



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

---

Институт Информационных Технологий

Кафедра Вычислительной техники

---

## ОТЧЕТ О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ РАБОТЫ №2

«Проектирование синхронных цифровых автоматов»

по дисциплине  
«Теория автоматов»

Выполнил студент группы ИВБО-10-23

*Кудряшов Д.С.*

Принял старший преподаватель

*Боронников А.С.*

Практическая работа выполнена

«\_\_»\_\_\_\_\_ 2024 г.

«Зачтено»

«\_\_»\_\_\_\_\_ 2024 г.

Москва 2024

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 ПОСТАНОВКА ЗАДАЧИ И ПЕРСОНАЛЬНЫЙ ВАРИАНТ .....	4
2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ЦИФРОВЫХ АВТОМАТОВ .....	5
2.1 Проектирование и реализация цифрового автомата на основе условий задачи №1 .....	5
2.2 Проектирование и реализация цифрового автомата на основе условий задачи №2 .....	8
2.3 Проектирование и реализация цифрового автомата на основе условий задачи №3 .....	12
2.4 Проектирование и реализация цифрового автомата на основе условий задачи №4 .....	16
ЗАКЛЮЧЕНИЕ .....	18

# ВВЕДЕНИЕ

Целью данной работы является проектирование синхронных автоматов с учетом заданных параметров и требований к их функционированию. Основная задача состоит в разработке схем синхронных автоматов с использованием различных методов проектирования и моделирования, реализуемых в программной среде Logisim.

Процесс проектирования синхронного автомата определяется типом поставленной задачи. Этот процесс включает построение таблицы или графа переходов, описывающих зависимости между текущими состояниями и следующими состояниями автомата. Такая структура позволяет точно определить логику работы устройства и разработать управляющую схему. На основе графа переходов формируется таблица переходов, которая затем подвергается минимизации. Минимизация выполняется для сокращения количества используемых логических элементов и повышения эффективности схемы.

На завершающем этапе проектируется и реализуется схема синхронного автомата в лабораторной среде Logisim. Разработанная модель проходит тестирование для проверки её корректности и соответствия заданным параметрам.

# 1 ПОСТАНОВКА ЗАДАЧИ И ПЕРСОНАЛЬНЫЙ ВАРИАНТ

В рамках данной практической работы требуется выполнить четыре задания:

1. Спроектировать синхронный автомат, который повторяет на выходе входные значения, но только установившиеся, т.е. измеренные на фронте синхросигнала и совпадающие не менее двух раз подряд. Первое выходное значение совпадает с первым входным.
2. Спроектировать автомат с двухразрядным входом и одnorазрядным выходом, который отмечает событие, если строб поглотил хотя бы один строб на другой линии.
3. Спроектировать синхронный автомат, который вычисляет минимальное число из двух положительных чисел. Числа поступают одновременно по 2-разрядной шине (каждое число по своей 1 разрядной шине), начиная со старших разрядов, в сопровождении синхросигналов. На одnorазрядном выходе синхронно появляется результат: значение разрядов минимального из чисел.
4. Спроектировать автомат, который вычисляет свертку по mod 5 для положительного числа, поступающего последовательно по одному разряду, начиная с младшего. Текущее значение свертки присутствует на трехразрядном выходе.

## 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ЦИФРОВЫХ АВТОМАТОВ

### 2.1 Проектирование и реализация цифрового автомата на основе условий задачи №1

Для того, чтобы спроектировать синхронный автомат, работа которого описана в первом задании, составим граф переходов (Рисунок 2.1):

