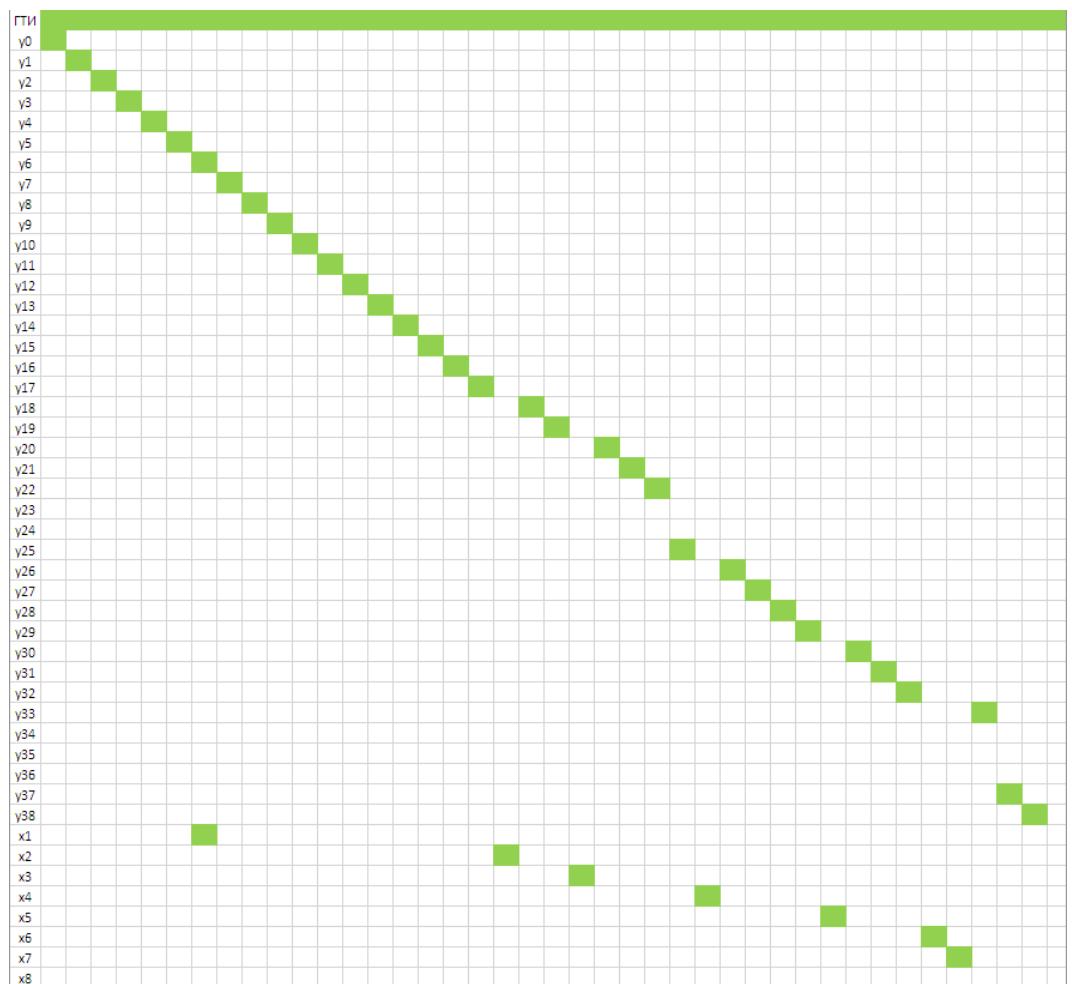


Рисунок 11 – временная диаграмма итерации деления с восстановлением остатка

## 2.8 Контрольный пример

Контрольные примеры выполняются в учебной программе АЛУ-Р.

На рисунках 12-13 представлены контрольные примеры поразрядной конъюнкции.

