Белорусский национальный технический университет

Факультет информационных технологий и робототехники

Кафедра "Электропривод и автоматизация промышленных установок и технологических комплексов"

СОГЛАСОВАНО
Заведующий кафедрой
____
__ _____________20__г.

СОГЛАСОВАНО
Декан факультета
____
__ __________________20__г.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

«Проектирование дискретных систем управления»

Для специальности 1-53 01 05 «Автоматизированные электроприводы»

Составитель: Мигдалёнок Александр Анатольевич

Рассмотрено и утверждено
На заседании совета ФИТР __ ____________________20___г.
протокол №_____

Перечень материалов

1. Конспект лекций по дисциплине
2. План практических занятий
3. Методическое пособие по выполнению курсовой работы
4. Типовые задания к зачёту
5. Учебная программа по дисциплине и информационно-аналитические материалы, рекомендуемые для изучения учебной дисциплины

Пояснительная записка

Цель УМК по дисциплине «Проектирование дискретных систем управления» заключается в приобретении студентами знаний и умений для описания, проектирования и наладки дискретных систем управления и закрепления этих знаний в процессе выполнения практических задач и самостоятельной работы.

Теоретический раздел УМК содержит материалы для теоретического изучения дисциплины и представлен конспектом лекций. В практическом разделе представлен план практических занятий, а так же задание и методическое пособие по выполнению курсовой работы. В методическом пособии подробно рассматривается пример выполнения курсовой работы по одному из вариантов задания. Раздел контроля содержит описание типовых заданий для проведения зачёта в письменной форме. Вспомогательный раздел содержит учебную программу по дисциплине, а также перечень учебных изданий и информационно-аналитических материалов, рекомендуемых для изучения данной учебной дисциплины.

При работе с УМК в первую очередь изучается теоретическая часть с подробным рассмотрением примеров, представленных в конспекте лекций. Для закрепления теоретических знаний проводятся практические занятия в соответствии с предложенной тематикой. В ходе самостоятельной работы студенты выполняют курсовую работу по теме «Проектирование схемы управления перемещением механизма по заданному циклу». Контроль полученных знаний и умений осуществляется в виде письменного зачёта с решением типовых заданий. В качестве вспомогательного материала может использоваться рабочая программа дисциплины, а так же информационно-аналитические материалы, рекомендуемые для изучения данной учебной дисциплины.

СОДЕРЖАНИЕ УМК

1. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

Теоретический раздел УМК представлен конспектом лекций по дисциплине «Проектирование дискретных систем управления» для студентов специальности 1-53 01 05 «Автоматизированные электроприводы».

ВВЕДЕНИЕ

Цель курса: изучение формализованных методов проектирования дискретных управляющих устройств и применение этих методов при решении инженерных задач.

Действие любой системы управления можно описать функциональными зависимостями между входными и выходными сигналами.

Системы могут быть дискретными и непрерывными.

Устройства, сигналы на входах и выходах которых могут принимать конечное число значений называются *устройствами релейного действия* или *дискретными устройствами*.

Дискретное устройство в общем случае можно рассматривать как m, n – полюсник с n входами и m выходами. Кроме этого, в зависимости от типа устройства, могут присутствовать внутренние сигналы.

В общем случае дискретное устройство может быть представлено следующим образом:

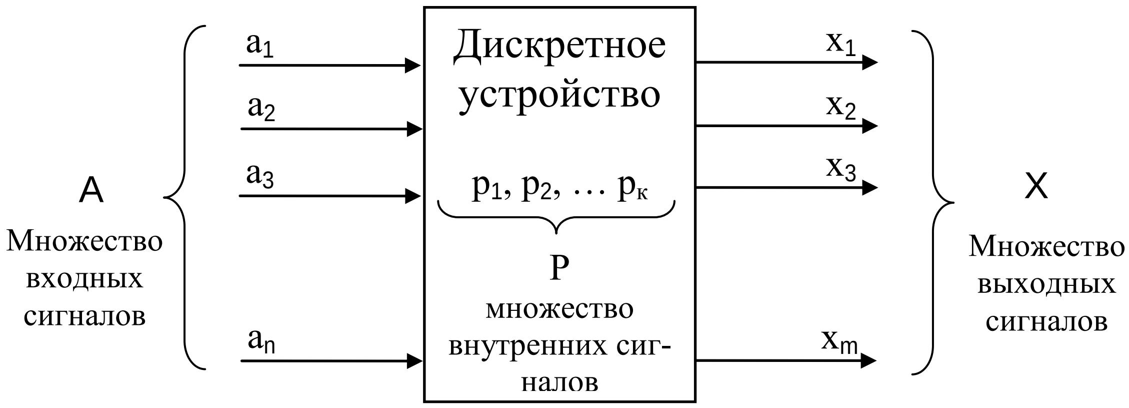


Рисунок 1.1 – Функциональная схема дискретного устройства

Модель дискретного устройства, отражающую только его свойства по переработке сигналов, называют *дискретным автоматом*.