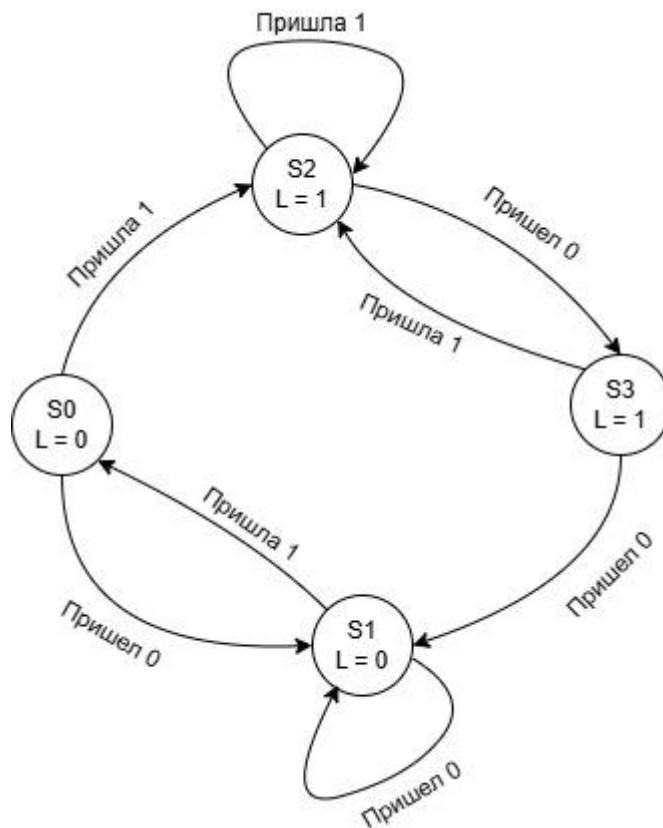


Рисунок 2.1 – Граф переходов исходного синхронного автомата

Далее, руководствуясь графом, составим таблицу переходов (Таблица 2.1) со следующими столбцами:

- $S$  – текущее состояние синхронного автомата;
- $T$  – входной сигнал;
- $S'$  – следующее состояние синхронного автомата.

Таблица 2.1 – Таблица переходов между состояниями синхронного автомата

S	T	S'
S0	0	S1
S0	1	S2
S1	0	S1
S1	1	S0
S2	0	S3
S2	1	S2
S3	0	S1
S3	1	S2

Для удобства внесения данных в ПЗУ, закодируем все устойчивые состояния синхронного автомата в Таблице 2.2:

Таблица 2.2 – Таблица кодирования устойчивых состояний синхронного автомата

S	Код
S0	00
S1	01
S2	10
S3	11

Руководствуясь этими данными, перепишем Таблицу 2.1 в ином виде, добавив столбец со значениями ПЗУ (Таблица 2.3):

Таблица 2.3 – Таблица переходов между состояниями синхронного автомата

S		T	S'		ПЗУ
S1	S0	T	S'1	S'0	
0	0	0	0	1	1
0	0	1	1	0	2
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	1	1	3
1	0	1	1	0	2
1	1	0	0	1	1
1	1	1	1	0	2

Из Таблицы 2.3 составим СДНФ для функции состояния  $S'1$  (Формула 2.1):

$$S'1 = \overline{S1} \wedge \overline{S0} \wedge T \vee S1 \wedge \overline{S0} \wedge \overline{T} \vee S1 \wedge \overline{S0} \wedge T \vee S1 \wedge S0 \wedge T. \quad (2.1)$$

Из Таблицы 2.3 составим СДНФ для функции состояния  $S'0$  (Формула 2.2):

$$S'0 = \overline{S1} \wedge \overline{S0} \wedge \overline{T} \vee \overline{S1} \wedge S0 \wedge \overline{T} \vee S1 \wedge \overline{S0} \wedge \overline{T} \vee S1 \wedge S0 \wedge \overline{T}. \quad (2.2)$$

Далее спроектируем выходную комбинационную логику для выхода  $L$  в Таблице 2.4:

Таблица 2.4 – Таблица соответствия устойчивого состояния и выходного сигнала

S1	S0	L
0	0	0
0	1	0
1	0	1
1	1	1

Из Таблицы 2.4 составим СДНФ для функции выхода  $L$  (Формула 2.3):

$$L = S1 \wedge \overline{S0} \vee S1 \wedge S0. \quad (2.3)$$

Все необходимые данные для разработки автомата получены и будут применены при его реализации в лабораторном комплексе Logisim. Готовый синхронный автомат, удовлетворяющий условиям задания №1 представлен на Рисунке 2.2:

