

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И УПРАВЛЕНИЯ В ТЕХНИЧЕСКИХ СИСТЕМАХ

Кафедра информатики и управления в технических системах

**МЕТОДЫ АНАЛИЗА И СИНТЕЗА
ЦИФРОВЫХ ЭЛЕМЕНТОВ
СИСТЕМ АВТОМАТИКИ.
ЛАБОРАТОРНЫЙ ПРАКТИКУМ
ПО ДИСЦИПЛИНЕ**

Методические указания
к выполнению лабораторных работ
для студентов направления
27.03.04 – Управление в технических системах



Севастополь
СевГУ
2022

УДК 004.312:519.713(076.5)
ББК 32.815я73
Л125

Р е ц е н з е н т :

В. В. Альчаков – канд. техн. наук, доцент кафедры
"Информатика и управление в технических системах"

Составители: А. И. Грушун, Т. А. Грушун

Л125 Методы анализа и синтеза цифровых элементов систем автоматизации. Лабораторный практикум по дисциплине : методические указания к выполнению лабораторных работ для студентов направления 27.03.04 - Управление в технических системах / Севастопольский государственный университет, Институт информационных технологий и управления в технических системах, кафедра Информатики и управления в технических системах ; сост.: А. И. Грушун, Т. А. Грушун. – Севастополь : СевГУ, 2022. – 48 с. – Текст : электронный.

Цель лабораторного практикума: обеспечить качественное усвоение студентами теоретических положений курса для наиболее полного овладения основами математического и прикладного аппарата анализа и синтеза цифровых элементов автоматизации и робототехники. Каждая лабораторная работа, кроме теоретических сведений и указаний по ее выполнению, содержит требования к оформлению и содержанию отчета, а также перечень контрольных вопросов.

Лабораторный практикум предназначен для студентов направления подготовки 27.03.04 – Управление в технических системах.

УДК 004.312:519.713(076.5)
ББК 32.815я73

Лабораторный практикум рассмотрен и утвержден на заседании кафедры "Информатика и управление в технических системах", протокол № 7 от 23 марта 2022 г.

Лабораторный практикум рассмотрен и рекомендован к изданию на заседании ученого совета Института информационных технологий и управления в технических системах, протокол № 6 от 18 апреля 2022 г.

СОДЕРЖАНИЕ

Введение	4
1. Лабораторная работа № 1. Арифметические операции в цифровых устройствах	4
2. Лабораторная работа № 2. Синтез комбинационной схемы	10
3. Лабораторная работа № 3. Абстрактный анализ конечного автомата	19
4. Лабораторная работа № 4. Структурный синтез конечного автомата	32
Библиографический список	45
Приложение А	46

ВВЕДЕНИЕ

Методические указания включают четыре лабораторные работы, охватывающие основные разделы изучаемой дисциплины. В процессе подготовки к выполнению работы студент должен изучить основные теоретические положения, ознакомиться с описанием используемого программного продукта и выполнить предварительные теоретические расчеты.

Лабораторные работы позволяют более глубоко изучить арифметические основы цифровых элементов систем автоматики, логические функции и синтез комбинационных схем, анализ и синтез цифровых автоматов на абстрактном и структурном уровне, получить практические навыки исследования процессов в цифровых устройствах.

1. ЛАБОРАТОРНАЯ РАБОТА № 1 АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ В ЦИФРОВЫХ УСТРОЙСТВАХ

1.1. Цель работы

Освоить перевод чисел в двоичную систему счисления и основные алгебраические операции с ними.

1.2. Теоретические сведения

Цифровые устройства работают с двухуровневыми сигналами. Обозначим их U_0 и U_1 . Считается, что физические сигналы U_0 и U_1 представляют цифры 0 и 1, т.е. напряжению U_0 сопоставлена цифра 0, U_1 – цифра 1. Таким образом, вместо привычных цифр от 0 до 9, при описании работы цифровых устройств, используются только две цифры 0 и 1.

Для представления чисел и их преобразования (суммирования, вычитания, умножения, перевода в другие системы и др.) используются те же понятия и приемы, которые были усвоены при изучении арифметики и применяются в обыденной жизни, только во всех этих преобразованиях участвуют две цифры 0 и 1. Использование этих принципов составляет содержание основ выполнения арифметических операций в цифровых устройствах [3, 6].

Позиционной системой счисления называется совокупность понятий и приемов представления и преобразования чисел, причем значимость цифры в числе определяется положением (позицией) этой цифры в числе. Например: $91=90+1$, $19=10+9$.

Цифры 9 и 1 в зависимости от места в числах 91 и 19 представляют разные числовые значения.

Будем рассматривать смешанные дроби с точкой, разделяющей целую и дробную части числа. Позиции слева и справа от точки пронумерованы натуральными числами. Влево от точки – числами 0, 1, 2, ..., справа – отрицательными числами -1, -2, Цифру на i -ом месте обозначим через a_i .

«целая часть» . «дробная часть»

$$\begin{array}{ccccccc} n & \dots & 2 & 1 & 0 & . & -1 & -2 & -3 & \dots & -k \\ a_n & \dots & a_2 & a_1 & a_0 & . & a_{-1} & a_{-2} & a_{-3} & \dots & a_{-k} \end{array}$$

Основанием системы счисления называется число p различных цифр в ней. Используются цифры $0, 1, \dots, p-1$.

$$\begin{array}{l} p=10 \quad a_i \in \{0, 1, \dots, 9\}; \\ p=2 \quad a_i \in \{0, 1\}. \end{array}$$

Система с основанием p называется p -ной (десятичной, двоичной, восьмеричной, шестнадцатеричной).

Весомозначная позиционная система сопоставляет цифре, стоящей в позиции i , количество $a_i p^i$, например: $19 = 1 \cdot 10^1 + 9 \cdot 10^0$.

Система называется аддитивной, если общее количество, соответствующее числу, вычисляется как сумма количеств, сопоставленных его цифрам, что удовлетворяет следующей формуле:

$$a_n \dots a_0 . a_{-1} \dots a_{-k} = \sum_{i=k}^{i=n} a_i p^i, \quad a_i \in \{0, \dots, p-1\}.$$

Эту формулу обычно называют формулой значимости. Она является фундаментальной формулой описания числа.

В обычной жизни используется десятичная ($p=10$) позиционная весомозначная аддитивная система.

Пользуясь формулой значимости, можно легко перевести небольшое десятизначное число в произвольную p -ную систему счисления, подбирая коэффициенты a_i при p^i . Особенно просто переводить в двоичную систему, так как коэффициенты a_i в этой системе могут быть равными только 1 или 0, и достаточно просто исходное десятичное число представить в виде суммы степеней двойки. Принято все вычисления в формуле значимости выполнять в десятичной системе. Пример:

$$27 = 16 + 11 = 2^4 + 8 + 3 = 2^4 + 2^3 + 2 + 1 = 2^4 + 2^3 + 2^1 + 2^0 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 11011.$$

Десятичная система удобна и привычна человеку. Двоичная система удобна для описания работы цифровых устройств на уровне схем и алгоритмов. Восьми и шестнадцатеричные системы используются как промежуточные.

Переводом числа из p -ной системы счисления в q -ную назовем построение числа в q -ной системе, представляющего ту же величину, что и исходное число p -ной системы. Для выполнения этой операции число p -ной системы разбивают на целую часть и дробную, и каждую часть переводят отдельно.

Рассмотрим перевод целого числа. Число делим на q , получая остаток и целое частное. Остаток используется для формирования значения очередной цифры целой

части q -ного числа, начиная с младшей. Результат деления (частное) снова делят на q , получая очередные остаток и частное. Процесс продолжается до тех пор, пока очередное частное не станет равным нулю. Все операции осуществляются в r -ной системе счисления.

Например, переведем в двоичную систему число 23:

$$\begin{array}{r}
 \underline{23} \mid \underline{2} \\
 \underline{22} \quad \underline{11} \mid \underline{2} \\
 \mathbf{1} \quad \underline{10} \quad \underline{5} \mid \underline{2} \\
 \quad \mathbf{1} \quad \underline{4} \quad \underline{2} \mid \underline{2} \\
 \quad \quad \mathbf{1} \quad \underline{2} \quad \mathbf{1} \\
 \quad \quad \quad \mathbf{0}
 \end{array}$$

Получаем: $23=10111$.

Проверим результат в системе Matlab:

```
>> dec2bin(23)
ans =
10111
```

Арифметика двоичной системы определяется двумя таблицами – сложения (таблица 1.1) и умножения (таблица 1.2). Таким образом, все (любые) алгебраические действия с двоичными числами сводятся к операциям сложения и умножения.

Таблица 1.1- Таблица сложения

+	0	1
0	0	1
1	1	[/] 0

Таблица 1.2 – Таблица умножения

*	0	1
0	0	0
1	0	1

В таблице $1 \text{ } ^{/}0$ означает, что разряд суммы равен 0, однако возникает единица переноса в следующий (старший) разряд. При сложении очередных разрядов a_i и b_i двоичных чисел A и B , участвуют сами разряды и перенос p_{i-1} из предыдущего разряда. Формируются значение c_i разряда суммы и перенос p_i в следующий разряд.

Например, выполним сложение двух семиразрядных двоичных чисел A и B : $A=0110101$, $B=0001111$ (таблица 1.3).

Таблица 1.3 – Таблица сложения чисел A и B

A	0	1	1	0	1	0	1
B	0	0	0	1	1	1	1
P							
C	1	[/] 0	[/] 0	[/] 0	[/] 1	[/] 0	[/] 0

Проверим результат с помощью системы Matlab. Для этого создадим М-файл с необходимыми функциями [5]. Из основного окна Matlab командой File/New/M-File открываем новый файл. После чего в открывшемся окне М-файла сформируем программу (рисунок 1.1).

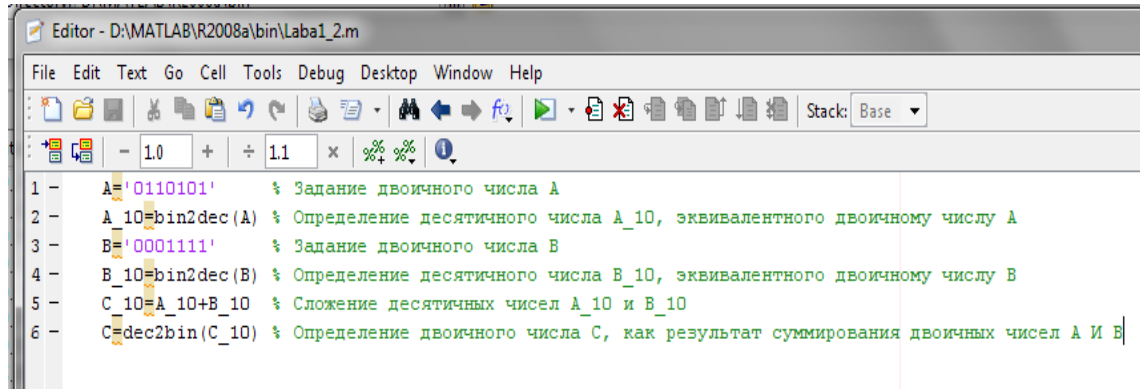


Рисунок 1.1 – Пример программы вычисления суммы двоичных чисел

При необходимости М-файл сохраняется. Нажатие на кнопку с изображением зеленого треугольника в окне М-файла произведет запуск процесса вычислений, после чего результат может быть проконтролирован в основном окне Matlab:

```

C =
1000100
>>.
  
```

Умножение двоичных чисел сводится к суммированию и сдвигу множимого, причем при умножении на 0 осуществляется только сдвиг, при умножении на 1 – сдвиг и суммирование множимого с текущей суммой. Исходная текущая сумма равна нулю.

Например, выполним умножение двух десятичных чисел $A=53$ и $B=15$ в двоичной форме. Переведем, с помощью системы Matlab, десятичные числа A и B в двоичную систему счисления:

```

>> dec2bin(53) % Перевод десятичного числа A в двоичную систему
ans =
110101
>>
>> dec2bin(15) % Перевод десятичного числа B в двоичную систему
ans =
1111
>>.
  
```

Таким образом, $A=110101$, $B=1111$. Число B располагается в четырех двоичных разрядах. Если его необходимо представить, например, в семи разрядах, то старшие (левые) три разряда заполняются нулями.

Процесс перемножения имеет вид:

$$\begin{array}{rcccccc}
 & & & 1 & 1 & 0 & 1 & 0 & 1 \\
 & & & & & & & & \\
 & & x & & & & & & \\
 & & & & & 1 & 1 & 1 & 1 \\
 & & & \hline
 & & & 1 & 1 & 0 & 1 & 0 & 1 \\
 & & + & & & & & & \\
 & & & 1 & 1 & 0 & 1 & 0 & 1 \\
 & & & \hline
 & & & 1 & 1 & 0 & 1 & 1 & 1 \\
 & & & /0 & /0 & 1 & 1 & 1 & 1 \\
 & & + & & & & & & \\
 & & & 1 & 1 & 0 & 1 & 0 & 1 \\
 & & & \hline
 & & & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & /0 & 1 & 1 & /1 & /0 & /0 \\
 & & + & & & & & & \\
 & & & 1 & 1 & 0 & 1 & 0 & 1 \\
 & & & \hline
 & & & 1 & 1 & 1 & 0 & 1 & 1 \\
 & & & /1 & /0 & /0 & /0 & 1 & 1
 \end{array}$$

Получаем: $C=A*B=795=1100011011$.

Проверим результат в системе Matlab:

```
>> dec2bin(795)
ans =
1100011011
```

1.3. Порядок теоретической подготовки

Предметом исследований, выполняемых в лабораторной работе, являются числа в двоичной системе счисления.