В таком автомате выделяют три множества:

1. $A = (a_1, a_2, \ldots, a_n)$ – множество входных сигналов;
2. $X = (x_1, x_2, \ldots, x_m)$ – множество выходных сигналов;
3. $P = (p_1, p_2, \ldots, p_k)$ – множество внутренних сигналов элементов памяти.

В зависимости от принципа действия различают два вида дискретных автоматов – *комбинационные автоматы* и *дискретные автоматы с памятью*.

Комбинационный автомат (однотактная схема) – это устройство релейного действия с n входами и m выходами, у которого при идеальных (безынерционных) элементах состояние выходов (сигналы на выходах) в некоторый момент времени однозначно определяется состоянием входов (сигналами на входах) в тот же момент времени.

Дискретным автоматом с памятью называется устройство, у которого состояние выхода зависит не только от того, какие сигналы присутствуют на его входах в данный момент времени, но и от того, какие последовательности сигналов поступали на входы автомата в предшествующие моменты времени.

Для запоминания входных последовательностей, поступавших на вход устройства в предыдущие моменты времени, используются внутренние сигналы элементов памяти.

Обычно релейные устройства могут находиться в двух состояниях: включено - выключено, малое сопротивление - большое сопротивление и т. д. Такие устройства описываются двухзначной алгеброй логики (Булевой алгеброй), в которой переменные могут принимать только два значения, условно обозначаемые "0" и "1".

Ноль и единица в алгебре логики не отражают количество, а отражают состояние. Кроме двухзначной алгебры логики имеются многозначные алгебры логики и т. д..

При проектировании дискретного устройства основной задачей является получение математического описания принципа его действия. Математическое описание условий работы управляющего устройства обладает двумя достоинствами:

1. Строгостью и однозначностью.
2. Позволяет получить новое выводное знание.

Используя математическое описание можно построить принципиальную схему управляющего устройства на соответствующей элементной базе.

Для получения наиболее полного и точного математического описания принципа действия устройства используются формализованные методы.

Применение формализованных методов проектирования позволяет решить следующие задачи:

1. Выбор оптимальных вариантов структур, как отдельных блоков, так и устройства в целом.
2. Обеспечение высокой надёжности.
3. Сокращение времени проектирования, изготовления и наладки систем управления.

При этом, совсем не обязательно во всех случаях использовать формальные методы, само знание этих методов, закономерностей часто позволяет быстро и качественно решить задачу.

4. Возможность реализации дискретных устройств на различных элементах (контактных элементах, бесконтактных элементах И-НЕ, ИЛИ-НЕ и т. д.).
5. Переход к автоматизированному проектированию.

1 ЛОГИЧЕСКИЕ ОСНОВЫ ОПИСАНИЯ И ПРЕОБРАЗОВАНИЯ ДИСКРЕТНЫХ УСТРОЙСТВ

1.1 Общие определения

Логическая переменная – величина, которая может принимать только два значения ("0" или "1") истинно или ложно.

Логическая функция – функция, которая, как и её аргументы может принимать только два значения.

Так как логическая функция и логическая переменная могут принимать только два значения, то для их описания и нумерации используется двоичная система исчисления.

Для перевода двоичного числа $N_2 = a_n...a_2a_1$, где $a_i \in \{0, 1\}$ в десятичное число N_{10} необходимо каждому i-му разряду (считая с права на лево) присвоить вес $q_i = 2^{i-1}$. Тогда

$$N_{10} = a_n \cdot 2^{n-1} + a_{n-1} \cdot 2^{n-2} + ... + a_2 \cdot 2 + a_1 = \sum_{i=1}^{n} a_i \cdot 2^{i-1}$$

Пример. Перевести двоичное число $N_2 = 100101$ в десятичное.

$$N_{10} = 1*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1 = 32 + 4 + 1 = 37.$$

Для перевода десятичного числа в двоичное число поступают следующим образом: десятичное число делят на два и записывают частное и остаток (0 или 1), частное снова делят на два, запоминают остаток и так до тех пор, пока не дойдут до нуля. Остатки составят двоичное число, причём первый остаток соответствует младшему разряду, а последний – старшему разряду двоичного числа.

Пример. Перевести десятичное число $N_{10} = 246$ в двоичное.