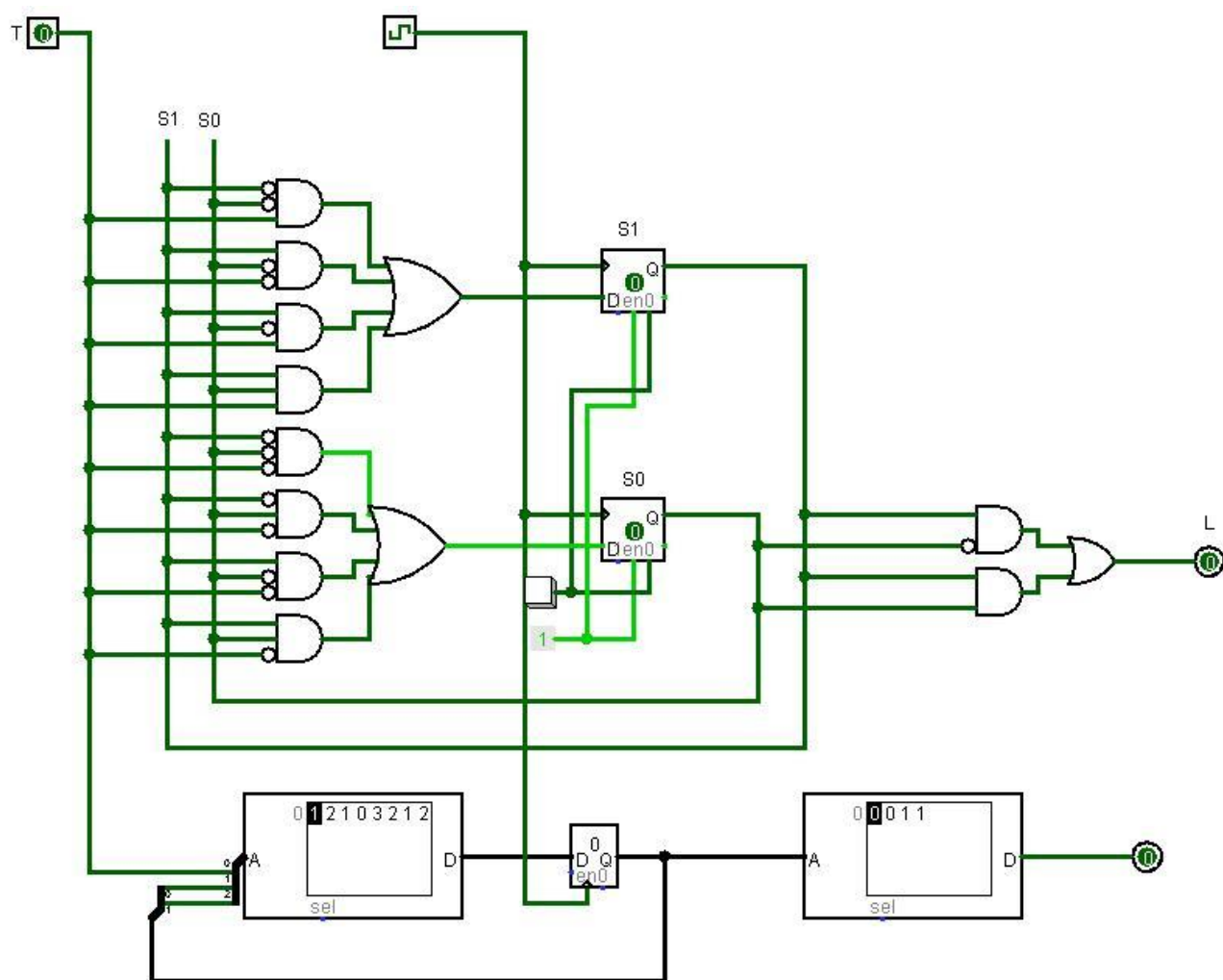


Рисунок 2.2 – Реализация синхронного автомата в среде моделирования Logisim

## 2.2 Проектирование и реализация цифрового автомата на основе условий задачи №2

Для того, чтобы спроектировать синхронный автомат, распознающий асинхронный язык, необходимо заранее найти все входные слова, которые необходимо распознать (Рисунок 2.3):

Кодировка символов:  $a = \frac{0}{0}$ ;  $b = \frac{0}{1}$ ;  $c = \frac{1}{0}$ ;  $d = \frac{1}{1}$ .

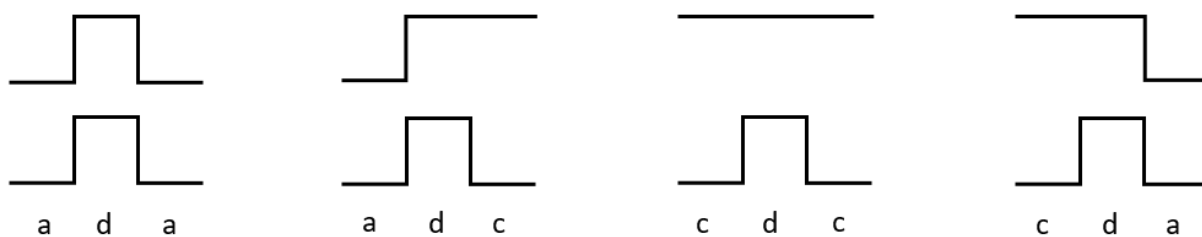


Рисунок 2.3 – Входные слова, которые необходимо распознать



Далее необходимо составить таблицу переходов между состояниями синхронного автомата, за основу которого будет взят автомат Милли, при поступлении нового символа (Таблица 2.5):

