# Министерство специального и высшего образования

# Хабаровский государственный технический университет

# Кафедра «Автоматика и системотехника»

# Курсовой проект

# По предмету: Математические основы теории систем

|  |  |
| --- | --- |
| Выполнил студент гр. УИТС-21у: | Д.И. Хоменко |
| Проверил: | В.В. Воронин |

# г. Хабаровск

# 2003 г.ЗАДАНИЕ

Курсовая работа состоит из 3-х разделов, в каждом из которых рассматривается отдельный параграф дисциплины «Математические основы теории систем».

В первом разделе данной курсовой работы требуется, имея схему системы автоматического управления перейти к сигнальному графу, определить его структурные характеристики и проанализировать с помощью формулы Мезона.

Во втором разделе необходимо рассмотреть логические функции, способы их задания и синтез комбинационных схем.

В третьем разделе необходимо синтезировать автомат с памятью на основе содержательного описания алгоритма его работы.

# РЕФЕРАТ

Курсовая работа содержит пояснительную записку состоящую из трех разделов на 38 листах формата А4, включающую 6 рисунков, 2 схемы, 14 таблиц и 3 литературных источника.

Объектом исследования являются система автоматического управления и логическое устройство, в данном случае семисегментный элемент.

Цель работы состоит в том чтобы закрепить на практике теоретический материал курса лекций «Математические основы теории систем» и приобретение навыков по анализу систем и синтезу схем.

Ключевые слова: структурная схема, сигнальный граф, путь, конур, САУ, синтез схем, конечный автомат, логическая функция, таблица истинности, минимизация, карты Карно, неопределенные коэффициенты, первичные импликаты, минитермы, функциональная схема, триггер.

# Содержание

[ЗАДАНИЕ 2](#_Toc60127684)

[РЕФЕРАТ 3](#_Toc60127685)

[Содержание 4](#_Toc60127686)

[Задание 1. Анализ сигнальных графов. 7](#_Toc60127687)

[1.1 Выбор варианта задания 7](#_Toc60127688)

[1.2 Преобразование структурной схемы к сигнальному графу 7](#_Toc60127689)

[1.2 Преобразование структурной схемы к сигнальному графу 8](#_Toc60127690)

[1.4 Матрица инцидентности 9](#_Toc60127691)

[1.5 Построение бинарных матриц путей выхода для заданных контрольных точек. 10](#_Toc60127692)

[1.6 Бинарная матрица контуров. 12](#_Toc60127693)

[1.7 Матрица касания контуров 12](#_Toc60127694)

[1. 8 Матрица касания путей и контуров 13](#_Toc60127695)

[1.9 Формула Мэзона для заданного сигнального графа 13](#_Toc60127696)

[Задание 2. Синтез комбинационных схем. 16](#_Toc60127706)

[2.1 Определение поставленной задачи 16](#_Toc60127707)

[2.2 Составление логических функций 19](#_Toc60127708)

[2.2.1 Дизъюнктивная совершенная нормальная форма 19](#_Toc60127709)

[2.2.2 Конъюнктивная совершенная нормальная форма 20](#_Toc60127710)

[2.3 Минимизация булевых функций 20](#_Toc60127711)

[2.3.1 Пример минимизации методом неопределенных коэффициентов 21](#_Toc60127712)

[2.3.2 Пример минимизации методом Квайна-Мак-Класки. 22](#_Toc60127713)

[2.3.3 Пример минимизации картами Карно 25](#_Toc60127714)

[2.4 Совместная минимизация всех функций 26](#_Toc60127715)

[2.5 Запись МДНФ в заданном базисе 27](#_Toc60127716)

[3. СИНТЕЗ АВТОМАТА С ПАМЯТЬЮ 29](#_Toc60127717)

[3.1 Анализ технического задания 29](#_Toc60127718)

[3.2 Формальное описание абстрактного автомата 29](#_Toc60127719)

[3.3 Кодирование входных и выходных символов состояний 31](#_Toc60127720)

[3.4 Обобщенная функциональная схема структурного автомата 32](#_Toc60127721)

[3.5 Каноническая система логических уравнений 33](#_Toc60127722)

[3.6 Минимизация логических функций 35](#_Toc60127723)

[*3*.7Построение комбинационной схемы автомата с памятью 35](#_Toc60127724)

[ЗАКЛЮЧЕНИЕ 36](#_Toc60127725)

[Приложение 1. 37](#_Toc60127726)

[Приложение 2 38](#_Toc60127727)

# Задание 1. Анализ сигнальных графов.

## Выбор варианта задания

Из букв, образующих фамилию, имя и отчество получим три множества А, В и С символов русского алфавита.

**Хоменко А={Х, О, М, Е, Н, К}**

**Дмитрий B={Д, М, И, Т, Р, Й}**

**C={И, Г, О, Р, Е, В, Ч}**

Произведя соответствующие операции над множествами получим их мощности. Из таблицы возможных мощностей методического указания выбираются типы соответствующих полученным результатам типы соединений элементов в системе автоматического управления.