Программа теоретической подготовки:

- Изучить формы задания и представления чисел в цифровых устройствах, системы счисления.
- Изучить перевод чисел из десятичной системы счисления в другие.
- Изучить арифметику двоичной системы счисления.
- Изучить алгоритмы алгебраических операций с двоичными числами.

1.4. Порядок выполнения экспериментальных исследований

Выполнение расчетных операций производится с использованием ПЭВМ и системы научных и инженерных расчетов Matlab.

По заданным значениям десятичных чисел в соответствии с вариантом исходных данных (таблица 1.4), указанным преподавателем, необходимо:

- а) Провести перевод исходных десятичных чисел в двоичную систему счисления с помощью алгоритма, основанного на делении чисел.
- б) Выполнить проверку полученных двоичных чисел с помощью системы Matlab.
- в) Провести сложение двоичных чисел.
- г) Выполнить проверку результата суммирования с помощью системы Matlab.
- д) Провести перемножение двоичных чисел.
- в) Выполнить проверку результата перемножения с помощью системы Matlab.

1.5. Содержание отчета о выполнении лабораторной работы

Отчет должен содержать:

- а) Цель работы.
- б) Предмет исследования.
- в) Описание процесса перевода десятичных чисел в двоичную систему с проверкой в Matlab.
- г) Описание процесса сложения двоичных чисел с проверкой в системе Matlab.
- д) Описание процесса перемножения двоичных чисел с проверкой в системе Matlab.
- е) Развернутые комментарии к полученным результатам.
- ж) Краткие выводы.

Таблица 1.4 – Варианты исходных десятичных чисел

№ вар.	Значения десятичных чисел		№ вар.	Значения десятичных чисел	
	А	В		А	В
1	315	237	14	312	215
2	349	216	15	347	226
3	378	209	16	335	234
4	324	219	17	309	241
5	356	247	18	302	257
6	341	231	19	327	253
7	317	227	20	347	268
8	364	264	21	351	275
9	385	253	22	369	286
10	358	278	23	372	254
11	346	285	24	328	243
12	387	291	25	305	207
13	321	217			

1.6. Контрольные вопросы

1. Какая система счисления называется позиционной и почему?
2. Что понимают под основанием системы счисления?
3. Для чего нужна формула значимости?
4. Как перевести число из десятичной системы счисления в другие?
5. Какими таблицами определяется арифметика двоичной системы?
6. Опишите методику сложения двоичных чисел.
7. Опишите методику перемножения двоичных чисел.
8. Какой код называется прямым?
9. Какой код называется обратным?
10. Какой код называется дополнительным?
11. Как представить число в форме с фиксированной точкой?
12. Как представить число в форме с плавающей точкой?

2. ЛАБОРАТОРНАЯ РАБОТА № 2 СИНТЕЗ КОМБИНАЦИОННОЙ СХЕМЫ

2.1. Цель работы

Освоить синтез комбинационных схем с помощью минимизации логических функций на основе метода Квайна – Мак-Класки и их компьютерное моделирование.

2.2. Теоретические сведения

Существуют цифровые элементы (устройства) систем автоматики, в которых выходные сигналы в каждый момент времени однозначно определяются входными сигналами в этот же момент времени и не зависят от того, какие значения принимали входные сигналы в предшествующие моменты времени. В устройствах этого типа отсутствуют элементы памяти. Такие устройства получили название **комбинационные**. Они реализуются с помощью логических (комбинационных) схем, представляющих собой условное графическое обозначение логических функций. Логические функции при этом должны быть минимизированы.

Задача минимизации булевой функции состоит в том, чтобы найти эквивалентную ей формулу, имеющую минимальную сложность [1]. Под сложностью формулы понимают количество входящих в нее букв. Наиболее хорошо разработаны методы минимизации булевых функций, представленных в дизъюнктивной нормальной форме (ДНФ) [2, 4]. Эти методы основаны на понятии простой импликанты.

Определение 1. Если некоторая булева функция φ равна нулю на тех же наборах, на которых равна нулю другая функция F , а также и на некоторых других наборах, то говорят, что функция φ входит в функцию F и является ее импликантой. Условие вхождения записывается следующим образом: $\varphi \subset F$.

Например, рассмотрим функцию $F = X \oplus Y$. Ее импликантами являются функции $G_1 = X\bar{Y}$, $G_2 = \bar{X}Y$, $G_3 = 0$.

Определение 2. Простыми импликантами булевой функции F называют такие элементарные конъюнкции, которые сами входят в данную функцию, но ни какая собственная часть этих конъюнкций в функцию F не входит. Собственной частью называют произведение, полученное путем исключения из данного произведения одного или нескольких сомножителей.

Например, для $F = X \oplus Y$ $\bar{X}Y \subset F$, $X\bar{Y} \subset F$, $\bar{X} \not\subset F$, $Y \not\subset F$, $X \not\subset F$, $\bar{Y} \not\subset F$. Значит $\bar{X}Y$ и $X\bar{Y}$ являются простыми импликантами функции F .

Простые импликанты представляют собой самые короткие элементарные конъюнкции, входящие в данную булеву функцию.

Определение 3. Любая булева функция может быть представлена в виде дизъюнкции всех своих простых импликант. Дизъюнкция всех простых импликант называется сокращенной ДНФ булевой функции.

Существует несколько методов (алгоритмов) получения сокращенной ДНФ булевой функции. Одним из наиболее хорошо разработанных является метод Квайна – Мак-Класки.

Теорема Квайна. Если в совершенной ДНФ (СДНФ) булевой функции провести все операции неполного склеивания, а затем все операции поглощения, то получится сокращенная ДНФ этой функции, т.е. дизъюнкция всех ее простых импликант.

Операция склеивания может применяться не только к соотношениям, но и к наборам: склеиваются два набора, если множества их переменных совпадают, и значение одной и только одной переменной различается. Результат склеивания двух наборов изображается в виде набора, у которого вместо значения различающейся переменной проставляется черточка:

$$\left. \begin{array}{cccc} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{array} \right\} = 0 \ 0 \ 1 \ -.$$

Также, булеву функцию можно представить в виде совокупности интервалов. Например, функция $f(x_1, x_2, x_3, x_4) = x_1 \bar{x}_2 \vee \bar{x}_3 x_4$ соответствует совокупности двух интервалов:

$$\begin{array}{cccc} 1 & 0 & - & - \\ - & - & 0 & 1 \end{array}$$

Минимизация булевой функции осуществляется следующим образом. Сначала получают сокращенную ДНФ, а затем ее упрощают до минимальной [1].

Мак-Класки предложил метод (алгоритм) сокращения перебора при проведении операций склеивания и поглощений на основе использования интервальной формы и разбиения специального множества M_1 на классы M_1^i ($0 \leq i \leq n$). M_1^i содержит наборы веса i . Основу метода составляют пять следующих положений:

1. Склеиваться могут наборы только соседних классов M_1^i и M_1^{i+1} ($0 \leq i \leq n-1$).
2. Если организовать процесс склеиваний так, что результаты склеиваний наборов помещаются в новое множество, то наборы исходного множества и нового множества не склеиваются. При склеивании наборов множеств M_1^i и M_1^{i+1} получается новое множество M_1^{in} наборов, содержащих i единиц.
3. Если наборы склеиваются, то они могут участвовать в других склеиваниях, но должны считаться поглощенными.
4. Если наборы исходного множества не склеивались, то они уже не будут поглощаться, т.е. соответствующие им конъюнкции являются простыми импликантами и должны быть включены в сокращенную ДНФ.
5. Новое множество результатов склеиваний может в следующей итерации считаться исходным и с наборами этого множества снова нужно производить все возможные склеивания.

Алгоритм построения множества наборов, соответствующих конъюнкциям сокращенной ДНФ (назовем эти наборы максимальными).

1. По исходной (подлежащей минимизации) СДНФ записывается множество M_1 и разбивается на классы M_1^i ($0 \leq i \leq n$). Исходное множество M_p максимальных наборов пусто.

2. Производятся все возможные склеивания наборов множеств M_1^i и M_1^{i+1} . Склеиваемые наборы отмечаются, результат склеивания записывается с приведением подобных в множество M_1^{iH} . Пункт 2 выполняется для $i=0, 1, \dots, n-1$.

3. Если во множествах $M_1^i (0 \leq i \leq n)$ есть неотмеченные наборы, то они вносятся во множество M_p максимальных наборов.

4. Если множества M_1^{iH} не пусты, то индекс «н» уничтожается и для них выполняются пп. 2-4, иначе множество максимальных наборов построено.

Рассмотрим пример. Исходная СДНФ имеет вид:

$$f(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 \vee x_1\bar{x}_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1\bar{x}_2\bar{x}_3x_4 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee \bar{x}_1x_2x_3\bar{x}_4 \vee x_1\bar{x}_2\bar{x}_3x_4 \vee \bar{x}_1x_2\bar{x}_3x_4 \vee x_1\bar{x}_2x_3x_4 \vee x_1x_2\bar{x}_3x_4 \vee x_1x_2x_3x_4.$$

Получение сокращенной ДНФ представлено в таблице 2.1.

Таблица 2.1 – Получение сокращенной ДНФ

Итерация 1				Итерация 2					Итерация 3				
Класс (число единиц в наборе)	Наборы	Текущий номер набора	Признак склеивания набора (∨)	Класс (число единиц в наборе)	Наборы	Показатель набора (k,m), где k и m – номера склеившихся наборов на предыдущей итерации	Текущий номер набора	Признак склеивания набора (∨)	Класс (число единиц в наборе)	Наборы	Показатель набора (k,m), где k и m – номера склеившихся наборов на предыдущей итерации	Текущий номер набора	Признак склеивания набора (∨)
0	0000	1	∨	0	-000 000-	(1,2) (1,3)	1* 2*	∨ ∨	0	- 00 -	(1*,5*), (2*,4*)	1**	-
1	1000 0001	2 3	∨ ∨	1	10-0 100- -001 0-01	(2,4) (2,6) (3,6) (3,7)	3* 4* 5* 6*	∨ ∨ ∨ ∨	1	10 - - - - 01	(3*,8*), (4*,7*) (5*,10*), (6*,9*)	2** 3**	- -
2	1010 0110 1001 0101	4 5 6 7	∨ - ∨ ∨	2	101- 10-1 1-01 -101	(4,8) (6,8) (6,9) (7,9)	7* 8* 9* 10*	∨ ∨ ∨ ∨	2	1 - - 1	(9*,11*), (8*,12*)	4**	-
3	1011 1101	8 9	∨ ∨	3	1-11 11-1	(8,10) (9,10)	11* 12*	∨ ∨		M ^{2H} ₁			
4	1111	1 0	∨		M ^{1H} ₁								
	M ₁												

Чтобы проиллюстрировать процесс склеивания, наборы множества M_1 пронумерованы и рядом с результатами склеиваний записаны пары наборов, участвующие в склеивании. В результате склеиваний, на третьей итерации, получили множество M_1^{2H} , в котором уже наборы между классами не склеиваются. Это говорит о завершении процесса. На первой итерации набор 0110 не склеился ни

с каким набором из соседних классов, следовательно, он сразу заносится в множество M_p максимальных наборов.

В итоге, множество M_p будет содержать пять наборов:

$$M_p = \{0\ 1\ 1\ 0, -0\ 0-, 1\ 0\ -\ -, -\ -\ 0\ 1, 1\ -\ -1\}$$

Сокращенная ДНФ содержит пять конъюнкций:

$$f_{\text{сокр.ДНФ}} = \bar{x}_1 x_2 x_3 \bar{x}_4 \vee \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 \vee \bar{x}_3 x_4 \vee x_1 x_4.$$

Для того, чтобы найти минимальную дизъюнктивную форму, можно воспользоваться импликантной матрицей. В импликантной матрице столбцы помечаются конституентами единицы заданной функции (исходной СДНФ), а строки – импликантами, полученными в результате склеивания (таблица 2.2).

Таблица 2.2 – Импликантная матрица

Простые импликанты	Конституенты единицы									
	$\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$	$x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$	$\bar{x}_1 \bar{x}_2 \bar{x}_3 x_4$	$x_1 \bar{x}_2 \bar{x}_3 x_4$	$\bar{x}_1 x_2 \bar{x}_3 \bar{x}_4$	$x_1 x_2 \bar{x}_3 \bar{x}_4$	$\bar{x}_1 x_2 \bar{x}_3 x_4$	$x_1 x_2 \bar{x}_3 x_4$	$\bar{x}_1 \bar{x}_2 x_3 \bar{x}_4$	$x_1 \bar{x}_2 x_3 \bar{x}_4$
$\bar{x}_1 x_2 x_3 \bar{x}_4$					*					
$\bar{x}_2 \bar{x}_3$	*	*	*			*				
$x_1 \bar{x}_2$		*		*		*		*		
$\bar{x}_3 x_4$			*			*	*		*	
$x_1 x_4$						*		*		*

Если простая импликанта является собственной частью некоторой конституенты единицы, то клетка матрицы, соответствующей этой импликанте и конституенте единицы, помечается звездочкой. Чтобы получить минимальную ДНФ заданной функции, нужно найти минимальное число импликант, которые совместно накрывают звездочками все колонки импликантной матрицы. В приведенном примере минимальная дизъюнктивная форма заданной функции имеет вид:

$$f_{\text{мин.ДНФ}} = \bar{x}_1 x_2 x_3 \bar{x}_4 \vee \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 \vee \bar{x}_3 x_4 \vee x_1 x_4$$

и не отличается от сокращенной ДНФ.