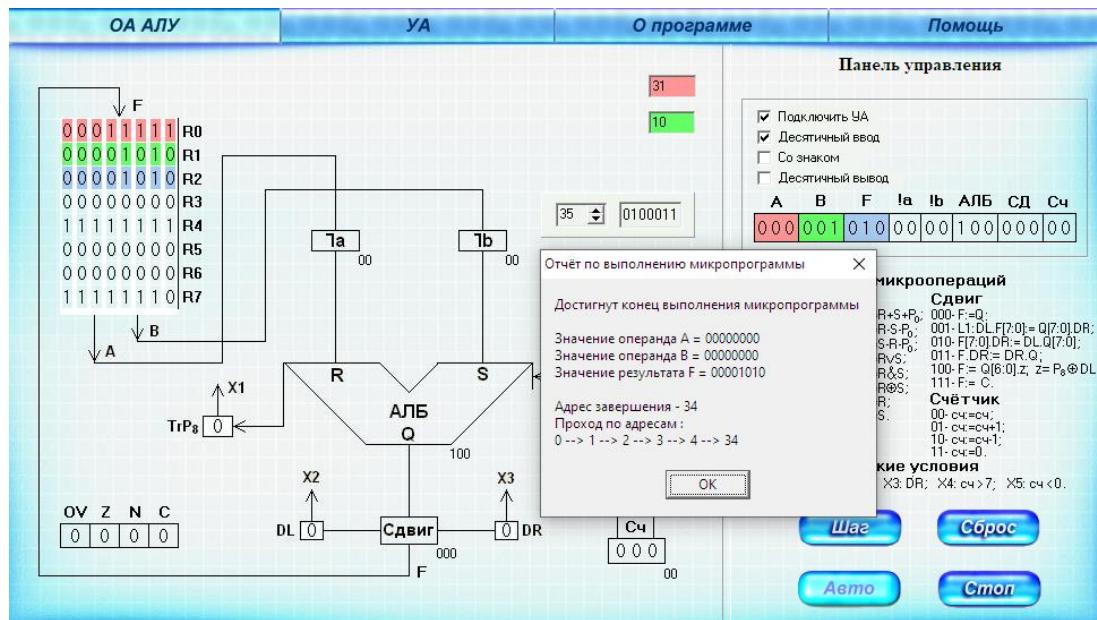


Рисунок 12 – первый контрольный пример поразрядной конъюнкции

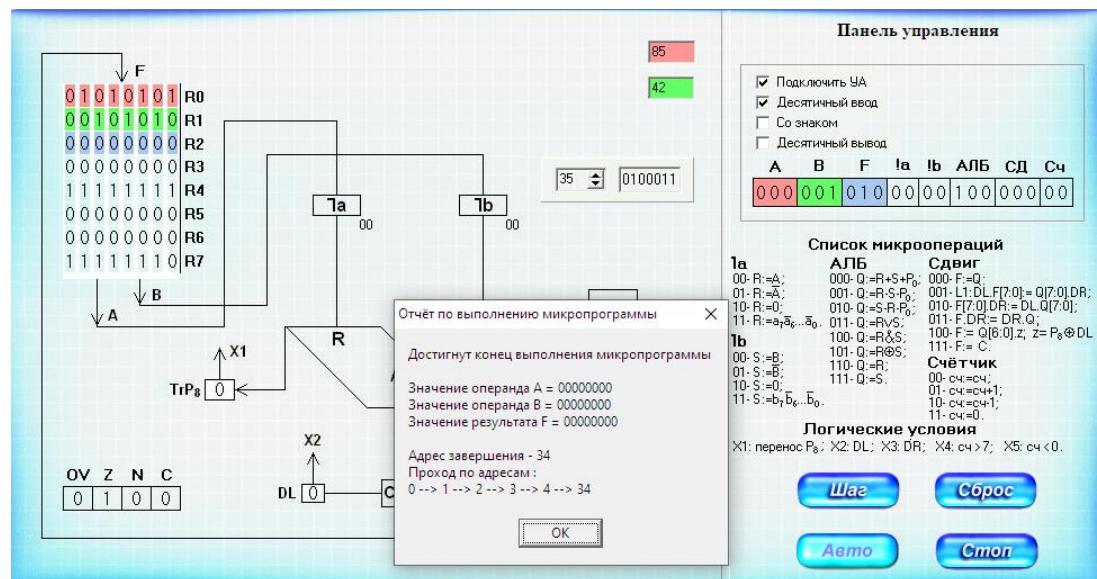


Рисунок 13 – второй контрольный пример поразрядной конъюнкции

На рисунках 14-16 представлены контрольные примеры деления с восстановлением остатка.