**⏐A∪B⏐=⏐{ Х, О, М, Е, Н, К , Д, И, Т, Р, Й }⏐=11**

**⏐( A∪B)∩С⏐=⏐{Е, И, О, Р}⏐=4**

**⏐C\A⏐=⏐{И, Г, Р, В, Ч}⏐=5**

**⏐A∪B⏐=⏐U\ A∪B⏐=33-11=22**

По полученным результатам построим схему автоматического управления системой.

## 1.2 Преобразование структурной схемы к сигнальному графу

**Рисунок 1.1.1**

**W1**

**W2**

**W5**

**W6**

**W7**

**W8**

**W4**

**W3**

x

**x1**

**x2**

**x3**

**x4**

**x5**

**x6**

**x8**

**x9**

x

**x10**

y

**x11**

**x12**

**x13**

**x7**

Граф прохождения сигнала **G=<x, θ>,** где Х – множество вершин, **θ** - множество дуг, имеет следующие особенности.

1. Каждой вершине графа **x*i*∈X** ставится в соответствие одна переменная структурной схемы (обозначение переменных сигналов приведено на рисунке 1.1).
2. Каждой дуге **(x*i*, x*j*)∈X** поставлена в соответствие передаточная функция одного из блоков структурной схемы.
3. Если из вершины исходит несколько дуг, то для них входная величина общая. Это устраняет в графе точки разветвления.
4. Если в вершину входит несколько ребер, то соответствующая этой вершине переменная равна сумме входных сигналов. Это делает не нужным использование в графе сумматоров.

Учитывая перечисленные особенности перехода от структурной схемы к сигнальному графу, перейдем от схемы рис. 1.1 к соответствующему сигнальному графу (см. рис. 1.2).

Вершины отмеченные серым цветом – это заданные контрольные точки.1.3 Матрица смежности

**Рисунок 1.2.1**

**w1**

**w5**

**w6**

**w2**

**w7**

**w8**

**w4**

**w3**

**u9**

**u10**

**u11**

**u12**

**u15**

**u16**

**u13**

**u17**

**u18**

**u19**

**u20**

**u14**

Матицей смежности графа G называется матрица R=[rij] размером nxn, где n – число вершин графа, в которой



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | x | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 | x10 | x11 | x12 | x13 | y |
| x | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| x5 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x7 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| x8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| x9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| x10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| x11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| x12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| x13 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| y | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

## 1.4 Матрица инцидентности

Матрицей инцидентности графа G называется матрица S=[sij] размера nxm, где n – число вершин графа, а m – число дуг графа, в которой:



Для построения графа пронумеруем все дуги графа в произвольном порядке, но с учетом нумерации передаточных функций.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | w1 | w2 | w3 | w4 | w5 | w6 | w7 | w8 | u9 | u10 | u11 | u12 | u13 | u14 | u15 | u16 | u17 | u18 | u19 | u20 |
| x | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 |
| x1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x2 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x3 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | -1 | -1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x5 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x6 | 0 | 0 | 0 | 0 | -1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x7 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| x8 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| x9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| x10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | 1 | 0 | 0 | 0 |
| x11 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | 0 |
| x12 | 0 | 0 | -1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x13 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| y | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | -1 | 1 | 0 | 0 |

## 1.5 Построение бинарных матриц путей выхода для заданных контрольных точек.

Согласно заданию на курсовую работу выделено множество **К** контрольных точек (выходов). Оно имеет вид:

**К={x1, x4, y, x13}**

Построим матрицы путей для каждого из этих выходов.

Бинарная матрица **P=||*pij*||** путей размера *l*x*m,* где *l* – число путей, строится по следующему правилу:



*Матрица путей выхода для x1*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | w1 | w2 | w3 | w4 | w5 | w6 | w7 | w8 | u9 | u10 | u11 | u12 | u13 | u14 | u15 | u16 | u17 | u18 | u19 | u20 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Матрица путей выхода для x4*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | w1 | w2 | w3 | w4 | w5 | w6 | w7 | w8 | u9 | u10 | u11 | u12 | u13 | u14 | u15 | u16 | u17 | u18 | u19 | u20 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Матрица путей выхода для y*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | w1 | w2 | w3 | w4 | w5 | w6 | w7 | w8 | u9 | u10 | u11 | u12 | u13 | u14 | u15 | u16 | u17 | u18 | u19 | u20 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 4 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 6 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

*Матрица путей выхода для x13*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | w1 | w2 | w3 | w4 | w5 | w6 | w7 | w8 | u9 | u10 | u11 | u12 | u13 | u14 | u15 | u16 | u17 | u18 | u19 | u20 |
| x | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| x1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| x2 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| x3 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| x4 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| x5 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