Для полученной минимальной ДНФ составим логическую (комбинационную) схему, представляющую собой условное графическое обозначение логических функций. На рисунке 2.1 показаны графические обозначения некоторых элементарных логических функций. Блоки, обозначенные символом «1», выполняют операцию логического сложения (логическое ИЛИ, OR), а блоки, обозначенные символом «&» - операцию логического умножения (логическое И, AND).

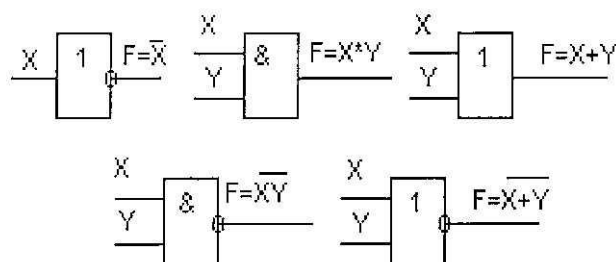


Рисунок 2.1 – Графическое обозначение элементарных логических функций

На рисунке 2.2 показана логическая схема для полученной минимальной ДНФ $f_{\text{мин.ДНФ}}$.

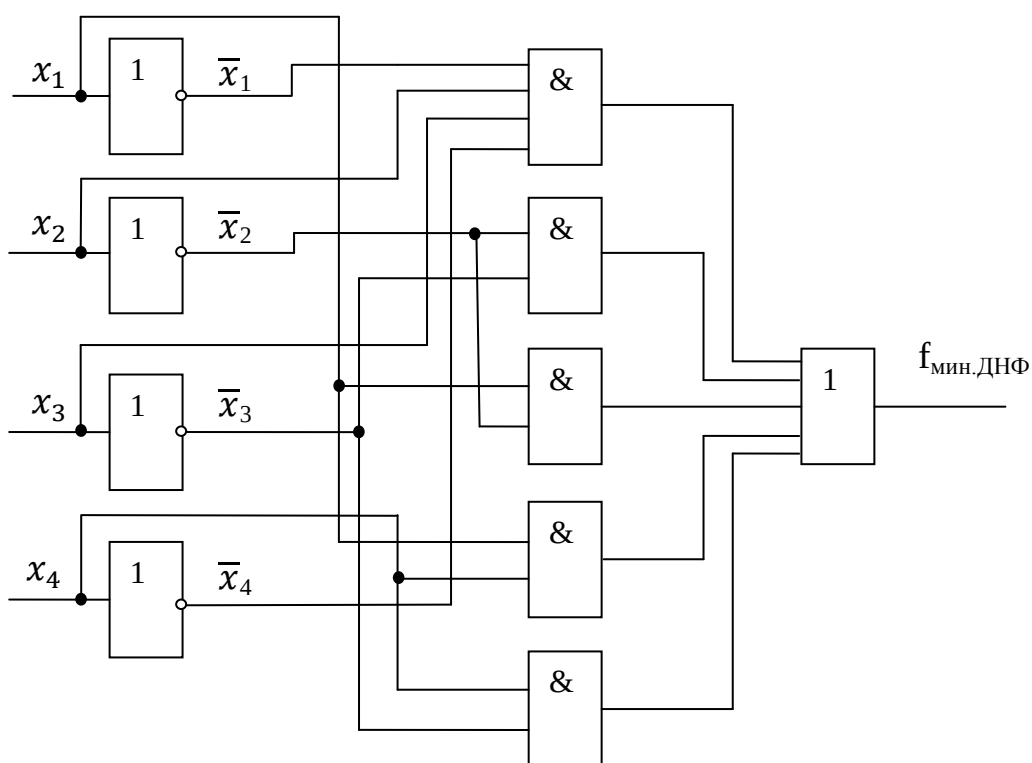


Рисунок 2.2 – Логическая схема для минимальной ДНФ $f_{\text{мин.ДНФ}}$

2.3. Порядок теоретической подготовки

Объектом исследований, выполняемых в лабораторной работе, является булева функция, заданная в совершенной дизъюнктивной нормальной форме.

Программа теоретической подготовки:

- Изучить элементарные логические функции.
- Изучить способы задания и формы представления логических функций.
- Изучить метод Квайна – Мак-Класки минимизации булевых функций.

- г) Изучить систему элементов, из которых должны строиться комбинационные (логические) схемы.
- д) Изучить методику построения и компьютерного моделирования комбинационных (логических) схем.

2.4. Порядок выполнения экспериментальных исследований

По заданной в СДНФ булевой функции в соответствии с вариантом исходных данных (таблица 2.3), указанным преподавателем, необходимо:

- а) Получить с помощью метода Квайна – Мак-Класки сокращенную ДНФ булевой функции.
- б) Получить с помощью импликантной матрицы минимальную ДНФ.
- в) Составить логическую схему для исходной СДНФ и полученной минимальной ДНФ.
- г) Выполнить при нескольких наборах входных переменных моделирование исходной СДНФ и полученной минимальной ДНФ в системе Matlab. Сравнить результаты их моделирования при одинаковых входных данных. Сравнить количество элементов и блоков Matlab, используемых для моделирования исходной и минимальной булевой функции.

Моделирование производится непосредственно по составленной логической схеме с помощью подсистемы Simulink пакета Matlab [5]. Для этого необходимо из основного окна Matlab запустить Simulink и командой File/New/Model создать новый файл модели. После чего в окне модели сформировать схему модели перемещением с помощью компьютерной мыши необходимых блоков и элементов из соответствующих библиотек, которые соединяются между собой, как того требует схема. Для формирования модели использовать следующие элементы библиотек блоков Simulink (Simulink Library Browser):

- в библиотеке Simulink/Sources (источники сигнала) – блок источника постоянного сигнала Constant, которым задается логическая единица или логический ноль. В блоке Constant, в меню Signal Attributes, во вкладке Output data type выбрать Boolean. Параметр Constant value задается равным или единице (для логической единицы) или нулю (для логического нуля);
- в библиотеке Simulink/Sinks – осциллограф Scope для построения графиков моделируемых процессов (визуальной фиксации значения булевой функции);
- в библиотеке Simulink/Logic and Bit Operations – блок логических операций Logical Operator. Для того, чтобы выбрать необходимую логическую операцию необходимо переместить блок Logical Operator в окно модели и дважды щелкнуть по нему левой кнопкой мыши. После чего в меню Signal Attributes, во вкладке Output data type выбрать Boolean. В меню Main, во вкладке Operator выбрать требуемую логическую операцию (AND, OR, NOT и т.д.), а во вкладке Number of input ports – необходимое число входов логического блока.
- в библиотеке Simulink/Signal Routing – блок переключателя сигналов Manual Switch для задания логической единицы или логического нуля в комбинации набора входных переменных (сигналов).

При необходимости файл модели сохраняется.

Выбор опции меню Simulation/Start (Моделирование/Старт) или нажатие на кнопку с изображением черного треугольника в окне моделирования произведет запуск процесса моделирования, после чего результат моделирования может быть проконтролирован на осциллографе Scope.

Схема моделирования в Matlab/Simulink булевой функции для рассмотренного выше примера, и полученной логической схемы (рисунок 2.2), представлена на рисунке 2.3. На схеме видно, что моделирование производится при следующих входных данных: $x_1=1$, $x_2=0$, $x_3=1$, $x_4=1$. Значение логической функции $f_{\text{мин.ДНФ}}$ при этом будет равно единице (рисунок 2.4).

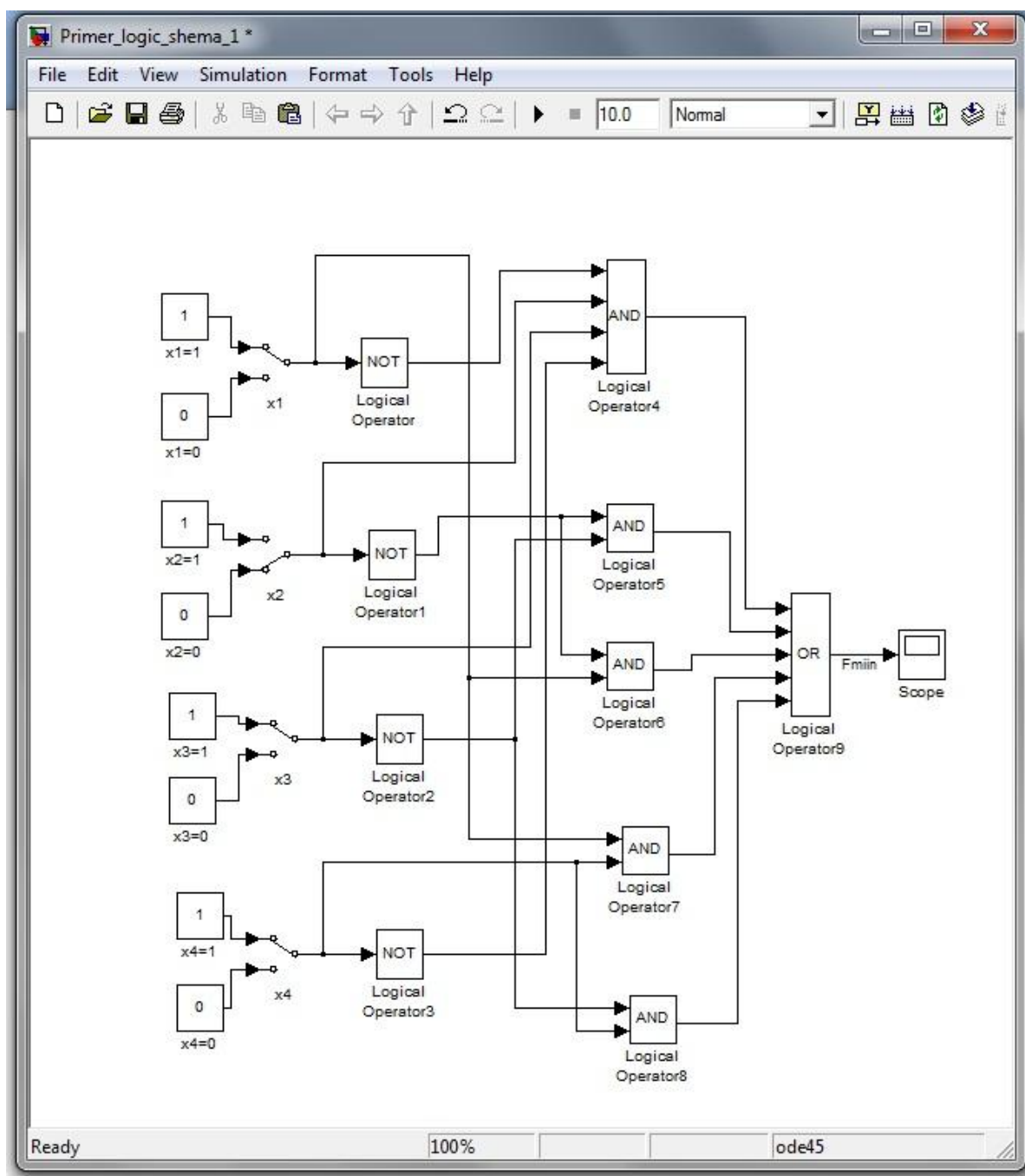


Рисунок 2.3 – Схема моделирования логической функции

№ вар.	Логическая функция в классе СДНФ
	$\vee x_1 x_2 \bar{x}_3 x_4 \vee x_1 x_2 x_3 x_4$
14	$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4$
15	$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 x_3 x_4 \vee x_1 \bar{x}_2 x_3 x_4$
16	$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4$
17	$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4$
18	$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4$
19	$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 x_3 x_4 \vee x_1 \bar{x}_2 x_3 x_4$
20	$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 x_3 x_4 \vee x_1 \bar{x}_2 x_3 x_4$
21	$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4$
22	$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4$
23	$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 x_3 x_4 \vee x_1 \bar{x}_2 x_3 x_4$
24	$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4$
25	$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4$

2.5. Содержание отчета о выполнении лабораторной работы

Отчет должен содержать:

- Цель работы.
- Предмет исследования.
- Получение сокращенной ДНФ булевой функции.
- Получение минимальной ДНФ булевой функции.
- Логическую схему исходной булевой функции.
- Логическую схему минимальной булевой функции.
- Схему моделирования исходной булевой функции в Matlab и результаты моделирования.
- Схему моделирования минимальной булевой функции в Matlab и результаты моделирования.
- Развернутые комментарии к полученным результатам.
- Краткие выводы.

2.6. Контрольные вопросы

1. Какая переменная называется булевой переменной?
2. Какая функция называется булевой?
3. Опишите основные законы булевой алгебры.
4. Какие существуют способы задания логических функций?
5. Дайте определение конститутенты единицы.
6. Дайте определение конститутенты нуля.
7. Какая форма представления логических функций называется дизъюнктивной нормальной формой?
8. Какая форма представления логических функций называется конъюнктивной нормальной формой?
9. Что понимают под совершенной дизъюнктивной нормальной формой записи логических функций?
10. Что понимают под минимизацией булевых функций?
11. Дайте определение сокращенной ДНФ.
12. Дайте определение минимальной ДНФ.
13. Опишите метод Квайна – Мак-Класки минимизации булевых функций.
14. Для чего нужна импликантная матрица и как ее построить?
15. Для чего нужны карты Карно и как их построить?

3. ЛАБОРАТОРНАЯ РАБОТА № 3 АБСТРАКТНЫЙ АНАЛИЗ КОНЕЧНОГО АВТОМАТА

3.1. Цель работы

Освоить основные понятия теории автоматов и основные методы анализа и оптимизации конечных автоматов на абстрактном уровне.

3.2. Теоретические сведения

Автоматом называют дискретный преобразователь информации, способный принимать различные состояния, переходить под воздействием входных сигналов из одного состояния в другое и выдавать выходные сигналы.