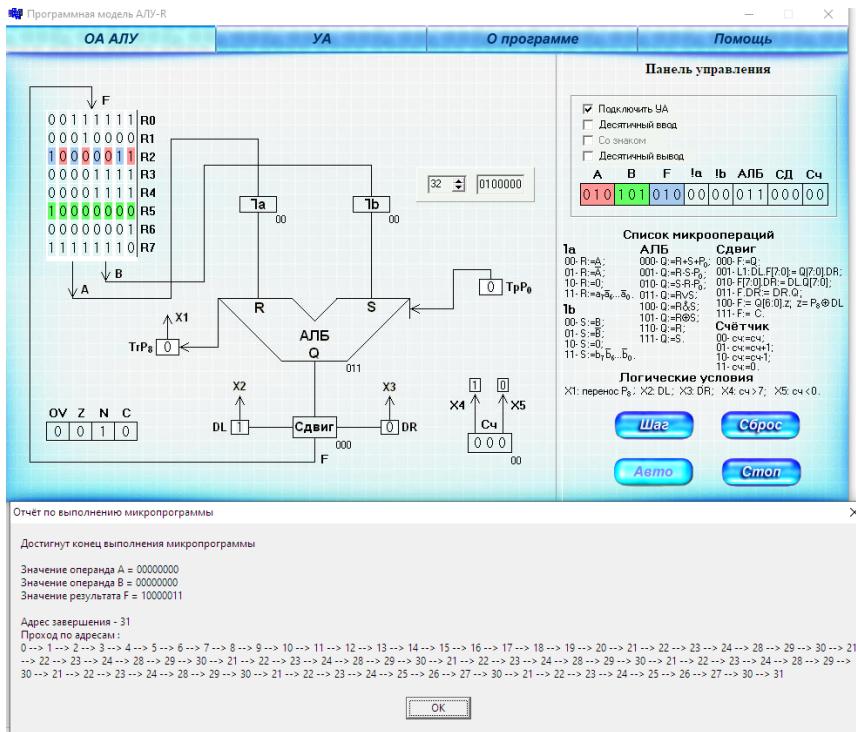


Рисунок 14 – первый контрольный пример деления с восстановлением остатка

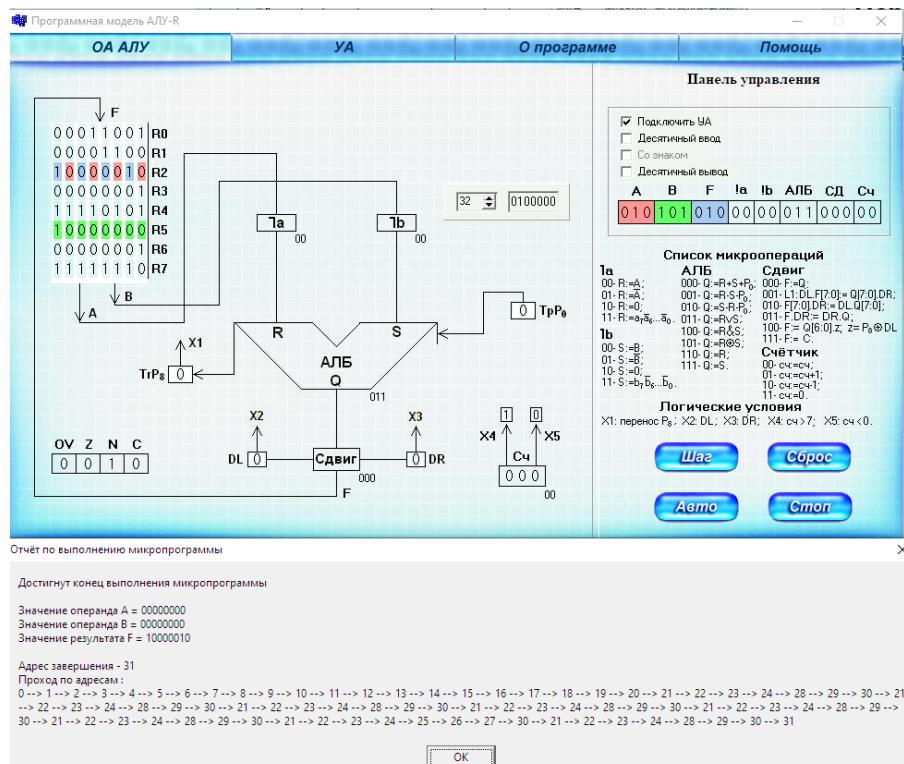


Рисунок 15 – второй контрольный пример деления с восстановлением остатка

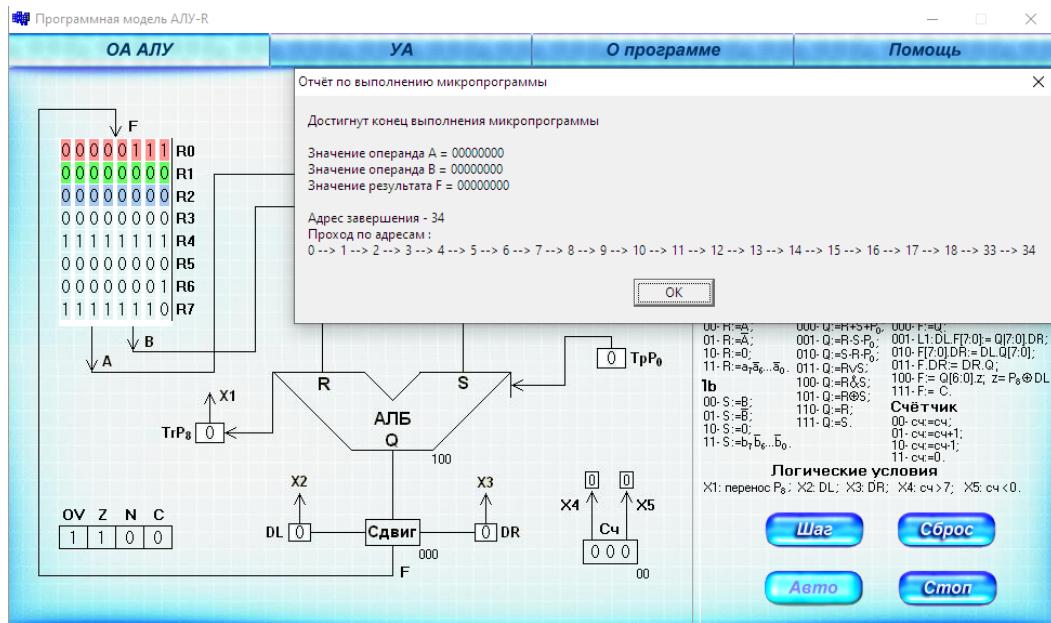


Рисунок 16 – третий контрольный пример деления с восстановлением остатка

В таблице 2 приведены назначения регистров, используемых в АЛУ-Р.

Таблица 2 – Назначение регистров АЛУ-Р

Адрес	Описание
R0	Операнд A
R1	Операнд B
R2	Регистр С - результат выполненной операции
R3	Регистр D – остаток деления
R4	Регистр U – выбор операции (0 – конъюнкция, 1 – деление с восстановлением остатка). Далее используется в качестве буфера
R5	Регистр Sr – знак результата
R6	Регистр маски для установки в младший разряд числа 1
R7	Регистр маски для установки в младший разряд числа 0

## ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы было спроектировано и реализовано арифметико-логическое устройство (АЛУ). Данное 8-разрядное устройство выполняет операции деления с восстановлением остатка и поразрядной конъюнкции над числами в прямом коде, функционируя под управлением автомата Мили с жёсткой логикой.

В процессе проектирования был пройден полный цикл разработки: от формализации спецификаций на уровне «чёрного ящика» и создания структурной схемы до синтеза алгоритмов функционирования и построения временных диаграмм. Детальная проработка взаимодействия операционного и управляющего узлов обеспечила корректность системы.

Тестирование на контрольных примерах подтвердило правильность функционирования АЛУ и верифицировало точность формирования флагов состояния (OV и C). Выполненная работа позволила систематизировать знания о принципах микропрограммного управления и методах логического проектирования цифровых вычислительных устройств.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Жмакин, А. П. Архитектура ЭВМ: 2-е изд., перераб. и доп.: учеб. пособие. — СПб.: БХВ-Петербург, 2010. — 352 с.
2. Баранов С. И. Синтез микропрограммных автоматов. — Л.: Энергия, 1974. — 216 с.
3. Ожиганов, А. А. Теория автоматов : Учебное пособие / А. А. Ожиганов. — Санкт-Петербург : Университет ИТМО, 2013. — 86 с. — EDN ZUZSBZ.
4. ГОСТ 7.32-2017 «Отчет о научно-исследовательской работе. Структура и правила оформления»