## 1.6 Бинарная матрица контуров.

Бинарная матрица контуров **C=||*cij*||** размера ***h*x*m***, где ***h*** - число контуров, строится по следующему правилу:



Предварительно пронумеруем все контуры в произвольном порядке.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | w1 | w2 | w3 | w4 | w5 | w6 | w7 | w8 | u9 | u10 | u11 | u12 | u13 | u14 | u15 | u16 | u17 | u18 | u19 | u20 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 2 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 4 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 5 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 7 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

## 1.7 Матрица касания контуров

Бинарная матрица контуров **C*k*=||*cij*||** размера ***h*x*k***, где ***k*** - число контуров, строится по следующему правилу:



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

## 1. 8 Матрица касания путей и контуров

Бинарная матрица контуров **C*l*=||*cij*||** размера ***l*x*k***, где ***l*** - число путей для заданного выхода, строится по следующему правилу:



**Для x1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

**Для x4**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

**Для y**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

**Для x13**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

## 1.9 Формула Мэзона для заданного сигнального графа

Используя универсальную топологическую формулу, носящую имя Мэзона, можно получить передачу между любыми двумя вершинами. Формула имеет следующий вид:



где ****- передача ***k***-го пути между вершинами ***j*** и ***r;*** **Δ -** определитель графа. Он характеризует контурную часть графа и имеет следующий вид:



где, **L** – множество индексов контуров, ***L2***- множество пар индексов не касающихся контуров, ***L3***- множество троек индексов не касающихся контуров, ***Ki*** – передача i-го контура,  - минор пути, это определитель подграфа, полученного удалением из полного графа вершин и дуг, образующих путь .

**Δ=1-К1-К2-К3-К4-К5-К6-К7-К8+К7К2+К7К3+К7К5+К7К6+К7К8=1- К1-К2-К3-К4-К5-К6-К7-К8+К7(К2+К3+К5+К6+К8)**

**К1=W1W3W4W5W6**

**K2=W3W4W7**

**K3=W1W3W4W8**

**K4=W2W3W4W6 W7**

**K5=W2W3W4W7**

**K6=W2W3W4W8**

**K7=W5W6**

**K8=W3W4**

**Δ=1- W3W4(W1W5W6+ W7+ W1W8+ W2W6 W7+ W2W7+2W2W8+ 1)+ W5W6(W3W4(W7+ W1W5W6+ W2W7+ W2W8+1)-1)**

**Для x1**

****

****

****

**Для x4**

****



** **



**Для y**



****

****

****

****

****



****

# 

****

# 

# 

# 

# Для х13

****

****

****

****

****

****

****

****

# 

****

# 

# 

# 

# Задание 2. Синтез комбинационных схем.

## 2.1 Определение поставленной задачи

Устройство, работа которого может быть представлена на языке алгебры высказываний, принято называть логическим. Пусть такое устройство имеет ***n*** выходов и ***m*** входов. На каждый вход может быть подан произвольный символ конечного множества **Х,** называемого входным алфавитом. Совокупность входных символов, поданных на входы устройства, образует входное слово **Р*i*** в алфавите **Х**. На выходе устройства появляются выходные слова ***Qj***, составленные из символов выходного алфавита **Y**. В силу конечности алфавитов **X, Y** и слов **P*i*, Q*j*** (длина слова всегда равна ***m***, а выходного слова - ***h***) общее количество различных входных и выходных слов также конечно.

Элементарный такт работы устройства состоит в том, что при появлении на входе слова **Р*i*** устройство выдает на выходах комбинацию символов **y*i***, образующих слово **Q*j***. Если слово **Q*j*** определяется только входным словом на данном такте, то устройство называется конечным автоматом без памяти, или комбинационной схемой.

Алгоритм функционирования комбинационного устройства будет определен, если задать таблицу соответствия **{P*i*}->{Q*j*}** для всех слов **P*i***. Если входной алфавит **X** состоит из **K** различных символов, в таблице соответствия будет **K*m*** строк. Так как символы входного и выходного алфавитов принимают только два значения (в данном случае «1» или «0»), то при синтезе и анализе логического устройства применяется булева алгебра.

Произвольные входной и выходной алфавиты могут быть приведены к автомату с двойным входом и выходом путем соответствующего кодирования. Однако этот автомат должен оперировать со словами входного и выходного алфавитов, длина которых больше длин соответствующих слов исходного алфавита.

Под синтезом комбинационной схемы подразумевается построение логической схемы проектируемого устройства в заданном базисе логических элементов. Исходным материалом к синтезу является словесное описание работы устройства.

Согласно заданию на курсовое проектирование было предложено закодировать исходный алфавит кодом Грея и использовать для синтеза конечного автомата базис {и, не}.