Если множество состояний автомата, а также множества входных и выходных сигналов конечны, то автомат называют конечным автоматом. Все реальные автоматы конечны.

Для того чтобы задать конечный автомат, нужно указать:

- множество возможных входных сигналов $X = \{x_1, x_2, \dots, x_m\}$;
- множество возможных выходных сигналов $Y = \{y_1, y_2, \dots, y_k\}$;
- множество возможных внутренних состояний автомата $A = \{a_1, a_2, \dots, a_n\}$;
- функцию переходов, определяющую состояние автомата $a(t+1)$ в момент дискретного времени $(t+1)$ в зависимости от состояния автомата $a(t)$ и значения входного сигнала $X(t)$ в момент времени t ;
- функцию выходов, определяющую зависимость выходного сигнала автомата $y(t)$ от состояния автомата и значения входного сигнала в момент времени t . Кроме того на

множестве состояний автомата фиксируют одно из внутренних состояний в качестве начального состояния.

Существует два типа конечных автоматов [2, 3, 7]. Для автоматов первого типа функции переходов и выходов имеют вид:

$$a(t+1)=f[a(t), x(t)], \quad y(t)=\varphi[a(t), x(t)].$$

У автоматов этого типа выходной сигнал в данном такте определяется внутренним состоянием и входным сигналом в данном такте. Такие автоматы получили название *автоматы Мили*.

Автоматы второго типа отличаются тем, что выходной сигнал такого автомата в данном такте определяется только внутренним состоянием автомата в этом же такте. Функции переходов и выходов для автомата второго типа, заданные на множестве входных сигналов $X=\{x_1, x_2, \dots, x_m\}$, множестве внутренних состояний $B=\{b_1, \dots, b_n\}$ и множестве выходных сигналов $Y=\{y_1, \dots, y_k\}$ имеют вид:

$$b(t+1)=f[b(t), x(t)], \quad y(t)=\varphi[b(t)].$$

Такие автоматы называются *автоматами Мура*.

Функции переходов и выходов можно задать таблицами переходов и выходов, либо графом автомата. Таблицы переходов и выходов автомата Мили приведены в таблицах 3.1 и 3.2.

Таблица 3.1 – Таблица переходов

Состояние Входной сигнал	a_0	a_1	a_2	a_3
x_1	a_1	a_2	a_3	a_3
x_2	a_0	a_0	a_0	a_0

Таблица 3.2 – Таблица выходов

Состояние Входной сигнал	a_0	a_1	a_2	a_3
x_1	y_2	y_2	y_1	y_2
x_2	y_2	y_2	y_2	y_3

В клетку таблицы переходов, находящуюся на пересечении столбца с буквой a_i и строки с буквой x_j , записывается состояние автомата $f(a_i, x_j)$, в которое он

переходит из состояния a_i при подаче на вход сигнала x_j . Аналогично записывается в таблице выходов сигнал y , который формируется автоматом при таком переходе.

При построении графа автомата вершины графа отождествляются с внутренними состояниями автомата. Каждая ветвь отмечается входным сигналом, вызывающим в автомате соответствующий данной ветви переход, и выходным сигналом, который возникает при этом переходе. На рисунке 3.1 приведен граф автомата, соответствующий автомату Мили, заданному таблицами 3.1 и 3.2.

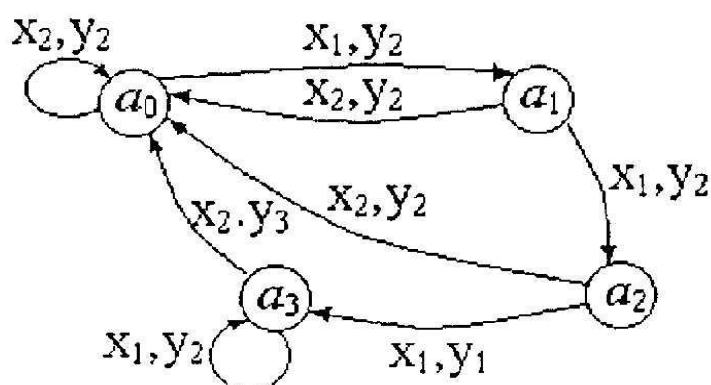


Рисунок 3.1 – Граф автомата Мили

Для автомата Мура функции переходов и выходов можно задать одной таблицей, которая называется отмеченной таблицей переходов. Отмеченная таблица переходов автомата Мура строится так же, как и таблица переходов автомата Мили, но над символами каждого внутреннего состояния записываются выходные сигналы, которые автомат выдает в данном состоянии. На графе автомата Мура значения выходных сигналов записываются около узлов. В качестве примера приведем отмеченную таблицу переходов (таблица 3.3) и граф автомата Мура (рисунок 3.2).

Таблица 3.3 - Отмеченная таблица переходов автомата Мура

Выходной сигнал		y_2	y_2	y_2	y_1	y_2	y_3
Состояние		b_0	b_1	b_2	b_3	b_4	b_5
Входной сигнал	x_1	b_1	b_2	b_3	b_4	b_4	b_1
	x_2	b_0	b_0	b_0	b_5	b_5	b_0

Работу автомата при произвольной последовательности входных сигналов можно проследить с помощью ленты автомата. Лента автомата представляет собой таблицу, в которой указываются такты работы автомата, значения входного сигнала в каждом такте и значения внутреннего состояния и выходного сигнала в каждом такте.

Пример ленты автомата, заданного таблицей 3.1 переходов и таблицей 3.2 выходов, приведен в таблице 3.4.

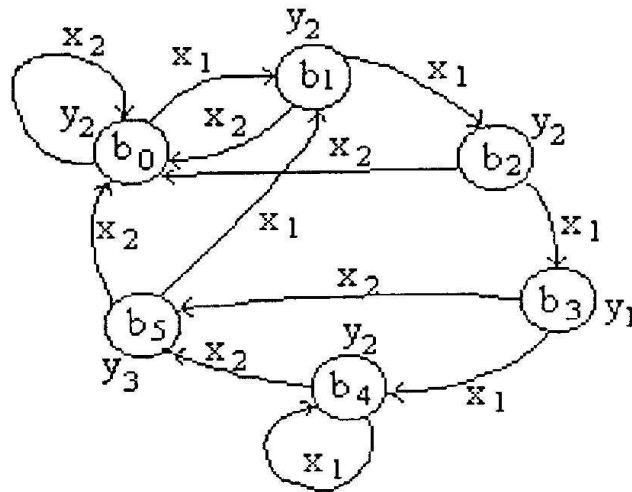


Рисунок 3.2 - Граф автомата Мура

Таблица 3.4 – Лента автомата Мили

Такт	1	2	3	4	5	6	7	8	9	10	11	12
Вход. сигнал	x_1	x_2	x_1	x_1	x_1	x_2	x_2	x_2	x_1	x_2	x_2	x_2
Состояние	a_0	a_1	a_0	a_1	a_2	a_3	a_0	a_0	a_0	a_1	a_0	a_0
Выход. сигнал	y_2	y_2	y_2	y_2	y_1	y_3	y_2	y_2	y_2	y_2	y_2	y_2

Для любого автомата Мили существует эквивалентный ему автомат Мура и наоборот.

Пусть конечный автомат Мили имеет функцию переходов $f_a(a, x)$ и функцию выходов $f_b(a, x)$. Найдем функцию переходов $f_b(b, x)$ и выходов $\varphi_b(b)$ автомата Мура, эквивалентного заданному автомату Мили. Поставим в соответствие каждой паре, включающей состояние a_i и входной сигнал x_j автомата Мили, состояние b_{ij} автомата Мура. Кроме того, во множестве состояний автомата Мура включим начальное состояние a_0 автомата Мили, обозначив его через b_0 . Для автомата Мили, заданного таблицами переходов и выходов (таблицы 3.1 и 3.2), такое соответствие можно представить в виде таблицы кодирования (таблица 3.5). Таблицу переходов эквивалентного автомата Мура строим в такой последовательности.

Таблица 3.5 - Таблица кодирования

Входной сигнал	Состояние			
	a_0 b_0	a_1	a_2	a_3
x_1	a_1 b_{01}	a_2 b_{11}	a_3 b_{21}	a_3 b_{31}
x_2	a_0 b_{02}	a_0 b_{12}	a_0 b_{22}	a_0 b_{32}

1. Выписать из таблицы кодирования состояния автомата Мили и соответствующие каждому состоянию множества состояний автомата Мура: $a_0=\{b_0, b_{02}, b_{12}, b_{22}, b_{32}\}$, $a_1=b_{01}$, $a_2=b_{11}$, $a_3=\{b_{21}, b_{31}\}$.

2. Если состояние b_{ij} входит в множество, соответствующее состоянию a_p , то в колонку таблицы переходов автомата Мура для состояния b_{ij} следует записать состояния, находящиеся в колонке для a_p (таблицы кодирования). Например, колонки таблицы переходов автомата Мура с состояниями $b_0, b_{02}, b_{12}, b_{22}, b_{32}$ совпадают с колонкой для a_0 таблицы кодирования, колонка b_{01} – с колонкой для a_1 и т.д.

Чтобы отметить выходными сигналами состояния, достаточно наложить таблицу выходов автомата Мили на таблицу кодирования состояний автомата Мура, и каждое состояние отметить тем выходным сигналом, с которым оно совпадает. Таблица переходов автомата Мура приведена таблицей 3.6.

Таблица 3.6 – Таблица переходов автомата Мура

Входной сигнал	y_2	y_2	y_2	y_2	y_2	y_1	y_2	y_2	y_3
	b_0	b_{01}	b_{02}	b_{11}	b_{12}	b_{21}	b_{22}	b_{31}	b_{32}
x_1	b_{01}	b_{11}	b_{01}	b_{21}	b_{01}	b_{31}	b_{01}	b_{31}	b_{01}
x_2	b_{02}	b_{12}	b_{02}	b_{22}	b_{02}	b_{32}	b_{02}	b_{32}	b_{02}

Обратная задача перехода от автомата Мура к эквивалентному автомату Мили решается следующим образом. Таблицы переходов автомата Мили и эквивалентного ему автомата Мура совпадают. Таблица выходов автомата Мили составляется так, что в каждую клетку таблицы записывается выходной сигнал, которым отмечено состояние, расположенное в данной клетке. При этом граф автомата Мили отличается от графа автомата Мура только тем, что выходные сигналы из каждого узла переносятся на все ветви, которые входят в данный узел.

Автоматы могут иметь "лишние" внутренние состояния. Число состояний автомата может быть минимизировано [2, 7]. Минимизация количества его состояний означает определение эквивалентного ему конечного автомата с минимальным количеством состояний.

Пусть в числе возможных состояний конечного автомата есть несколько состояний, обладающих таким свойством: произвольному входному слову соответствует некоторое выходное слово, не зависящее от того, в каком из указанных состояний находился конечный автомат в исходный момент. В этом случае можно считать, что все указанные состояния по существу одинаковы и лишь в разных случаях обозначены по-разному. Такие состояния можно объединить, обозначив одним символом.

Минимизация производится в несколько этапов. Первый этап минимизации количества состояний заключается в том, что просматриваются таблицы выходов и переходов и в верхней строке обеих таблиц (строка исходных состояний) выявляются такие состояния, которым соответствуют одинаковые столбцы, как в таблице выходов, так и в таблице переходов. Эти состояния объединяются, т.е. в обеих таблицах вычеркиваются все одинаковые столбцы, как в таблице выходов, так и в таблице переходов, кроме одного, соответствующего какому либо из выявленных состояний, а в оставшейся части таблицы состояний всем объединяемым состояниям присваивается один и тот же символ. В результате такого преобразования могут снова появиться состояния, которым в обеих таблицах соответствуют одинаковые столбцы.

Если столбцы в таблицах переходов и выходов, соответствующие различным исходным состояниям, содержат прочерки и только поэтому не совпадают, то соответствующие таким столбцам исходные состояния тоже можно объединить.

На втором этапе объединения все состояния объединяются в группы таким образом, чтобы состояниям, входящим в каждую группу, соответствовали одинаковые столбцы в таблице выходов. Каждой такой группе присваивается некоторое обозначение. Запись производится следующим образом:

символ группы	z_1	z_2	z_3
	$\underbrace{\hspace{1.5cm}}$	$\underbrace{\hspace{1.5cm}}$	$\underbrace{\hspace{1.5cm}}$
символ состояния	$a_i, a_j, \dots$	$a_k, a_l, \dots$	$a_m, a_n, \dots$

Состояния, входящие в разные группы, наверняка нельзя объединить. Возможность объединения состояний, входящих в одну и ту же группу, нужно исследовать дополнительно. Для этого под каждым состоянием выписывается соответствующий ему столбец из таблицы состояний, однако каждое состояние в этом столбце обозначается не собственным символом, а символом группы, в которую оно входит.

В результате такого переобозначения некоторые не совпадающие друг с другом столбцы таблицы состояний могут преобразоваться в совпадающие друг с другом столбцы групповых символов. Состояния, которым соответствуют такие совпадающие друг с другом столбцы, можно объединить. Далее проводим новую разбивку всех состояний на группы. Каждую старую группу нужно попытаться разбить на новые более мелкие группы. Все новые группы заново обозначаются

какими-либо символами. С каждым состоянием, входящим в новые группы, выполняются такие же операции, какие выполнялись ранее с состояниями, входившими в старые группы: выписываются из таблицы переходов столбцы, соответствующие этим состояниям, и каждое состояние в этих столбцах обозначается новым групповым символом.

Рассмотрим пример. Пусть автомат Мили задан таблицами переходов и выходов, приведенными в таблицах 3.7 и 3.8. Проведем минимизацию числа состояний этого автомата.