## ПРИЛОЖЕНИЕ А

### Листинг программы в бинарном виде

111 111 111 10 10 110 000 00 000 0 0000000

111 111 111 01 10 110 000 00 000 0 0000000

111 111 111 00 00 110 100 00 000 0 0000000

100 111 100 00 01 001 000 00 000 0 0000000

00 00 00 00 - - 111 00 001 0 0100010

101 101 101 10 10 110 000 00 000 0 0000000

101 101 110 01 01 110 000 00 000 0 0000000

110 110 110 00 00 110 011 00 000 0 0000000

101 101 101 00 00 110 011 00 000 0 0000000

101 101 100 00 00 110 000 00 000 0 0000000

111 111 110 01 01 110 000 00 000 0 0000000

000 001 010 10 10 000 000 00 000 0 0000000

000 101 101 00 00 100 000 00 000 0 0000000

000 101 000 00 00 001 000 00 000 0 0000000

001 100 100 00 00 100 000 00 000 0 0000000

001 100 001 00 00 001 000 00 000 0 0000000

100 101 101 00 00 101 000 00 000 0 0000000

110 001 100 00 00 010 000 00 000 0 0000000

00 00 00 00 - - 111 00 001 0 0100001

000 000 000 00 00 110 000 00 000 0 0000000

000 000 010 00 00 110 000 00 000 0 0000000

010 000 010 00 00 110 001 00 000 0 0000000

011 000 011 00 00 110 100 00 000 0 0000000

011 001 100 00 00 001 000 00 000 0 0000000

00 00 00 00 - - 111 00 001 0 0011100

100 000 011 00 00 110 000 00 000 0 0000000

00 00 01 00 - - 111 00 000 0 0000000

010 110 010 00 00 011 000 01 111 0 0011110

00 00 10 00 - - 111 00 000 0 0000000

010 111 010 00 00 100 000 01 000 0 0000000

00 00 00 00 - - 111 00 100 1 0010101

010 101 010 00 00 011 000 00 000 0 0000000

01 00 00 00 - - 111 00 000 0 0000000

000 001 010 00 00 100 000 00 000 0 0000000

## ПРИЛОЖЕНИЕ Б

### Листинг программы в виде мнемоник

R7 R7 0 0 R Q Ch 0 0 0

R7 R7 !A 0 R Q Ch 0 0 0

R7 R7 A B R RD Ch 0 0 0

R7 R4 A !B SUBA Q Ch 0 0 0

P8 DL DR - - C Ch X1 0 34

R5 R5 0 0 R Q Ch 0 0 0

R5 R6 !A !B R Q Ch 0 0 0

R6 R6 A B R LM Ch 0 0 0

R5 R5 A B R LM Ch 0 0 0

R5 R4 A B R Q Ch 0 0 0

R7 R6 !A !B R Q Ch 0 0 0

R1 R2 0 0 ADD Q Ch 0 0 0

R5 R5 A B AND Q Ch 0 0 0

R5 R0 A B SUBA Q Ch 0 0 0

R4 R4 A B AND Q Ch 0 0 0

R4 R1 A B SUBA Q Ch 0 0 0

R5 R5 A B XOR Q Ch 0 0 0

R1 R4 A B SUBB Q Ch 0 0 0

P8 DL DR - - C Ch X1 0 33

R0 R0 A B R Q Ch 0 0 0

R0 R2 A B R Q Ch 0 0 0

R0 R2 A B R L Ch 0 0 0

R0 R3 A B R RD Ch 0 0 0

R1 R4 A B SUBA Q Ch 0 0 0

P8 DL DR - - C Ch X1 0 28

R0 R3 A B R Q Ch 0 0 0

P8 DR - - C Ch 0 0 0

R6 R2 A B OR Q inc 1 0 30

P8 DR - - C Ch 0 0 0

R7 R2 A B AND Q inc 0 0 0

P8 DL DR - - C Ch X4 1 21

R5 R2 A B OR Q Ch 0 0 0

P8 DL DR - - C Ch 0 0 0

R1 R2 A B AND Q Ch 0 0 0