Код Грея является циклическим кодом, получается из двоично-десятичного кода по следующим правилам:

1. пусть **gn…..g1g0** – кодовый набор в коде Грея с (n+1) разрядами.
2. **bn…b1b0** – соответствующее двоичное число.
3. тогда разряд **g0** получается из следующего выражения:

**gi=bi⊕bi+1; 0≤i≤n-1; gn=bn;** где **⊕** - символ операции сложения по модулю 2 (0+0=0, 0+1=1, 1+0=1, 1+1=0).

Закодируем входной алфавит в соответствии с этими правилами и с учетом значений **yi** составим таблицу истинности (см. таблицу 2.1.1).

**Таблица 2.1.1**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выходной символ** | **Сигнал (код)** | | | | **y1** | **y2** | **y3** | **y4** | **y5** | **y6** | **y7** |  |
| **x4** | **x3** | **x2** | **x1** |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 2 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 3 |
| 3 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 2 |
| 4 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 6 |
| 5 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 7 |
| 6 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 5 |
| 7 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 4 |
| 8 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 12 |
| 9 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 13 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | 1 | 0 | 1 | 0 | \* | \* | \* | \* | \* | \* | \* | 8 |
| 11 | 1 | 0 | 1 | 1 | \* | \* | \* | \* | \* | \* | \* | 9 |
| 12 | 1 | 1 | 1 | 0 | \* | \* | \* | \* | \* | \* | \* | 10 |
| 13 | 1 | 1 | 1 | 1 | \* | \* | \* | \* | \* | \* | \* | 11 |
| 14 | 1 | 0 | 0 | 0 | \* | \* | \* | \* | \* | \* | \* | 14 |
| 15 | 1 | 0 | 0 | 1 | \* | \* | \* | \* | \* | \* | \* | 12 |

Если не удается заполнить все строки этой таблицы, то функция называется не полностью определенной, а наборы на которых она не определена, носят название запрещенных. В этом случае схема называется неполной. Доопределение функции производится произвольно.

## 2.2 Составление логических функций

Существует два способа записи логической функции по таблице истинности: запись дизъюнктивной совершенной нормальной формы (ДСНФ) и запись конъюнктивной совершенной нормальной формы (КСНФ). В первом случае образуют дизъюнкции, соответствующие входным наборам, на которых функция принимает значение «1», их объединяют знаками конъюнкции. Во втором случае организуют конъюнкции, соответствующие входным словам, на которых функция принимает значение «0», эти конъюнкции объединяют знаками дизъюнкции. Рассмотрим на примере функции у3.

### 2.2.1 Дизъюнктивная совершенная нормальная форма

Введем понятие логической степени, которою будем обозначать . Тогда, логическая степень будет определяться по формуле (2.1)

 (2.1)

Любую функцию от n переменных можно представить в виде (см.(2.2))

 (2.2)

означает, что коллективные члены формируются только для таких наборов (α1, α2,…, αn) для которых функция принимает единичное значение.

Форма (2.2) определяет алгоритм перехода от таблицы истинности к аналитическому ее виду. Для такого перехода используется следующий алгоритм:

а) Выбрать в таблице истинности все наборы, в которых функция обращается в единицу.

б) Выписать конъюнкции, соответствующие этим наборам аргументов. При этом если аргумент *х*i входит в данный набор как «1», то он вписывается без изменения в конъюнкцию, соответствующую данному набору. Если же входит в данный набор как «0», то в соответствующую конъюнкцию вписывается его отрицание.

в) Все полученные конъюнкции объединяют знаком дизъюнкции.

Для функции у3 СДНФ будет иметь вид:



### 2.2.2 Конъюнктивная совершенная нормальная форма

Любую функцию от n переменных можно представить в виду:

f= 

означает, что коллективные члены формируются только для таких наборов, (α1, α2, …, αn) для которых функция принимает нулевое значение.

Следующий алгоритм позволяет перейти от табличной записи функции к СКНФ:

а) Выбрать в таблице истинности все наборы, в которых функция обращается в «0».

б) Выписать дизъюнкции, соответствующие этим наборам аргументов. При этом если аргумент *х*i входит в данный набор как «0», то он вписывается без изменения в дизъюнкцию, соответствующую данному набору. Если же входит в данный набор как «1», то в соответствующую дизъюнкцию вписывается его отрицание.

в) Все полученные дизъюнкции объединяют знаком конъюнкции

Функция У3 в виде СКНФ будет иметь вид:



## 2.3 Минимизация булевых функций

Минимизация сводится к поиску минимальных форм (ДНФ и КНФ). Минимальной форме соответствует самая простая логическая схема, реализующая данную функцию и требующая минимальных аппаратных затрат.

В ходе курсового проектирования была выполнена минимизация полученных булевых функций следующими методами:

1. метод неопределенных коэффициентов
2. метод Квайна-Мак-Класки
3. карты Карно

Для примера в курсовом проекте рассмотрена минимизация этими способами для функции y3.