Таблица 2.5 – Таблица переходов между состояниями синхронного автомата

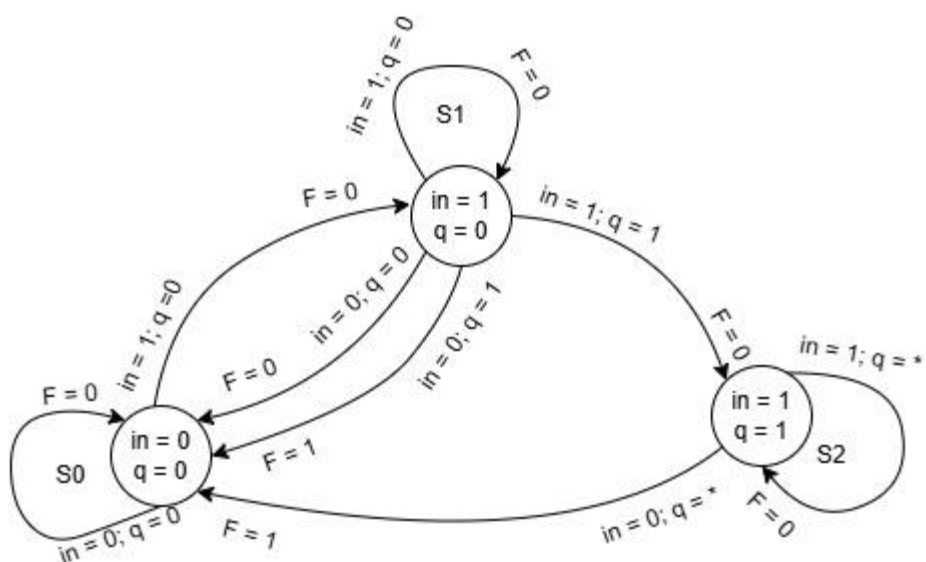
	a	b	c	d
$\lambda$	a	$\lambda$	c	$\lambda$
a	\$	$\lambda$	$\lambda$	ad
c	$\lambda$	$\lambda$	\$	cd
ad	$a/1$	$\lambda$	$c/1$	\$
cd	$a/1$	$\lambda$	$c/1$	\$

Тогда исходные состояния  $Q_i$  есть:  $Q_0 = \lambda$ ;  $Q_1 = a$ ;  $Q_2 = c$ ;  $Q_3 = ad$ ;  $Q_4 = cd$ . Перерисуем исходной Таблицу 2.5 с учетом переобозначений (Таблица 2.6):

Таблица 2.6 – Таблица переходов между состояниями синхронного автомата

	a	b	c	d
$Q_0$	$Q_1$	$Q_0$	$Q_2$	$Q_0$
$Q_1$	$Q_1$	$Q_0$	$Q_0$	$Q_3$
$Q_2$	$Q_0$	$Q_0$	$Q_2$	$Q_4$
$Q_3$	$Q_1/1$	$Q_0$	$Q_2/1$	$Q_3$
$Q_4$	$Q_1/1$	$Q_0$	$Q_2/1$	$Q_4$

Для того, чтобы отследить событие, когда строб поглотил хотя бы один строб на другой линии, необходимо спроектировать еще один синхронный автомат. Спроектируем его с помощью графа переходов (Рисунок 2.4):



**Рисунок 2.4 – Граф переходов для отслеживания события поглощения одного строба другим на другой линии**

По данному графу переходов восстановим таблицу переходов для исходного синхронного автомата (Таблица 2.7):

*Таблица 2.7 – Таблица переходов между состояниями синхронного автомата*

S	in	q	S'
$S_0$	0	0	$S_0/0$
$S_0$	0	1	$x/0$
$S_0$	1	0	$S_1/0$
$S_0$	1	1	$x/0$
$S_1$	0	0	$S_0/0$
$S_1$	0	1	$S_0/1$
$S_1$	1	0	$S_1/0$
$S_1$	1	1	$S_2/0$
$S_2$	0	X	$S_0/1$
$S_2$	0		$S_0/1$
$S_2$	1	X	$S_2/0$
$S_2$	1		$S_2/0$

В данной таблице  $S$  – текущее состояние синхронного автомата,  $S'$  – следующее состояние,  $in$  – состояние сигнала поглощающей линии,  $q$  – состояние сигнала поглощения.