Таблица 3.7 – Таблица переходов

Состояние Входной сигнал	a_0	a_1	a_2	a_3	a_4	a_5	a_6
x_1	a_0	a_1	a_1	a_2	a_3	a_0	a_2
x_2	a_1	a_2	a_6	a_3	a_5	a_1	a_3

Таблица 3.8 – Таблица выходов

Состояние Входной сигнал	a_0	a_1	a_2	a_3	a_4	a_5	a_6
x_1	y_0	y_1	y_1	y_0	y_1	y_1	y_0
x_2	y_0	y_0	y_1	y_0	y_0	y_0	y_0

1 этап. Находим состояния, для которых совпадают столбцы в таблице выходов и в таблице переходов. Такими являются состояния a_3 и a_6 . Заменяем $a_3=a_6$ (состояние a_6 исключаем) и получаем таблицы 3.9 и 3.10.

Таблица 3.9 – Таблица переходов после первого этапа

Состояние Входной сигнал	a_0	a_1	a_2	a_3	a_4	a_5
x_1	a_0	a_1	a_1	a_2	a_3	a_0
x_2	a_1	a_2	a_6	a_3	a_5	a_1

Таблица 3.10 – Таблица выходов после первого этапа

Состояние Входной сигнал	a_0	a_1	a_2	a_3	a_4	a_5
x_1	y_0	y_1	y_1	y_0	y_1	y_1
x_2	y_0	y_0	y_1	y_0	y_0	y_0

2 этап. Группируем состояния в группы таким образом, чтобы в каждой группе были состояния, у которых совпадают столбцы в таблице выходов:

a_1, a_4, a_5 – группа 1 (z_1);

a_0, a_3 – группа 2 (z_2).

Могут объединяться лишь такие состояния, которые принадлежат одной и той же группе. Для выяснения возможности объединения таких состояний выписываем состояния одной группы, а под ним в столбик состояния из таблицы переходов, заменив обозначение состояния обозначением группы, в которую оно входит. Состояния с одинаковыми столбцами можно объединять. Покажем это:

$$\frac{a_1}{z_1} \quad \frac{a_4}{z_2} \quad \frac{a_5}{z_2}, \quad a_4=a_5.$$

$$a_2 \quad z_1 \quad z_1$$

$$\frac{a_0}{z_2} \quad \frac{a_3}{a_2}, \quad a_0 \neq a_3.$$

$$z_1 \quad z_2$$

Таблицы переходов и выходов автомата после второго этапа минимизации представлены таблицами 3.11 и 3.12.

Таблица 3.11 – Таблица переходов после второго этапа

Состояние Входной сигнал	a_0	a_1	a_2	a_3	a_4
x_1	a_0	a_1	a_1	a_2	a_3
x_2	a_1	a_2	a_6	a_3	a_4

Таблица 3.12 – Таблица выходов после второго этапа

Состояние Входной сигнал	a_0	a_1	a_2	a_3	a_4
x_1	y_0	y_1	y_1	y_0	y_1
x_2	y_0	y_0	y_1	y_0	y_0

3 этап. Группируем состояния, у которых одинаковые столбцы в таблице выходов:

$$(a_0, a_3)=z_1;$$

$$(a_1, a_4)=z_2.$$

Выясним возможность объединения состояний. Запишем:

$$\frac{a_0}{z_1} \quad \frac{a_3}{a_2}, \quad \frac{a_1}{z_2} \quad \frac{a_4}{z_1}$$

$$z_2 \quad z_1 \quad a_2 \quad z_2$$

Больше ничего не объединяется, полученные таблицы соответствуют автомату с минимальным числом состояний.

3.3. Порядок теоретической подготовки

Объектом исследований, выполняемых в лабораторной работе, является цифровой автомат Мили, заданный в виде функциональных таблиц.

Программа теоретической подготовки:

- Изучить основные понятия и определения цифровых автоматов.
- Изучить формы представления цифровых автоматов.
- Изучить метод построения автомата Мура по автомату Мили и автомата Мили по автомату Мура.
- Изучить показатели и условия, при которых необходима оптимизация конечного автомата.
- Изучить методику минимизации числа состояний конечного автомата.

3.4. Порядок выполнения экспериментальных исследований

Выполнение расчетных операций при анализе конечного автомата производится с использованием ПЭВМ и предлагаемых преподавателем пакетов прикладных программ (Matlab, Mathcad и др.).

По заданному таблицами переходов и выходов автомату Мили в соответствии с вариантом исходных данных (таблицы 3.13, 3.14), указанным преподавателем, необходимо:

- а) Построить граф автомата Мили.
 б) Построить ленту автомата Мили при условии произвольной последовательности входных сигналов и количестве тактов не менее 15. Начальным состоянием автомата считать состояние a_0 .

Проверить результат построения с помощью системы Matlab [5]. Для этого из основного окна Matlab командой File/New/M-File открыть новый М-файл. После чего в открывшемся окне М-файла сформировать (ввести) программу построения ленты автомата. Пример программы для рассмотренного выше примера автомата Мили (таблицы 3.1, 3.2) приведен в приложении А. По тексту программы понятно, какие параметры необходимо изменить для реализации индивидуального задания.

При необходимости М-файл сохранить. Нажатие на кнопку с изображением зеленого треугольника в окне М-файла произведет запуск процесса вычислений, после чего результат может быть проконтролирован в основном окне Matlab. Для рассмотренного примера получен результат:

```
Vhodn_signal =  
x1 x2 x1 x1 x1 x2 x2 x2 x1 x2 x2 x2
```

```
Tablica_perehodov =  
a1 a2 a3 a3  
a0 a0 a0 a0
```

```
Tablica_vihodov =  
y2 y2 y1 y2  
y2 y2 y2 y3
```

```
sostoyanie =  
a0
```

```
Lenta_avt =  
x1 x2 x1 x1 x1 x2 x2 x2 x1 x2 x2 x2  
a0 a1 a0 a1 a2 a3 a0 a0 a0 a1 a0 a0  
y2 y2 y2 y2 y1 y3 y2 y2 y2 y2 y2 y2
```

```
>>
```

- в) По исходному автомату Мили построить таблицу переходов и граф эквивалентного ему автомата Мура.
 г) Оптимизировать число состояний исходного автомата Мили и записать его результирующие таблицы переходов и выходов.

Таблица 3.13 – Варианты таблицы переходов автомата Мили

№ вар.	Входной сигнал	Состояние				
		a_0	a_1	a_2	a_3	a_4
1	x_1	a_0	a_2	a_3	a_2	a_2
	x_2	a_1	a_3	a_2	a_3	a_3
2	x_1	a_1	a_2	a_3	a_3	a_2

№ вар.	Входной сигнал	Состояние				
		a_0	a_1	a_2	a_3	a_4
	x_2	a_3	a_3	a_2	a_1	a_3
3	x_1	a_3	a_3	a_2	a_1	a_3
	x_2	a_0	a_2	a_1	a_2	a_2
4	x_1	a_2	a_1	a_2	a_3	a_1
	x_2	a_3	a_0	a_1	a_2	a_0
5	x_1	a_1	a_3	a_1	a_2	a_3
	x_2	a_2	a_0	a_3	a_1	a_0
6	x_1	a_1	a_0	a_2	a_0	a_0
	x_2	a_3	a_3	a_1	a_2	a_3
7	x_1	a_2	a_1	a_3	a_3	a_1
	x_2	a_0	a_2	a_0	a_2	a_2
8	x_1	a_3	a_0	a_2	a_1	a_0
	x_2	a_0	a_3	a_1	a_2	a_3
9	x_1	a_0	a_1	a_2	a_3	a_1
	x_2	a_3	a_2	a_3	a_1	a_2
10	x_1	a_2	a_1	a_2	a_3	a_1
	x_2	a_0	a_3	a_2	a_2	a_3
11	x_1	a_2	a_2	a_1	a_2	a_0
	x_2	a_3	a_3	a_2	a_3	a_1
12	x_1	a_2	a_3	a_3	a_2	a_1
	x_2	a_0	a_1	a_2	a_3	a_0
13	x_1	a_3	a_1	a_2	a_3	a_3
	x_2	a_2	a_2	a_1	a_2	a_0
14	x_1	a_1	a_3	a_0	a_1	a_2
	x_2	a_0	a_2	a_2	a_0	a_3
15	x_1	a_3	a_2	a_3	a_3	a_1
	x_2	a_0	a_1	a_2	a_0	a_2
16	x_1	a_0	a_0	a_1	a_0	a_1
	x_2	a_3	a_2	a_2	a_3	a_3
17	x_1	a_1	a_3	a_0	a_1	a_2
	x_2	a_2	a_2	a_3	a_2	a_0
18	x_1	a_0	a_1	a_1	a_0	a_3
	x_2	a_3	a_2	a_2	a_3	a_0
19	x_1	a_1	a_3	a_1	a_1	a_0
	x_2	a_2	a_1	a_3	a_2	a_3
20	x_1	a_1	a_3	a_2	a_1	a_2
	x_2	a_3	a_2	a_3	a_3	a_0

№ вар.	Входной сигнал	Состояние				
		a_0	a_1	a_2	a_3	a_4
21	x_1	a_1	a_0	a_1	a_1	a_2
	x_2	a_3	a_3	a_2	a_3	a_0
22	x_1	a_0	a_1	a_0	a_2	a_2
	x_2	a_1	a_2	a_3	a_3	a_3
23	x_1	a_1	a_0	a_1	a_3	a_2
	x_2	a_3	a_3	a_2	a_1	a_3
24	x_1	a_3	a_1	a_1	a_1	a_3
	x_2	a_0	a_2	a_3	a_2	a_2
25	x_1	a_2	a_0	a_1	a_0	a_1
	x_2	a_3	a_3	a_3	a_2	a_0

Таблица 3.14 – Варианты таблицы выходов автомата Мили

№ вар.	Входной сигнал	Состояние				
		a_0	a_1	a_2	a_3	a_4
1	x_1	y_3	y_2	y_3	y_1	y_2
	x_2	y_3	y_3	y_3	y_2	y_3
2	x_1	y_1	y_2	y_1	y_2	y_2
	x_2	y_1	y_2	y_3	y_1	y_3
3	x_1	y_1	y_2	y_3	y_3	y_2
	x_2	y_2	y_1	y_2	y_2	y_1
4	x_1	y_1	y_1	y_2	y_3	y_1
	x_2	y_3	y_1	y_2	y_2	y_1
5	x_1	y_3	y_3	y_3	y_1	y_3
	x_2	y_2	y_1	y_2	y_2	y_1
6	x_1	y_1	y_3	y_2	y_3	y_3
	x_2	y_1	y_2	y_1	y_2	y_2
7	x_1	y_1	y_2	y_2	y_2	y_2
	x_2	y_2	y_1	y_2	y_3	y_1
8	x_1	y_1	y_1	y_3	y_3	y_1
	x_2	y_3	y_2	y_3	y_2	y_2
9	x_1	y_1	y_3	y_2	y_3	y_3
	x_2	y_3	y_1	y_1	y_2	y_1
10	x_1	y_1	y_3	y_1	y_3	y_3
	x_2	y_3	y_1	y_3	y_2	y_1
11	x_1	y_2	y_1	y_3	y_2	y_3

№ вар.	Входной сигнал	Состояние				
		a_0	a_1	a_2	a_3	a_4
	x_2	y_3	y_2	y_3	y_3	y_3
12	x_1	y_2	y_2	y_1	y_2	y_1
	x_2	y_2	y_1	y_3	y_2	y_3
13	x_1	y_2	y_3	y_3	y_2	y_1
	x_2	y_1	y_2	y_2	y_1	y_2
14	x_1	y_1	y_3	y_2	y_1	y_1
	x_2	y_1	y_2	y_2	y_1	y_3
15	x_1	y_3	y_1	y_3	y_3	y_3
	x_2	y_1	y_2	y_2	y_1	y_2
16	x_1	y_3	y_3	y_2	y_3	y_1
	x_2	y_2	y_2	y_1	y_2	y_1
17	x_1	y_2	y_2	y_2	y_2	y_1
	x_2	y_1	y_3	y_2	y_1	y_2
18	x_1	y_1	y_3	y_3	y_1	y_1
	x_2	y_2	y_2	y_3	y_2	y_3
19	x_1	y_3	y_3	y_2	y_3	y_1
	x_2	y_1	y_2	y_1	y_1	y_3
20	x_1	y_3	y_3	y_1	y_3	y_1
	x_2	y_1	y_2	y_3	y_1	y_3
21	x_1	y_1	y_1	y_2	y_3	y_1
	x_2	y_3	y_1	y_2	y_2	y_1
22	x_1	y_3	y_3	y_3	y_1	y_3
	x_2	y_2	y_1	y_2	y_2	y_1
23	x_1	y_1	y_3	y_2	y_3	y_3
	x_2	y_1	y_2	y_1	y_2	y_2
24	x_1	y_1	y_2	y_2	y_2	y_2
	x_2	y_2	y_1	y_2	y_3	y_1
25	x_1	y_1	y_1	y_3	y_3	y_1
	x_2	y_3	y_2	y_3	y_2	y_2

3.5. Содержание отчета о выполнении лабораторной работы

Отчет должен содержать:

- Цель работы.
- Предмет исследования.
- Граф заданного автомата.
- Ленту заданного автомата.
- Построение автомата Мура по заданному автомату Мили.

- е) Процесс оптимизации числа состояний заданного конечного автомата.
- ж) Развернутые комментарии к полученным результатам.
- з) Краткие выводы.