### 2.3.1 Пример минимизации методом неопределенных коэффициентов

Данный метод является по своим идеям наиболее простым. Для функции записываются все возможные конъюнктивные члены, которые могут входить в дизъюнктивную форму представления функции. Коэффициенты К с различными индексами являются неопределенными и подбираются так, чтобы получающаяся после этого дизъюнктивная форма была минимальной.

Если теперь задавать всевозможные наборы значений аргументов и приравнивать полученное после этого выражение значению функции на выбранных наборах, то получим систему 24 уравнений для определения коэффициентов К:



Находим выражения, имеющие в правой части ноль. Исходя из определения дизъюнкции вычеркиваются все элементы в этих уравнениях и эти же элементы находящиеся в других уравнениях. В итоге получим уравнения:



Из оставшихся коэффициентов приравниваем единице коэффициент, определяющий конъюнкцию наименьшего возможного ранга, а остальные коэффициенты в левой части данного уравнения приравняем нулю(это можно сделать, так как дизъюнкция обращается а единицу, если хотя бы один член ее равен единице). Тогда уравнения примут вид:





### 2.3.2 Пример минимизации методом Квайна-Мак-Класки.

При минимизации по данному методу предполагается, что минимизируемая функция задана в СДНФ. Метод Квайна – Мак – Класки состоит из последовательного выполнения следующий этапов.

Метод Квайна состоит из последовательного выполнения этапов:

1. Нахождение первичных импликант
2. Расстановка меток
3. Нахождение существенных импликант
4. Вычеркивание лишних столбцов
5. Вычеркивание лишних
6. Получение минимального покрытия

Выполним, приведенные этапы, для функции У3.

Нахождение первичных импликант. Все минитермы (элементарные конъюнкции, входящие в ДНФ) данной функции записываются в виде их двоичных номеров. Все номера разбиваются по числу единиц в этих номерах на не пересекающиеся группы. При этом в *i*-тую группу войдут все номера, имеющие в своей двоичной записи ровно *i* единиц. По парное сравнение (склеивание) можно производить только между соседними по номеру группами. При образовании минитермов с рангом выше нулевого в разряды, соответствующие исключенным переменным, пишется знак тире. Минитермы, не склеивающиеся между собой, называются первичными импликантами.

СДНФ, которую мы нашли ранее, для функции У3 имеет вид:



Составим минитермы для этой функции:

**Таблица 2.2.1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Минитермы длиной 4** | | **Минитермы длиной 3** | |
| *Нулевая группа* | 0000+ | *Нулевая группа* | 0-00 |
| *Первая группа* | 0100+ | *Первая группа* | -100, -011 |
| *Вторая группа* | 1100, 1010, 0011 | *Вторая группа* | 11-0, 1-10, 101- |
| *Третья группа* | 1110, 1011 |  |  |

Расстановка меток. Остальные этапы нужны, чтобы отбросить некоторые первичные импликанты. На данном этапе составляется таблица, число строк которой равно числу полученных первичных импликант, число столбцов совпадает с числом минитермов СДНФ. Если в некоторый минитерм СДНФ входит какая – либо из первичных импликант, то на пересечении соответствующего столбца и строки ставится метка. В таблице 2.2.2 приведем результат расстановки меток:

**Таблица 2.2.2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **0000** | **0100** | **1100** | **1010** | **0011** | **1110** | **1011** |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | **0-00** | У | У |  |  |  |  |  |
| 2 | **-100** |  | У | У |  |  |  |  |
| 3 | **-011** |  |  |  |  | У |  | У |
| 4 | **11-0** |  |  | У |  |  | У |  |
| 5 | **1-10** |  |  |  | У |  | У |  |
| 6 | **101-** |  |  |  | У |  |  | У |

Нахождение существенных импликант. Если в каком – либо из столбцов таблицы меток стоит только одна метка, то первичная импликанта, стоящая в соответствующей строке, называется существенной импликантой. Все существенные импликанты запоминаются. А из таблицы меток исключаются строки, соответствующие существенным импликантам, и столбцы минитермов «покрываемых» этими существенными импликантами.

Существенными являются импликанты 0-00 и -011. Поэтому вычеркиваем 1-ю и 3-ю строки и столбцы: 1, 5, 2, 7.

Составим сокращенную таблицу меток:

**Таблица 2.2.3**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **1100** | **1010** | **1110** |
| **-100** | У |  |  |
| **11-0** | У |  | У |
| **1-10** |  | У | У |
| **101-** |  | У |  |

Выбор минимального покрытия. Исследуется результирующая таблица. Выбирается такая совокупность первичных импликант, которая иссключает метки во всех столбцах (по крайней мере по одной в каждом столбце). При нескольких возможных вариантах такого выбора отдается предпочтение варианту покрытия с минимальным суммарным числом букв в простых импликантах, образующих покрытие.

С учетом существенных импликант получим две МДНФ для этой функции имеет вид:

1.