Исходя из полученных данных в процессе проектирования были составлены следующие синхронные автоматы:

- синхронный автомат, распознающий необходимые слова асинхронного языка (Рисунок 2.5);
- синхронный автомат, распознающий событие поглощения одного строба другим на другой линии (Рисунок 2.6).

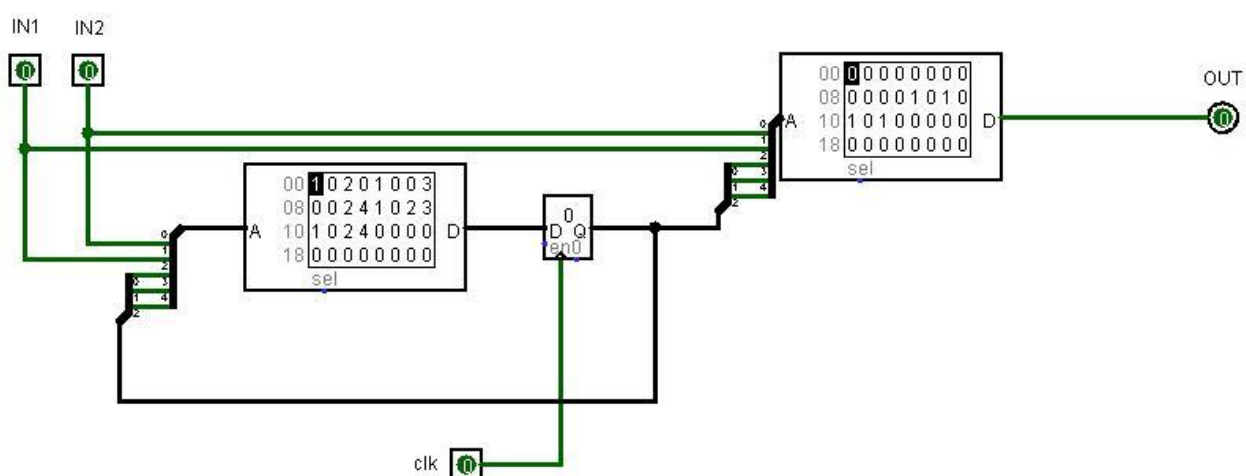


Рисунок 2.5 – Синхронный автомат, необходимые слова асинхронного языка

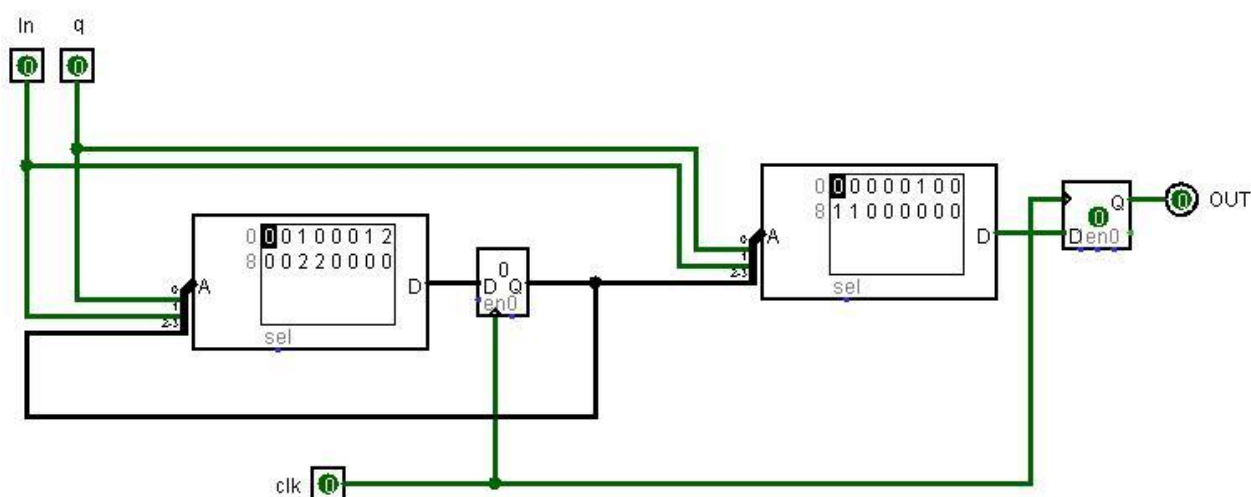


Рисунок 2.6 – Синхронный автомат, распознающий событие поглощения одного строба другим на другой линии