3.6. Контрольные вопросы

1. Что понимают под комбинационной схемой?
2. В чем отличие цифрового автомата от комбинационной схемы?
3. Какой автомат называется конечным?
4. Как формируется таблица переходов автомата?
5. Как формируется таблица выходов автомата?
6. Опишите автомат Мили.
7. Опишите автомат Мура.
8. Как построить граф автомата Мили?
9. Как построить граф автомата Мура?
10. Что такое лента автомата?
11. Как построить ленту автомата?
12. Как перейти от автомата Мили к автомату Мура?
13. Как перейти от автомата Мура к автомату Мили?
14. В чем особенность отмеченной таблицы переходов автомата Мура?
15. Какие внутренние состояния автомата называются "лишними"?
16. Для чего нужна оптимизация конечного автомата?
17. Как минимизировать число состояний конечного автомата?
18. Опишите первый этап минимизации числа состояний конечного автомата.
19. Опишите основной этап минимизации числа состояний конечного автомата.

4. ЛАБОРАТОРНАЯ РАБОТА № 4 СТРУКТУРНЫЙ СИНТЕЗ КОНЕЧНОГО АВТОМАТА

4.1. Цель работы

Освоить методику синтеза конечных автоматов на структурном уровне с помощью Т-триггеров.

4.2. Теоретические сведения

Задача синтеза конечных автоматов заключается в построении сложного автомата из более простых автоматов, называемых элементарными. Чаще всего применяют элементарные автоматы с двумя внутренними состояниями, которые соединяются между собой с помощью логических элементов.

Различают задачи абстрактного и структурного синтеза дискретных автоматов [2, 3, 7]. Задача абстрактного синтеза заключается в получении таблиц переходов и выходов автомата (либо графа автомата).

При решении задачи структурного синтеза таблицы переходов и выходов автомата считаются заданными. Кроме того, задается или выбирается набор элементарных автоматов и логических схем. В результате синтеза следует получить структурную схему автомата.

В инженерной практике используются различные методы структурного синтеза конечных автоматов, каждый из которых имеет свои достоинства и недостатки. Рассмотрим наиболее простой метод, получивший название канонического [2, 7]. Согласно этому методу структурный синтез автомата проводится в такой последовательности.

1. Кодирование входных, выходных сигналов и состояний автомата.
2. Выбор элементов памяти.
3. Запись уравнений функций выходов и возбуждения автомата.
4. Построение структурной схемы автомата.

Рассмотрим последовательно этапы структурного синтеза автомата на примере конечного автомата Мили, заданного таблицами переходов и выходов (таблицы 4.1 и 4.2).

Таблица 4.1 – Таблица переходов

Состояние	a_0	a_1	a_2	a_3
Входной сигнал				
x_1	a_1	a_2	a_3	a_3
x_2	a_0	a_0	a_0	a_0

Таблица 4.2 – Таблица выходов

Состояние	a_0	a_1	a_2	a_3
Входной сигнал				
x_1	y_2	y_2	y_1	y_2
x_2	y_2	y_2	y_2	y_3

Кодирование. Входной, выходной сигналы и состояния автомата кодируются двоичными векторами. Пусть n – число входных сигналов. Тогда число двоичных векторов $k_{вх}$, необходимых для кодирования входных сигналов, определяется как ближайшее большее или равное целое число от $\log_2 n$. Так, при $n=3$, $\log_2 3=1,58$, следовательно, $k_{вх}=2$. Аналогично определяется число $k_{вых}$ двоичных векторов для кодирования выходных сигналов и число $k_{сост.}$ – для кодирования состояний.

Далее составляют таблицы кодирования. В таблицах указываются все сигналы и соответствующие им двоичные векторы.

В качестве примера составим таблицы кодирования для автомата, заданного таблицами 4.1 и 4.2. Определим количество двоичных векторов для кодирования.

Входные сигналы: $n_{вх}=2$, $\log_2 2=1$, $k_{вх}=1$.

Выходные сигналы: $n_{вых}=3$, $\log_2 3=1,58$, $k_{вых}=2$.

Число состояний: $n_{сост.}=4$, $\log_2 4=2$, $k_{сост.}=2$.

Таблицы кодирования представлены таблицами 4.3, 4.4 и 4.5.

Таблица 4.3 – Таблица кодирования входных сигналов

Входной сигнал	Код, β
x_1	0
x_2	1

Таблица 4.4 – Таблица кодирования выходных сигналов

Выходной сигнал	Код	
	ω_1	ω_2
y_1	0	0
y_2	0	1
y_3	1	0

Таблица 4.5 – Таблица кодирования состояний

Состояние	Код	
	Q_1	Q_2
a_0	0	0
a_1	0	1
a_2	1	0
a_3	1	1

После того, как составлены таблицы кодирования, составляют структурные таблицы переходов и выходов (таблицы 4.6 и 4.7). Эти таблицы строятся на основании исходных таблиц переходов и выходов (таблицы 4.1 и 4.2). В каждой клетке таблицы переходов, расположенной на пересечении строки входного сигнала x_i и столбца состояния a_j , вписывают код состояния, в которое автомат переходит из состояния a_j под воздействием входного сигнала x_i , а в таблице выходов - код выходного сигнала, который при этом переходе появляется на выходе автомата.

Выбор элементов памяти. При каноническом методе структурного синтеза в качестве элементов памяти используют элементарные автоматы Мура, обладающие полной системой переходов и выходов. Такими автоматами являются триггеры D, T, R-S, J-K.

Обобщенная структурная схема автомата. Обобщенные структурные схемы автомата Мура и автомата Мили показаны на рисунке 4.1. Количество элементов памяти автомата должно быть равно числу компонент вектора его состояний. Каждому переходу автомата из одного состояния в другое должно соответствовать

Таблица 4.6 – Структурная таблица переходов

Входной сигнал, x	Код входного сигнала, β	Состояния автомата и коды состояний (Q_1, Q_2)							
		a_0		a_1		a_2		a_3	
		0	0	0	1	1	0	1	1
x_1	0	0	1	1	0	1	1	1	1
x_2	1	0	0	0	0	0	0	0	0

Таблица 4.7 – Структурная таблица выходов

Входной сигнал, x	Код входного сигнала, β	Состояния автомата и коды состояний (Q_1, Q_2)							
		a_0		a_1		a_2		a_3	
		0	0	0	1	1	0	1	1
x_1	0	0	1	0	1	0	0	0	1
x_2	1	0	1	0	1	0	1	1	0
		ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2

переключение одного из элементов памяти. Переключения осуществляются комбинационной схемой функции возбуждения, подключаемой к информационным входам элементарных автоматов. Для выработки выходных сигналов автомата служит своя комбинационная схема.

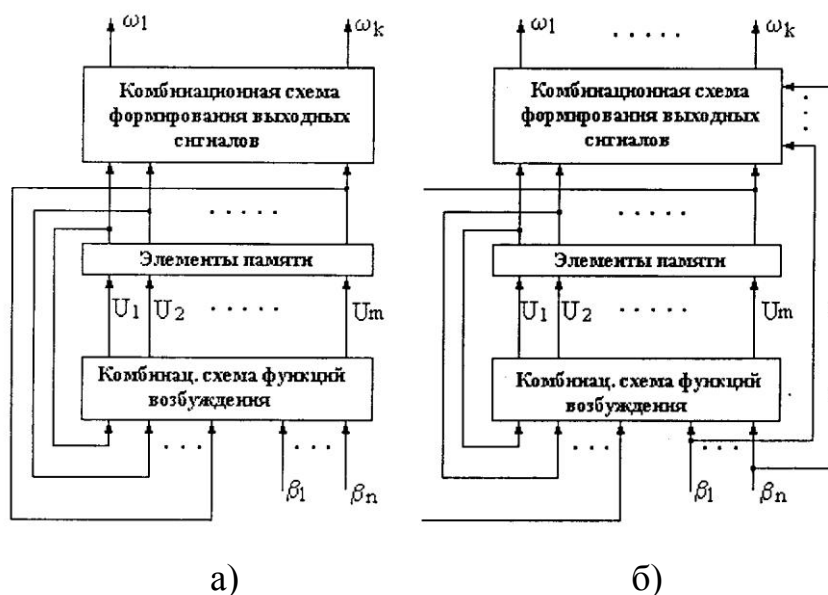


Рисунок 4.1 – Обобщенная структурная схема: а) автомата Мура; б) автомата Мили

Составление уравнений функций возбуждения и выходов автомата. Для составления уравнений функций возбуждения строят структурную таблицу функций возбуждения автомата. Эта таблица строится на основании структурной таблицы переходов и выходов и выбранных элементов памяти.

Предположим, что для рассматриваемого примера в качестве элементов памяти выбраны Т-триггеры. Таблицами 4.8 и 4.9 представлены элементарные автоматы (автоматы Мура) с одним входом, удовлетворяющие условиям детерминированности и полноты системы переходов. В этих таблицах приняты следующие обозначения: $q(t)$ – входной сигнал в момент времени t , $Q(t)$ – состояние и выходной сигнал, $Q(t+1)$ – состояние и выходной сигнал в момент времени $t+1$. Элементарный автомат, заданный таблицей 4.8 или 4.9, изменяет свое состояние только при подаче на вход сигнала, равного единице. Такой автомат называется триггером со счетным входом, или Т-триггером. Запишем функцию перехода для Т-триггера (таблица 4.8) в аналитической форме:

$$Q(t+1) = (\bar{q}Q + q\bar{Q})^t.$$

Таблица 4.8 – Элементарный автомат с одним входом первого типа

$q(t)$	$Q(t)$	$Q(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0

Таблица 4.9 – Элементарный автомат с одним входом второго типа

$q(t)$	$Q(t)$	$Q(t+1)$
0	0	1
0	1	0
1	0	0
1	1	1

Таблица функций возбуждения представлена таблицей 4.10. Таблица строится следующим образом. В верхней строке таблицы указаны коды состояний автомата, как и в таблицах 4.6 и 4.7.

Таблица 4.10 – Таблица функций возбуждения для Т-триггера

Код входного сигнала, β	Коды состояний (Q_1, Q_2)							
	0	0	0	1	1	0	1	1
0	0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	1	1
	u_1	u_2	u_1	u_2	u_1	u_2	u_1	u_2

В крайней левой колонке указаны коды входных сигналов. Рассмотрим клетку таблицы, стоящей на пересечении строки $\beta=0$ и столбца $(Q_1, Q_2)=(0, 0)$. Из таблицы переходов следует, что под воздействием входного сигнала $\beta=0$ автомат из состояния $(0, 0)$ переходит в состояние $(0, 1)$. Для выбранных элементов памяти (Т-

триггер) это означает, что первый триггер не изменил своего состояния, а второй триггер изменил свое состояние. Чтобы переключение триггеров соответствовало этому условию, сигнал на входе первого триггера должен быть равен нулю ($u_1=0$), а сигнал на входе второго триггера должен быть равен единице ($u_2=1$). Подобным же образом заполняются остальные клетки таблицы.

На основании таблицы функций возбуждения составляют таблицу истинности функций возбуждения (таблица 4.11), а затем определяют минимальные формы этих функций (таблицы 4.12 и 4.13).

Таблица 4.11 – Таблица истинности функций возбуждения и функций выхода

β	Q_1	Q_2	u_1	u_2	ω_1	ω_2
0	0	0	0	1	0	1
0	0	1	1	1	0	1
0	1	0	0	1	0	0
0	1	1	0	0	0	1
1	0	0	0	0	0	1
1	0	1	0	1	0	1
1	1	0	1	0	0	1
1	1	1	1	1	1	0

Таблица 4.12 – Карта Карно для функции u_1

	β		$\bar{\beta}$	
Q_1	1	1	0	0
$\bar{Q}_1$	0	0	1	0
	$\bar{Q}_2$	Q_2		$\bar{Q}_2$

Таблица 4.13 – Карта Карно для функции u_2

	β		$\bar{\beta}$	
Q_1	0	1	0	1
$\bar{Q}_1$	0	1	1	1
	$\bar{Q}_2$	Q_2		$\bar{Q}_2$

Для получения таблицы истинности функций выхода воспользуемся структурной таблицей выходов. Таблица истинности составляется следующим образом. В таблицах 4.6 и 4.7 для комбинации $\beta=0$, $Q_1=0$, $Q_2=0$ значения векторов выходного сигнала следующие: $\omega_1=0$, $\omega_2=1$. В таблице истинности (таблице 4.11) в строке $\beta=0$, $Q_1=0$, $Q_2=0$ вносим значения $\omega_1=0$, $\omega_2=1$ и т.д.

Функции возбуждения и функции выхода для рассматриваемого примера равны:

$$u_1 = \beta Q_1 + \bar{\beta} \bar{Q}_1 Q_2; u_2 = \beta Q_2 + \bar{\beta} (\bar{Q}_1 + \bar{Q}_2); \omega_1 = \beta Q_1 Q_2; \omega_2 = \bar{Q}_1 + \beta \bar{Q}_2 + \bar{\beta} Q_2.$$

Функциональная схема автомата на Т-триггерах показана на рисунке 4.2.