2. 

Число букв составляющих простые импликации в каждом варианте одинаково. Во втором варианте на одно отрицание меньше, поэтому берем его за искомое:



### 2.3.3 Пример минимизации картами Карно

Данный метод для минимизации функции в коде Грея. В каждую ячейку записывается значение функции на данном наборе. Затем выделяются группы ячеек размером 2a\*2b , где a, bε[0,1,2…], в которых функция принимает значение «1». В каждую группу должно входить максимальное число ячеек. Таких групп должно быть минимальное количество. Каждой группе будет соответствовать конъюнктивный член размером *n-a-b.* Для получения МДНФ каждую группу надо просматривать в горизонтальном и вертикальном измерениях, с нахождения таких переменных, которые не меняют своего значения в пределах группы. Если переменная не меняет своего нулевого значения, то она вписывается в конъюнкцию с отрицанием, если не меняет своего единичного значения, то вписывается без отрицания. Если имеются разорванные группы, то карту Карно надо свернуть в цилиндр. На неопределенных наборах следует доопределить нулем или единицей, в соответствии с выбираемой группой ячеек. Каждая единичная ячейка должна быть включена хотя бы в одну группу.

Составим карту Карно для функции У3 (рисунок 2.3.1).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | x3x4 | | | | |
| x1x2 |  | 00 | 01 | 11 | 10 |
| 00 | 1 |  | 1 |  |
| 01 | 1 |  |  |  |
| 11 | 1 |  |  | 1 |
| 10 |  |  | 1 | 1 |

**Рис. 2.3.1 Карта Карно для функции У3**

Таким образом, для функции У3 в МДНФ будет иметь следующий вид:



## 2.4 Совместная минимизация всех функций

Синтез схем на основе отдельно минимизированных функций является неоптимальным, с точки зрения количества используемых элементов. Так как вероятнее всего, имеются такие конъюнкции, которые дублируют друг друга. Целью данного пункта является нахождение этих конъюнкций.

Для этого составим карты Карно для каждой функции из таблицы истинности (таблица 2.1.1). Доопределим ее запрещенные наборы (таблица 2.1.1), а затем сгруппируем ячейки таким образом, чтобы таких групп было минимальное количество на данной карте и максимальное совпадение таких групп между картами для остальных функций.

Составим карты Карно для всех функций таблицы истинности (таблица 2.1.1)

**y5**

**y4**

**y3**

**y2**

**y1**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | \* | 1 |  |  |  | 1 | \* | 1 |  |  |  | 1 | \* | 1 |  |  |  | 1 | \* | 1 | 1 |  |  |  | \* | 1 |  |
|  | \* | \* | 1 |  |  |  | \* | \* |  |  |  | 1 | \* | \* |  |  |  | 1 | \* | \* | 1 |  |  | 1 | \* | \* | 1 |
|  | \* | \* | 1 |  |  | 1 | \* | \* |  |  |  | 1 | \* | \* | 1 |  |  |  | \* | \* | 1 |  |  | 1 | \* | \* | 1 |
|  | \* | 1 | 1 |  |  |  | \* |  | 1 |  |  |  | \* | 1 | 1 |  |  | 1 | \* | 1 | 1 |  |  |  | \* | 1 | 1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1  **y6**  **y7** | \* | 1 | 1 |  |  | 1 | \* | 1 | 1 |
| 1 | \* | \* | 1 |  |  | 1 | \* | \* |  |
| 1 | \* | \* |  |  |  | 1 | \* | \* | 1 |
| 1 | \* | 1 |  |  |  |  | \* | 1 | 1 |

Тогда МДНФ этих функций будет иметь вид:



## 2.5 Запись МДНФ в заданном базисе

Система функций, полученная в пункте 2.4 была записана в системе базисных функций отрицания, конъюнкции и дизъюнкции. Данный базис является не минимальным, и может быть уменьшен за счет выбрасывания из него конъюнкции или дизъюнкции. После чего полученные базисы являются минимальными.

Запишем функции, полученные в пункте 2.4 в базисе {**И-НЕ**}. Для этого примем правила Де Моргана. Тогда функции будут иметь вид:



Рассмотренная проблема минимизации имеет большое значение для практических целей. Если выбор той или иной базисной системы функций предопределяет выбор стандартного набора типовых логических элементов, то решение проблемы минимизации связано с проблемой экономной реализации различных схем и устройств на базе выбранных типовых элементов.

**2.6 Построение функциональной схемы**

Каждой логической функции можно поставить в соответствие особое устройство модулирующие данную функцию. Это устройство называется логическим элементом. Устройство имеет один вход и n выходов (n – число аргументов логической функции, причем i – ый вход соответствует i – му аргументу ).

Стандартные графические обозначения логических элементов приведены на рис. 2.6.1.