Совместив данные автоматы воедино, получим синхронный автомат, который функционирует согласно заданным условиям (Рисунок 2.7):

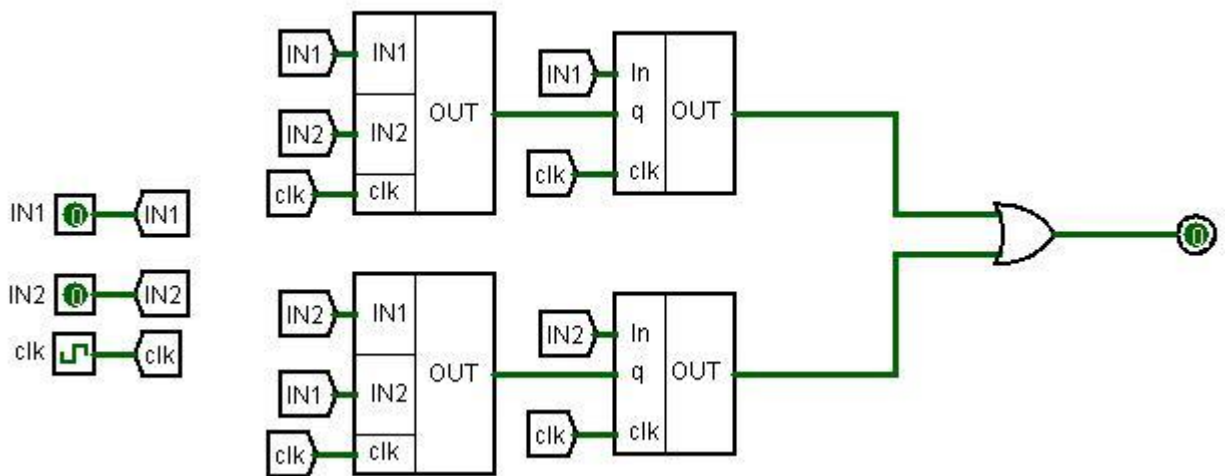


Рисунок 2.7 – Реализация исходного синхронного автомата в среде моделирования Logisim

## 2.3 Проектирование и реализация цифрового автомата на основе условий задачи №3

Для того, чтобы спроектировать синхронный автомат, работа которого описана в третьем задании, для каждого отдельного числа необходимо построить локальный автомат со следующими состояниями:

- М – число минимальное - значение на выходе равно входному;
- N – число не минимальное - на выходе автомата 1, т.е. наибольшее значение.

Локальный автомат переходит из состояния М в состояние N тогда, когда число перестает быть минимальным, то есть когда значение входного разряда числа больше значения на выходе конъюнкции. Выходное значение в состоянии М – равно входному, в состояние N – максимально возможному, то есть 1.

Выход с конъюнкции  $m$ , а значит и выход  $q_i$  (выход локального автомата), должен зависеть без сдвига от переменной  $d_i$  (входного сигнала). В то же время выход автомата  $q_i$  должен зависеть со сдвигом от переменной  $m$ , поскольку это переменная обратной связи. Поэтому локальный автомат должен быть автоматом Мили.

Для того, чтобы спроектировать данный автомат, составим автоматную таблицу (Рисунок 2.8):

$\begin{array}{c} d \\ \diagdown \\ S \\ \diagup \\ m \end{array}$	00	01	11	10
M	0,M	X,X	1,M	1,N
N	1,N	1,N	1,N	1,N

**Рисунок 2.8 – Автоматная таблица исходного синхронного автомата**

В данной таблице указаны два значения:

- значение сигнала  $q_i$ ;
- $g$  – функция перехода в следующее состояние автомата (M или N).

Далее вместо состояний M и N подставим числовые значения 0 и 1 соответственно (Рисунок 2.9):

$\begin{array}{c} d \\ \diagdown \\ S \\ \diagup \\ m \end{array}$	00	01	11	10
0	0,0	X,X	1,0	1,1
1	1,1	1,1	1,1	1,1

**Рисунок 2.9 – Автоматная таблица исходного синхронного автомата**

Разобьем данную автоматную таблицу на две и минимизируем каждую функцию ( $q$  и  $g$ ) (Рисунок 2.10):

d \ m S	00	01	11	10
	0	X	0	1
1	1	1	1	1

Рисунок 2.10 – Карта Карно для функции S

Из Рисунка 2.10 выпишем формулу МДНФ функции g – перехода в следующее состояние S исходного синхронного автомата (Формула 2.4):

$$g = S \vee d \wedge \bar{m}. \quad (2.4)$$

Аналогичные действия проведем для функции выхода q (Рисунок 2.11):

d \ m S	00	01	11	10
	0	X	1	1
1	1	1	1	1

Рисунок 2.11 – Карта Карно для функции q

Из Рисунка 2.11 выпишем формулу МКНФ функции выхода q исходного синхронного автомата (Формула 2.5):

$$q = S \vee d. \quad (2.5)$$

После нахождения всех необходимых параметров можно приступить к построению автомата в лабораторном комплексе Logisim. Результаты построения приведены на Рисунках 2.12, 2.13.