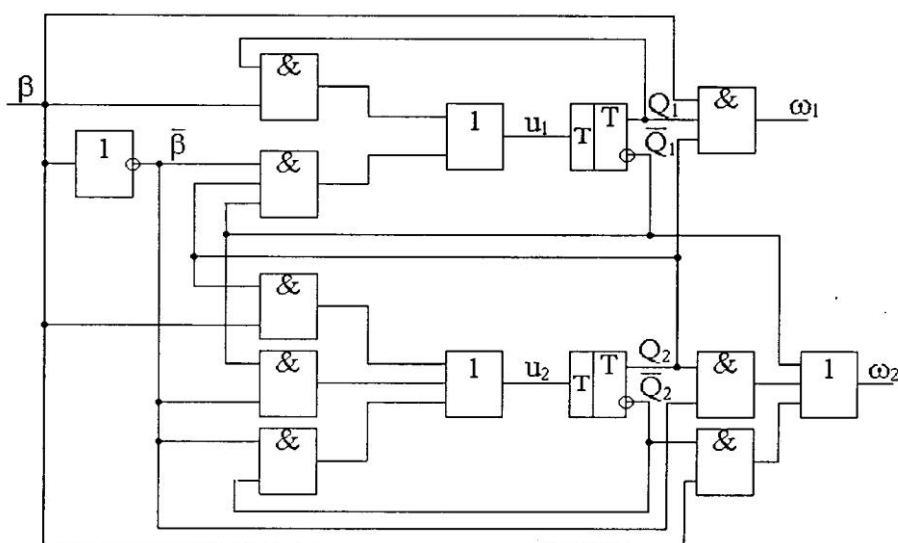


Рисунок 4.2 – Функциональная схема автомата, выполненного на Т-триггерах

4.3. Порядок теоретической подготовки

Объектом исследований, выполняемых в лабораторной работе, является цифровой автомат Мили, заданный в виде функциональных таблиц.

Программа теоретической подготовки:

- Изучить задачу синтеза цифровых автоматов.
- Изучить систему элементов, из которых должны строиться конечные автоматы на Т-триггерах.
- Изучить структурный синтез конечного автомата на Т-триггерах каноническим методом.

4.4. Порядок выполнения экспериментальных исследований

По заданному таблицами переходов и выходов автомату Мили в соответствии с вариантом исходных данных (таблицы 4.14, 4.15), указанным преподавателем, необходимо:

- Построить таблицы кодирования входных, выходных сигналов и состояний автомата.
- По исходным таблицам переходов и выходов автомата Мили составить структурные таблицы переходов и выходов.
- В качестве элементов памяти использовать автомат Мура на базе Т-триггера. Построить таблицу функций возбуждения для Т-триггеров.
- На основании таблицы функций возбуждения и структурной таблицы выходов составить таблицу истинности функций возбуждения и функций выхода, а затем с помощью карт Карно определить минимальные формы этих функций.
- По полученным минимальным формам (булевым функциям) составить функциональную схему синтезированного автомата на Т-триггерах.
- Выполнить моделирование синтезированного автомата в системе Matlab.

Моделирование производится непосредственно по составленной функциональной схеме с помощью подсистемы Simulink пакета Matlab [5]. Для этого необходимо из основного окна Matlab запустить Simulink и командой File/New/Model создать новый файл модели. После чего в окне модели сформировать схему модели перемещением с помощью компьютерной мыши необходимых блоков из соответствующих библиотек, которые соединяются между собой как того требует схема. Для формирования модели использовать следующие элементы библиотек блоков Simulink (Simulink Library Browser):

- в библиотеке Simulink Extras/Flip Flops – блоки триггеров. Там же находится блок Синхроимпульсов (Clock), который нужно использовать для задания периодических входных сигналов в схемах моделирования. Т-триггера нет в этой библиотеке. В качестве Т-триггера использовать универсальный JK-триггер (JK Flip Flop), на J, K входы которого подается сигнал логической единицы (блок Constant из библиотеки Simulink/Sources). В блоках триггеров символ «!» означает инверсию. Схема моделирования работы Т-триггера представлена на рисунке 4.3;

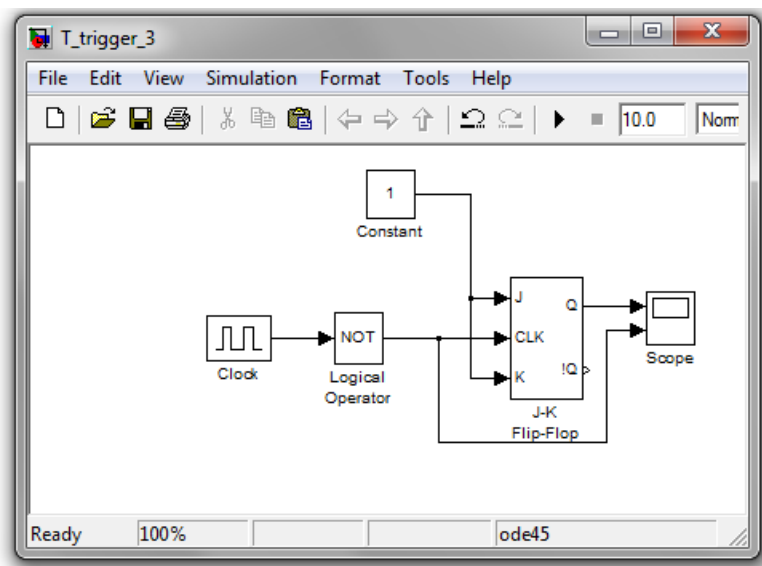


Рисунок 4.3 – Схема моделирования работы Т-триггера

- в библиотеке Simulink/Sources (источники сигнала) – блок источника постоянного сигнала Constant, которым задается логическая единица. В блоке Constant, в меню Signal Attributes, во вкладке Output data type выбрать Boolean. Параметр Constant value задается равным единице;
- в библиотеке Simulink/Sinks – осциллограф Scope для построения графиков моделируемых процессов;
- в библиотеке Simulink/Logic and Bit Operations – блок логических операций Logical Operator. Для того, чтобы выбрать необходимую логическую операцию необходимо переместить блок Logical Operator в окно модели и дважды щелкнуть по нему левой кнопкой мыши. После чего в меню Signal Attributes, во вкладке Output data type выбрать Boolean. В меню Main, во вкладке Operator выбрать требуемую логическую

операцию (AND, OR, NOT и т.д.), а во вкладке Number of input ports – необходимое число входов логического блока.

Примечание: В Matlab версии 6 и выше между выходом блоков, являющихся источниками сигналов (Clock, Constant), и входами блоков триггеров (FlipFlop) следует помещать буферный логический элемент, не изменяющий знака сигнала, например, одновходовый элемент И (ИЛИ).

При необходимости файл модели сохраняется.

Выбор опции меню Simulation/Start (Моделирование/Старт) или нажатие на кнопку с изображением черного треугольника в окне моделирования произведет запуск процесса моделирования, после чего результаты моделирования могут быть проконтролированы на осциллографе Scope.

Схема моделирования в Matlab/Simulink работы синтезированного конечного автомата для рассмотренного выше примера, и полученной функциональной схемы (рисунок 4.2), представлена на рисунке 4.4.

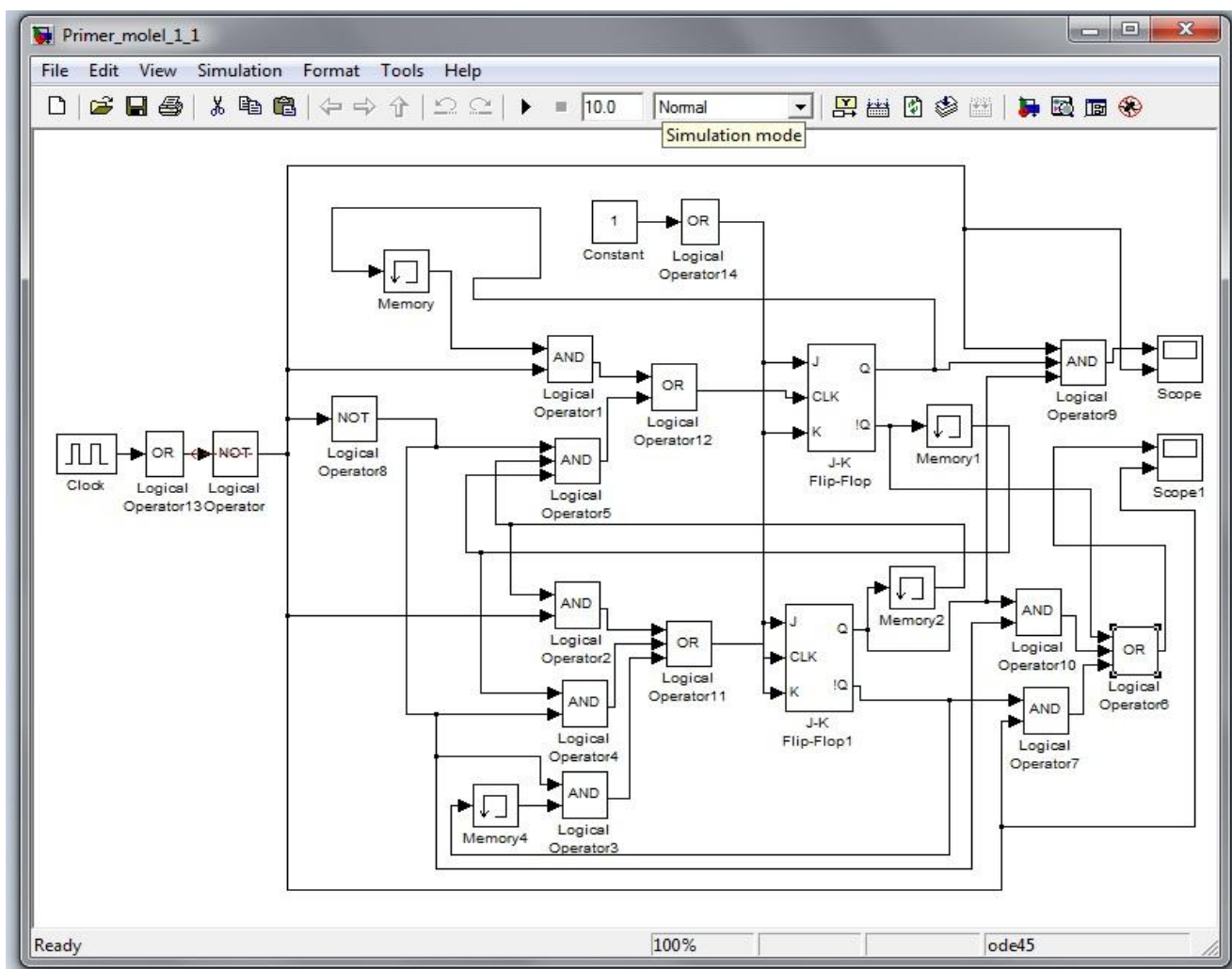


Рисунок 4.4 - Схема моделирования работы синтезированного конечного автомата

Таблица 4.14 – Варианты таблицы переходов автомата Мили

№ вар.	Входной сигнал	Состояние			
		a_0	a_1	a_2	a_3
1	x_1	a_1	a_2	a_3	a_2
	x_2	a_0	a_0	a_0	a_0
2	x_1	a_1	a_2	a_3	a_3
	x_2	a_0	a_0	a_2	a_1
3	x_1	a_1	a_2	a_3	a_1
	x_2	a_0	a_1	a_0	a_0
4	x_1	a_1	a_2	a_3	a_2
	x_2	a_3	a_0	a_0	a_0
5	x_1	a_1	a_2	a_3	a_0
	x_2	a_1	a_0	a_0	a_0
6	x_1	a_1	a_2	a_3	a_3
	x_2	a_3	a_0	a_1	a_0
7	x_1	a_1	a_2	a_3	a_0
	x_2	a_0	a_0	a_0	a_2
8	x_1	a_1	a_2	a_3	a_3
	x_2	a_0	a_3	a_0	a_0
9	x_1	a_1	a_2	a_3	a_2
	x_2	a_3	a_0	a_0	a_0
10	x_1	a_1	a_2	a_3	a_3
	x_2	a_0	a_0	a_2	a_0
11	x_1	a_2	a_3	a_2	a_1
	x_2	a_0	a_0	a_0	a_0
12	x_1	a_3	a_3	a_2	a_1
	x_2	a_1	a_2	a_0	a_0
13	x_1	a_1	a_3	a_2	a_1
	x_2	a_0	a_0	a_1	a_0
14	x_1	a_2	a_3	a_2	a_1
	x_2	a_0	a_0	a_0	a_3
15	x_1	a_0	a_3	a_2	a_1
	x_2	a_0	a_0	a_0	a_1
16	x_1	a_3	a_3	a_2	a_1
	x_2	a_0	a_1	a_0	a_3
17	x_1	a_0	a_3	a_2	a_1
	x_2	a_2	a_0	a_0	a_0
18	x_1	a_3	a_3	a_2	a_1
	x_2	a_0	a_0	a_3	a_0

№ вар.	Входной сигнал	Состояние			
		a_0	a_1	a_2	a_3
19	x_1	a_2	a_3	a_2	a_1
	x_2	a_0	a_0	a_0	a_3
20	x_1	a_3	a_3	a_2	a_1
	x_2	a_0	a_2	a_0	a_0
21	x_1	a_0	a_2	a_3	a_3
	x_2	a_2	a_0	a_1	a_1
22	x_1	a_0	a_2	a_3	a_3
	x_2	a_2	a_1	a_0	a_0
23	x_1	a_0	a_2	a_1	a_3
	x_2	a_2	a_3	a_0	a_0
24	x_1	a_1	a_2	a_3	a_3
	x_2	a_2	a_0	a_0	a_0
25	x_1	a_0	a_2	a_3	a_1
	x_2	a_2	a_0	a_2	a_2