&

*х*

**

&

*х1*

*х2*



1

*х1*

*х2*



1

*х*

**

а) б) в) г)

**Рис. 2.6.1 графическое обозначение логических элементов**

а)элемент «**НЕ**» б) элемент «**ИЛИ**» в) элемент «**И**» г) элемент в базисе «**И-НЕ**»

Построим функциональную схему на основе базиса {**И-НЕ**}( Приложение 1).

# 3. СИНТЕЗ АВТОМАТА С ПАМЯТЬЮ

## 3.1 Анализ технического задания

В данном разделе курсовой работы необходимо синтезировать автомат с памятью на основе содержательного описания алгоритма его работы.

Автомат управляет грузовым лифтом.

Грузовой лифт, обслуживает трехэтажный магазин, имеет кнопку вызова на каждом этаже и работает по следующим правилам: если нажата одна кнопка, то лифт движется на соответствующий этаж; если нажато одновременно несколько кнопок, то лифт движется на самый нижний из всех этажей на которые нажаты кнопки. Ни одна кнопка не может быть нажата во время движения.

Согласно заданию на курсовую работу, в качестве элементов памяти следует использовать D – триггеры.

## 3.2 Формальное описание абстрактного автомата

Абстрактный автомат зададим в виде автомата Милли.

Для формального описания абстрактного автомата необходимо задать входной алфавит, выходной алфавит, множество состояний, функцию переходов и функцию выходов.

Автомат имеет три входа, соответствующих различным комбинациям подаваемых сигналов.

Входной алфавит:

Х={х1,х2,х3},

где х1 означает, что нажата кнопка первого этажа; х2 означает, что нажата кнопка второго этажа, х3 означает, что нажата кнопка третьего этажа.

Выходной алфавит:

У= {у0, у1, у2, у3, у4},

где

у0 – лифт стоит на месте,

у1 – лифт едет на один этаж вверх,

у2 - лифт едет на один этаж вниз,

у3 - лифт едет на два этажа вверх,

у4 - лифт едет на два этажа вниз.

Множество состояний:

S= { s1, s2, s3};

где s1- лифт на первом этаже; s2 – лифт на втором; s3 – лифт на третьем этаже.

Теперь, можно составить таблицы переходов – входов и переходов выходов.

Таблица переходов – входов представлена в таблице 3.2.1 .

**Таблица 3.2.1**

|  |  |  |  |
| --- | --- | --- | --- |
| **S** | **s1** | **s2** | **s3** |
| **х1 х2 х3** |
| 0 0 0 | s1 | s1 | s1 |
| 0 0 1 | s2 | s2 | s2 |
| 0 1 0 | s1 | s1 | s1 |
| 0 1 1 | s3 | s3 | s3 |
| 1 0 0 | s1 | s1 | s1 |
| 1 0 1 | s3 | s3 | s3 |
| 1 1 0 | s2 | s2 | s2 |
| 1 1 1 | s1 | s1 | s1 |

Таблица переходов – выходов представлена в таблице 3.2.2 .

**Таблица 3.2.2**

|  |  |  |  |
| --- | --- | --- | --- |
| S | s1 | s2 | s3 |
| х1 х2 х3 |
| 0 0 0 | у0 | у0 | у0 |
| 0 0 1 | у0 | у2 | у4 |
| 0 1 0 | у1 | у0 | у2 |
| 0 1 1 | у0 | у2 | у4 |
| 1 0 0 | у3 | у1 | у0 |
| 1 0 1 | у0 | у2 | у4 |
| 1 1 0 | у1 | у0 | у2 |
| 1 1 1 | у0 | у2 | у4 |

Для того, чтобы хранить текущее состояние требуется **n=[logθM]** элементов памяти, где М – мощность алфавита состояний, θ – число состояний элементов памяти. Таким образом, необходимо log23=2 элементов памяти.

## 3.3 Кодирование входных и выходных символов состояний

Кодирование входных символов представлено в таблице 3.3.1 .

**Таблица 3.3.1**

|  |  |  |  |
| --- | --- | --- | --- |
| Х | х3 | х2 | х1 |
| х1 | 0 | 0 | 0 |
| х2 | 0 | 0 | 1 |
| х3 | 0 | 1 | 0 |
| х4 | 0 | 1 | 1 |
| х5 | 1 | 0 | 0 |
| х6 | 1 | 0 | 1 |
| х7 | 1 | 1 | 0 |
| х8 | 1 | 1 | 1 |

Кодирование выходных символов представлено в таблице 3.3.2 .

**Таблица 3. 3.2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | у1 | у2 | у3 |
| у0 | 1 | 0 | 1 |
| у1 | 0 | 0 | 0 |
| у2 | 1 | 0 | 0 |
| у3 | 1 | 1 | 1 |
| у4 | 1 | 1 | 0 |

Кодирование состояний автомата представлено в таблиц 3.3.3.