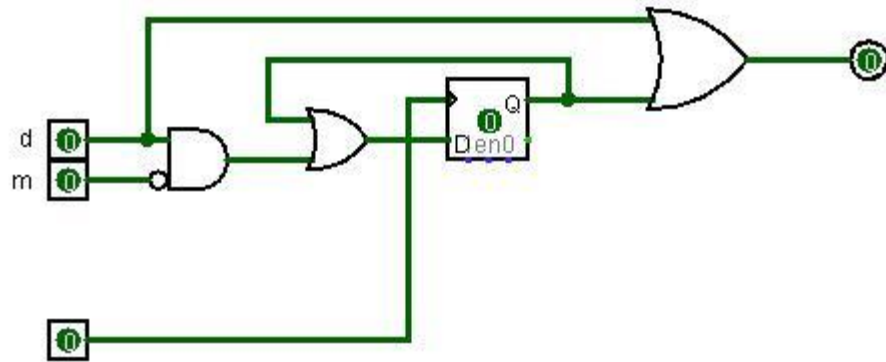


Рисунок 2.12 – Схема локального автомата

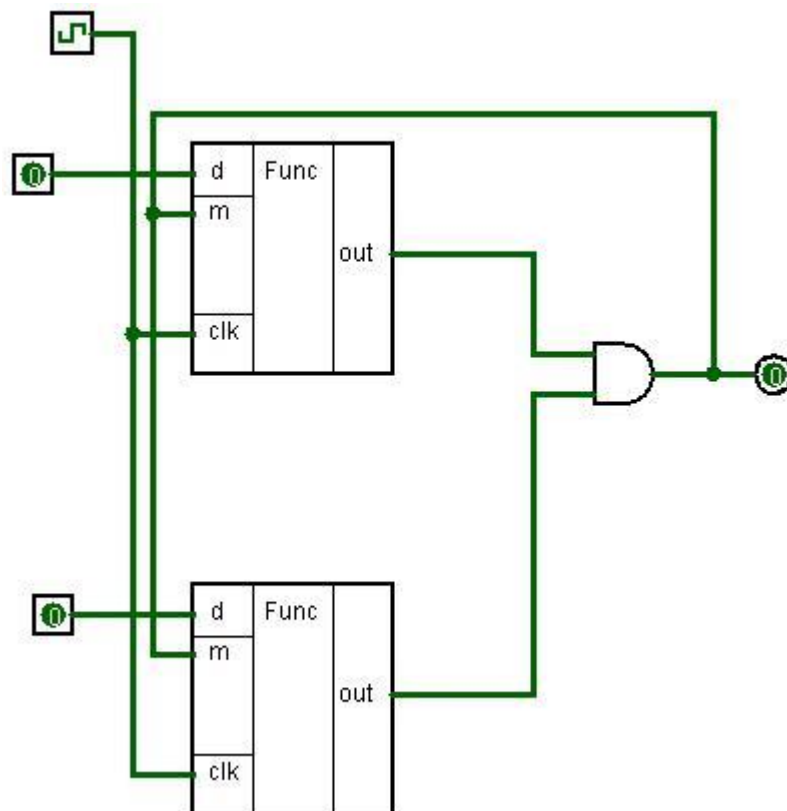


Рисунок 2.13 – Схема синхронного автомата поиска минимального значения

## 2.4 Проектирование и реализация цифрового автомата на основе условий задачи №4

Для реализации синхронного автомата, работа которого описана в четвертом задании, необходимо найти алгоритм нахождения остатка от деления на число  $X$  в двоичной системе счисления. Для этого необходимо разложить число на группы, длина которых соответствует числу  $s$  по Формуле 2.6:

$$(2^{s^k}) \bmod(X) = 1. \quad (2.5)$$

Воспользовавшись Формулой 2.6 для  $\bmod(5)$  получим число  $s = 4$ .