Таблица 4.15 – Варианты таблицы выходов автомата Мили

№ вар.	Входной сигнал	Состояние			
		a_0	a_1	a_2	a_3
1	x_1	y_2	y_2	y_1	y_2
	x_2	y_3	y_2	y_2	y_2
2	x_1	y_1	y_1	y_2	y_1
	x_2	y_1	y_1	y_1	y_3
3	x_1	y_3	y_3	y_2	y_3
	x_2	y_3	y_3	y_3	y_1
4	x_1	y_1	y_2	y_2	y_2
	x_2	y_2	y_2	y_2	y_3
5	x_1	y_2	y_1	y_2	y_2
	x_2	y_2	y_2	y_2	y_3
6	x_1	y_1	y_3	y_2	y_3
	x_2	y_1	y_1	y_3	y_1
7	x_1	y_1	y_1	y_1	y_2
	x_2	y_2	y_2	y_2	y_3
8	x_1	y_3	y_3	y_1	y_3
	x_2	y_3	y_3	y_2	y_3
9	x_1	y_2	y_2	y_1	y_2
	x_2	y_2	y_2	y_3	y_2

№ вар.	Входной сигнал	Состояние			
		a_0	a_1	a_2	a_3
10	x_1	y_1	y_1	y_2	y_1
	x_2	y_1	y_1	y_1	y_3
11	x_1	y_2	y_1	y_2	y_2
	x_2	y_2	y_2	y_2	y_3
12	x_1	y_1	y_2	y_1	y_1
	x_2	y_3	y_1	y_1	y_1
13	x_1	y_3	y_2	y_3	y_3
	x_2	y_1	y_3	y_3	y_3
14	x_1	y_2	y_2	y_2	y_1
	x_2	y_3	y_2	y_2	y_2
15	x_1	y_2	y_2	y_1	y_2
	x_2	y_3	y_2	y_2	y_2
16	x_1	y_3	y_2	y_3	y_1
	x_2	y_1	y_3	y_1	y_1
17	x_1	y_2	y_1	y_1	y_1
	x_2	y_3	y_2	y_2	y_2
18	x_1	y_3	y_1	y_3	y_3
	x_2	y_3	y_2	y_3	y_3
19	x_1	y_2	y_1	y_2	y_2
	x_2	y_2	y_3	y_2	y_2
20	x_1	y_1	y_2	y_1	y_1
	x_2	y_3	y_1	y_1	y_1
21	x_1	y_3	y_1	y_3	y_2
	x_2	y_1	y_1	y_1	y_3
22	x_1	y_1	y_1	y_2	y_1
	x_2	y_2	y_2	y_3	y_2
23	x_1	y_3	y_3	y_3	y_1
	x_2	y_3	y_3	y_3	y_2
24	x_1	y_2	y_2	y_2	y_1
	x_2	y_2	y_2	y_2	y_3
25	x_1	y_1	y_1	y_1	y_2
	x_2	y_1	y_1	y_3	y_1

4.5. Содержание отчета о выполнении лабораторной работы

Отчет должен содержать:

- Цель работы.
- Предмет исследования.
- Таблицы кодирования входных, выходных сигналов и состояний автомата.

- г) Структурные таблицы переходов и выходов для синтезируемого автомата.
- д) Таблицу функций возбуждения для элементов памяти в виде Т-триггеров.
- е) Таблицу истинности функций возбуждения и функций выхода автомата.
- ж) Минимальные формы функций возбуждения, построенные по картам Карно.
- з) Минимальные формы функций выхода, построенные по картам Карно.
- и) Функциональную схему синтезированного автомата, выполненного на Т-триггерах.
- к) Схему моделирования синтезированного автомата в Matlab и результаты моделирования.
- л) Развернутые комментарии к полученным результатам.
- м) Краткие выводы.

4.6. Контрольные вопросы

1. Что понимают под синтезом конечного автомата на абстрактном уровне?
2. Что понимают под синтезом конечного автомата на структурном уровне?
3. Какие автоматы называются элементарными?
4. Какие существуют элементарные автоматы с одним входом?
5. Сколько существует типов элементарных автоматов с одним входом?
6. Как получить минимальные формы функций возбуждения автомата?
5. Опишите D-триггер.
7. Опишите R-S-триггер.
9. Опишите обобщенную структурную схему автомата Мили.
10. Как построить таблицы кодирования входных, выходных сигналов и состояний автомата при его структурном синтезе?
11. Как построить структурные таблицы переходов при синтезе конечного автомата?
12. Как построить таблицу функций возбуждения Т-триггера?
13. Как построить таблицу истинности функций возбуждения и функций выхода конечного автомата на базе Т-триггеров?
14. Как построить функциональную схему автомата, выполненного на Т-триггерах?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Баврин, И.И. Дискретная математика. Учебник и задачник : для прикладного бакалавриата / И. И. Баврин. – Москва : Издательство Юрайт, 2019. – 193 с. – Режим доступа: <https://www.biblio-online.ru/bcode/432994>.
2. Бобриков С.А. Математические основы теории систем: Консп. лекц. для спец. 7.091401/ С.А.Бобриков. – Одесса: ОНПУ, 2001. – 81 с. – Режим доступа: <http://lib.mexmat.ru/books/106219>.
3. Гузик, В.Ф. Теория цифровых автоматов: учебное пособие / В.Ф. Гузик, В.Н. Пуховский, Е.Р. Мунтян, О.А. Мунтян. – Ростов-на-Дону: Издательство ЮФУ, 2015. – 147 с. – Режим доступа: <http://znanium.com/catalog/product/989754>.
4. Куликов, В.В. Дискретная математика: учебное пособие / В.В. Куликов. – М.: РИОР, 2007. – 174 с. – Режим доступа: <http://znanium.com/catalog/product/126799>.
5. Лазарев Ю. Моделирование процессов и систем в МАТЛАБ. Учебный курс/Ю. Лазарев. – СПб: Издательство Питер, 2005. – 512 с.
6. Новоселов В.Г. Дискретная математика в задачах и примерах: учеб. пособие. Ч. 2 / В. Г. Новоселов, Т. В. Пластун, А. В. Скатков ; Севастоп. нац. техн. ун-т. - Севастополь : Изд-во Севастоп. нац. техн. ун-та, 2002. – 122 с.
7. Постников, А.И. Прикладная теория цифровых автоматов : учеб. пособие / А.И. Постников, О.В. Непомнящий, Л.В. Макуха. - Красноярск : Сиб. федер. ун-т, 2017. – 206 с. – Режим доступа: <http://znanium.com/catalog/product/1032125>.

ПРИЛОЖЕНИЕ А

(справочное)

Текст программы построения ленты автомата Мили для системы Matlab

```

1 - Vhodn_signal=['x1 ' 'x2 ' 'x1 ' 'x1 ' 'x1 ' 'x2 ' 'x2 ' 'x2 ' 'x1 ' 'x2 ' 'x2 ' 'x2 ' ]
2 - Tablica_perehodov=['a1 ' 'a2 ' 'a3 ' 'a3 ' ;'a0 ' 'a0 ' 'a0 ' 'a0 ' ]
3 - Tablica_vihodov=['y2 ' 'y2 ' 'y1 ' 'y2 ' ;'y2 ' 'y2 ' 'y2 ' 'y3 ' ]
4 - sostoyanie='a0 '
5 - bb='r';
6 - d=Vhodn_signal;
7 - p=length(d);
8 - for i = 1: 3: p-2
9
10 - if (Vhodn_signal(i)=='x' && Vhodn_signal(i+1)=='1' && Vhodn_signal(i+2)==' ')
11 -     Lenta_avt(1,i)='x';
12 -     Lenta_avt(1,i+1)='1';
13 -     Lenta_avt(1,i+2)=' ';
14 -     Lenta_avt(2,i)=sostoyanie(1);
15 -     Lenta_avt(2,i+1)=sostoyanie(2);
16 -     Lenta_avt(2,i+2)=sostoyanie(3);
17 -     if (sostoyanie(1)=='a' && sostoyanie(2)=='0' && sostoyanie(3)==' ') && bb=='r'
18 -         Lenta_avt(3,i)=Tablica_vihodov(1,1);
19 -         Lenta_avt(3,i+1)=Tablica_vihodov(1,2);
20 -         Lenta_avt(3,i+2)=Tablica_vihodov(1,3);
21 -         sostoyanie(1)=Tablica_perehodov(1,1);
22 -         sostoyanie(2)=Tablica_perehodov(1,2);
23 -         sostoyanie(3)=Tablica_perehodov(1,3);
24 -         bb='g'; end;
25 -     if (sostoyanie(1)=='a' && sostoyanie(2)=='1' && sostoyanie(3)==' ') && bb=='r'
26 -         Lenta_avt(3,i)=Tablica_vihodov(1,4);
27 -         Lenta_avt(3,i+1)=Tablica_vihodov(1,5);
28 -         Lenta_avt(3,i+2)=Tablica_vihodov(1,6);
29 -         sostoyanie(1)=Tablica_perehodov(1,4);
30 -         sostoyanie(2)=Tablica_perehodov(1,5);
31 -         sostoyanie(3)=Tablica_perehodov(1,6);
32 -         bb='g'; end;
33 -     if (sostoyanie(1)=='a' && sostoyanie(2)=='2' && sostoyanie(3)==' ') && bb=='r'
34 -         Lenta_avt(3,i)=Tablica_vihodov(1,7);
35 -         Lenta_avt(3,i+1)=Tablica_vihodov(1,8);
36 -         Lenta_avt(3,i+2)=Tablica_vihodov(1,9);
37 -         sostoyanie(1)=Tablica_perehodov(1,7);
38 -         sostoyanie(2)=Tablica_perehodov(1,8);
39 -         sostoyanie(3)=Tablica_perehodov(1,9);
40 -         bb='g'; end
41 -     if (sostoyanie(1)=='a' && sostoyanie(2)=='3' && sostoyanie(3)==' ') && bb=='r'
42 -         Lenta_avt(3,i)=Tablica_vihodov(1,10);
43 -         Lenta_avt(3,i+1)=Tablica_vihodov(1,11);
44 -         Lenta_avt(3,i+2)=Tablica_vihodov(1,12);
45 -         sostoyanie(1)=Tablica_perehodov(1,10);
46 -         sostoyanie(2)=Tablica_perehodov(1,11);
47 -         sostoyanie(3)=Tablica_perehodov(1,12);
48 -         bb='g'; end;

```

```

49 - end;
50 - if (Vhodn_signal(i)=='x' && Vhodn_signal(i+1)=='2' && Vhodn_signal(i+2)==' ')
51 -     Lenta_avt(1,i)=='x';
52 -     Lenta_avt(1,i+1)=='2';
53 -     Lenta_avt(1,i+2)==' ';
54 -     Lenta_avt(2,i)=sostoyanie(1);
55 -     Lenta_avt(2,i+1)=sostoyanie(2);
56 -     Lenta_avt(2,i+2)=sostoyanie(3);
57 -     if (sostoyanie(1)=='a' && sostoyanie(2)=='0' && sostoyanie(3)==' ') && bb=='r'
58 -         Lenta_avt(3,i)=Tablica_vihodov(2,1);
59 -         Lenta_avt(3,i+1)=Tablica_vihodov(2,2);
60 -         Lenta_avt(3,i+2)=Tablica_vihodov(2,3);
61 -         sostoyanie(1)=Tablica_perehodov(2,1);
62 -         sostoyanie(2)=Tablica_perehodov(2,2);
63 -         sostoyanie(3)=Tablica_perehodov(2,3);
64 -         bb='g'; end
65 -     if (sostoyanie(1)=='a' && sostoyanie(2)=='1' && sostoyanie(3)==' ') && bb=='r'
66 -         Lenta_avt(3,i)=Tablica_vihodov(2,4);
67 -         Lenta_avt(3,i+1)=Tablica_vihodov(2,5);
68 -         Lenta_avt(3,i+2)=Tablica_vihodov(2,6);
69 -         sostoyanie(1)=Tablica_perehodov(2,4);
70 -         sostoyanie(2)=Tablica_perehodov(2,5);
71 -         sostoyanie(3)=Tablica_perehodov(2,6);
72 -         bb='g'; end
73 -     if (sostoyanie(1)=='a' && sostoyanie(2)=='2' && sostoyanie(3)==' ') && bb=='r'
74 -         Lenta_avt(3,i)=Tablica_vihodov(2,7);
75 -         Lenta_avt(3,i+1)=Tablica_vihodov(2,8);
76 -         Lenta_avt(3,i+2)=Tablica_vihodov(2,9);
77 -         sostoyanie(1)=Tablica_perehodov(2,7);
78 -         sostoyanie(2)=Tablica_perehodov(2,8);
79 -         sostoyanie(3)=Tablica_perehodov(2,9);
80 -         bb='g'; end
81 -     if (sostoyanie(1)=='a' && sostoyanie(2)=='3' && sostoyanie(3)==' ') && bb=='r'
82 -         Lenta_avt(3,i)=Tablica_vihodov(2,10);
83 -         Lenta_avt(3,i+1)=Tablica_vihodov(2,11);
84 -         Lenta_avt(3,i+2)=Tablica_vihodov(2,12);
85 -         sostoyanie(1)=Tablica_perehodov(2,10);
86 -         sostoyanie(2)=Tablica_perehodov(2,11);
87 -         sostoyanie(3)=Tablica_perehodov(2,12);
88 -         bb='g'; end;
89 -     end;
90 -     bb='r';
91 - end
92 -
93 - Lenta_avt

```

**МЕТОДЫ АНАЛИЗА И СИНТЕЗА
ЦИФРОВЫХ ЭЛЕМЕНТОВ
СИСТЕМ АВТОМАТИКИ.
ЛАБОРАТОРНЫЙ ПРАКТИКУМ
ПО ДИСЦИПЛИНЕ**

*Методические указания
к выполнению лабораторных работ*

*Составители: Грушун Андрей Иванович,
Грушун Татьяна Александровна*

В авторской редакции

Изд. № 106/2022. Объем 3 п.л. Усл. печ. л. 2,79. Уч.-изд. л. 2,94.
РИИЦМ ФГАОУ ВО «Севастопольский государственный университет»