**Таблица 3.3.3**

|  |  |  |
| --- | --- | --- |
| S | t1 | t2 |
| s1 | 0 | 0 |
| s2 | 0 | 1 |
| s3 | 1 | 1 |

В соответствии с таблицами 3.3.1 – 3.3.3 составляем таблицу переходов – входов в кодированном виде.

**Таблица 3.3.4**

|  |  |  |  |
| --- | --- | --- | --- |
| х3х2х1\s1s2s3 | 00 | 01 | 11 |
| 000 | 00 | 01 | 11 |
| 001 | 00 | 00 | 00 |
| 010 | 01 | 01 | 01 |
| 011 | 00 | 00 | 00 |
| 100 | 11 | 11 | 11 |
| 101 | 00 | 00 | 00 |
| 110 | 01 | 01 | 01 |
| 111 | 00 | 00 | 00 |

А также составим кодированную таблицу переходов выходов.

**Таблица 3.4.5**

|  |  |  |  |
| --- | --- | --- | --- |
| х3х2х1\s1s2s3 | 00 | 01 | 11 |
| 000 | 101 | 101 | 101 |
| 001 | 101 | 100 | 110 |
| 010 | 000 | 101 | 100 |
| 011 | 101 | 100 | 110 |
| 100 | 111 | 000 | 101 |
| 101 | 101 | 100 | 110 |
| 110 | 000 | 101 | 100 |
| 111 | 101 | 100 | 110 |

## 3.4 Обобщенная функциональная схема структурного автомата

Построим обобщенную функциональную схему структурного автомата с учетом заданного типа автомата и триггера (см. рис.3.4.1) .

П1

П2

КС1

х3 х2 х1

у1 у2 у3

Ф1

Ф2

t1

t2

**Рис.3.4.1**

На рисунке 3.4.1 функциональная схема состоит из двух блоков. Первый блок - блок памяти, который состоит из двух элементов памяти (D – триггер, П1, П2). Второй блок – комбинационная схема (КС1).

## 3.5 Каноническая система логических уравнений

Из таблицы переходов выходов (табл. 3.4.5) можно получить СДНФ для у ( выходов нашего автомата).







Нахождение функций возбуждения памяти ( Ф1, Ф2) производится в соответствии с типом триггера. D – триггер имеет один вход и один выход, его изображение приведено на рис. 3.5.1 .

**ϕ**

**D**

Рис.3.2 D – триггер

**τ**

Функция переходов данного автомата памяти приведен в таблице 3.8 . Если закодировать входные и выходные символы D – триггера (табл. 3.9.,3.10), то кодированная таблица переходов примет вид таблицы 3.5.1.

**Таблица 3.5.1**

|  |  |  |
| --- | --- | --- |
| ϕ\τ | 0 | 1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

При построении функций возбуждения памяти автомата строят функцию входов элемента памяти. Эту функцию задают в виде таблице. Функция входов структурного автомата памяти П приведена в таблице 3.5.2 .

**Таблица 3.5.2**

|  |  |  |
| --- | --- | --- |
| tисх | Ф | tнов |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Функция возбуждения автомата памяти, построенная в соответствии с таблицами 3.12 и 3.7 представлена в таблице 3.13 .

**Таблица 3.5.3**

|  |  |  |  |
| --- | --- | --- | --- |
| х3х2х1\t1t2 | 00 | 01 | 11 |
| 000 | 00 | 01 | 11 |
| 001 | 00 | 00 | 00 |
| 010 | 01 | 01 | 01 |
| 011 | 00 | 00 | 00 |
| 100 | 11 | 11 | 11 |
| 101 | 00 | 00 | 10 |
| 110 | 01 | 01 | 01 |
| 111 | 00 | 00 | 10 |

Из этой таблицы для единичных значений функции Ф1 и Ф2 имеем:





## 3.6 Минимизация логических функций

С помощью программы Logic можно минимизировать, полученные в прошлом разделе логические функции. В итоге они примут вид:





# 3.7 Построение комбинационной схемы автомата с памятью

Схема автомата с памятью основанной на D-триггере представлена в Приложении 2.

# ЗАКЛЮЧЕНИЕ

В курсовой работе по дисциплине « Математические основы теории систем» были выполнены два раздела, в которых были закреплены теоретические знания по темам: анализ сигнального графа и синтез комбинационных схем.

В первом разделе исследуется сигнальный граф: преобразование структурной схемы к сигнальному графу, расчет передач графа. Для описания графа, использовались его структурные характеристики: матрица смежности, матрица касания путей и контуров, матрица инциденций, матрица путей, матрица контуров, матрица касания контуров.

Во втором разделе были получены некоторые навыки в области синтеза комбинационных схем. Результатом проделанной работы явилась комбинационная схема управления светодиодами семисегментного индикатора.

В третьем разделе необходимо синтезировали автомат с памятью на основе содержательного описания алгоритма его работы.

# Приложение 1.

# Приложение 2