Для определения разряда в группе в проектируемом автомате дополнительно следует поставить на вход счетчик  $st$  по модулю  $s$ . Количество состояний есть количество вариантов выражения  $X \bmod 5$  ( $5: 0, 1, 2, 3, 4$ ). Если на вход  $in$  поступает «0» – свертка не изменяется –  $+0$ . Если на вход  $in$  поступает значение «1», то:

- если очередной разряд на нулевом разряде группы ( $st = 0$ ), то свертка увеличивается на вес разряда –  $+1 \cdot 2^0 (+1)$ ;
- если очередной разряд на первом разряде группы ( $st = 1$ ), то свертка увеличивается на вес разряда –  $+1 \cdot 2^1 (+2)$ .

Руководствуясь данными правилами, составим автоматную таблицу (Рисунок 2.14):

$\begin{matrix} in \\ st \\ Q \end{matrix}$	000	001	010	011	100	101	110	111
0000	0	0	0	0	1	2	4	3
0001	1	1	1	1	2	3	0	4
0010	2	2	2	2	3	4	1	0
0011	3	3	3	3	4	0	2	1
0100	4	4	4	4	0	1	3	2

Рисунок 2.14 – Автоматная таблица исходного синхронного автомата



Исходя из автоматной таблицы, реализуем данный синхронный автомат в лабораторном комплексе Logisim с помощью ПЗУ (Рисунок 2.15):

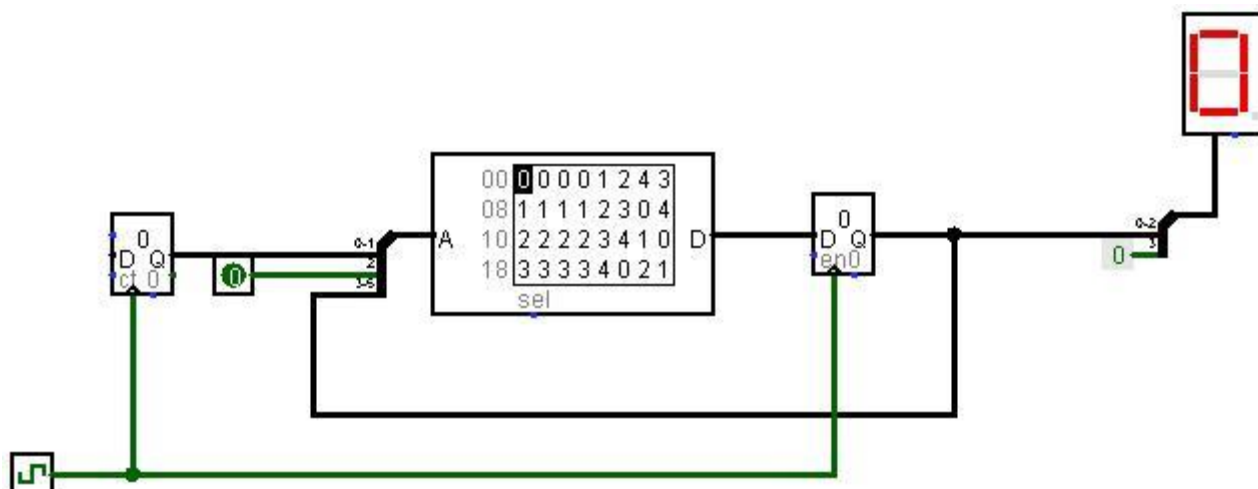


Рисунок 2.15 – Реализация синхронного автомата, выполняющего свертку по mod 5

## ЗАКЛЮЧЕНИЕ

В ходе выполнения данной практической работы были спроектированы и реализованы четыре синхронных цифровых автомата, соответствующих условиям заданий, описанных в первой главе.

Каждый автомат был реализован с помощью графов или таблиц перехода между состояниями (Таблицы 2.1–2.7, Рисунки 2.1, 2.4, 2.8 – 2.11, 2.14). Все автоматы были реализованы в лабораторном комплексе Logisim.

Результаты проектирования включают:

1. Автомат, повторяющий установившиеся входные значения, реализованный на основе графа переходов и таблицы состояний (Рисунок 2.2).
2. Автомат для распознавания событий поглощении одного строба другим на другой линии (Рисунки 2.5–2.7).
3. Автомат для вычисления минимального значения из двух чисел, включающий локальный автомат и итоговую реализацию (Рисунки 2.12–2.13).
4. Автомат для вычисления свертки по mod 5, реализованный с использованием ПЗУ (Рисунок 2.15).