частное	246	123	61	30	15	7	3	1
остаток	0	1	1	0	1	1	1	1

Получим двоичное число $N_2 = 11110110$

Различные комбинации значений входных переменных логической функции называются наборами. При n входных переменных количество наборов равно:

$$N = 2^n$$

Допустим, имеется логическая функция трёх переменных $x = f(a,b,c)$.

Количество входных наборов будет равно:

$$N = 2^3=8$$

Данные наборы будут иметь вид:

c	b	a
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

1.2 **Способы задания функций алгебры логики**

Существуют различные способы задания функций алгебры логики.

Функция является полностью заданной, если указаны её значения для всех входных наборов.

Рассмотрим основные способы задания функций алгебры логики.

1. *Табличный способ.*

При данном способе задания функция задаётся в виде таблицы истинности (соответствия). Данная таблица содержит 2^n строк по числу входных наборов, n столбцов по числу аргументов функции и один столбец значений функции. В таблице каждому набору входных переменных ставится в соответствие значение функции.

Пример. Задать с помощью таблицы истинности логическую функцию трёх переменных, значение которой равно единице на тех наборах, которые содержат только две единицы.

Строим таблицу истинности.

c	b	a	x
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

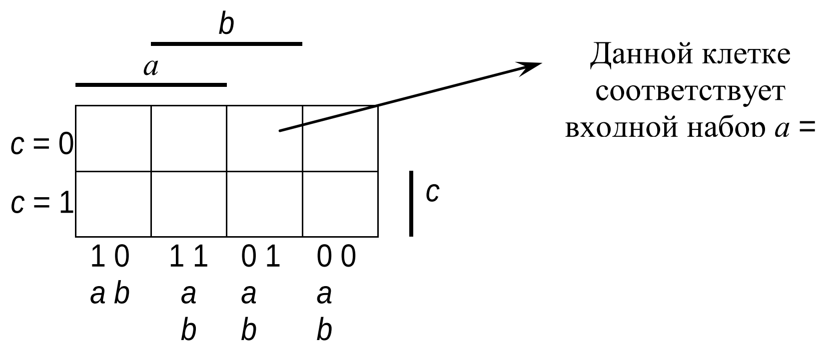
Сопоставляя каждому набору значение функции получаем задание логической функции с помощью таблицы соответствия (истинности).

2. *Координатный способ.*

При задании данным способом логическая функция задаётся в виде координатной карты соответствия, называемой картой Карно. Карта Карно содержит столько клеток, сколько входных наборов содержит логическая функция. Каждая клетка соответствует определённому входному набору. Так как карта Карно строится на плоскости, то все переменные функции делятся на две группы. Первая группа переменных определяет координаты столбцов, вторая группа – координаты строк. При этом если соответствующая клетка находится под чертой переменной, то в данном наборе она равна 1, если не под чертой, то переменная в наборе равна 0. В клетках карты Карно отмечаются значения функции на соответствующих наборах.

Пример. Задать предыдущую логическую функцию с помощью карты Карно.

Разделим переменные функции на две группы. Переменные a и b будут определять координаты столбцов, переменная c – координаты строк. Соответственно карта Карно будет иметь размерности 4×2 и содержать 8 клеток (по числу наборов). Внешний вид карты Карно будет следующий:



Для задания функции отмечаем её значение на соответствующих наборах.

b — над 2-м и 3-м столбцами; a — над 1-м и 2-м столбцами.

0	1	0	0	
1	0	1	0	c

3. *Числовой способ задания.*

При данном способе задания логическая функция задаётся в виде десятичных эквивалентов тех наборов, на которых она принимает значение 1. Для нашего примера получим:

$$f(a,b,c) = \{3,\ 5,\ 6\}.$$

4. *Аналитический способ задания.*

При данном способе логическая функция задаётся в виде аналитического выражения, включающего логические переменные, которые связаны стандартными логическими операциями.

Для нашего примера логическая функция может быть задана следующим выражением:

$$f(a,b,c) = a \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot c\ ,$$

где: $+$ - знак операции логического сложения (дизъюнкции);

$\cdot$ - знак операции логического умножения (конъюнкции).

Так как замена значения логической функции хотя бы на одном любом наборе приводит к получению новой логической функции, то возможное количество логических функций от n входных переменных:

$$M = 2^{2^n} = 2^N.$$

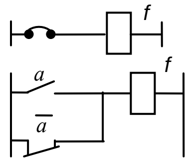
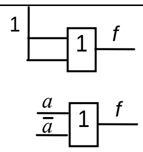
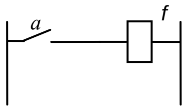
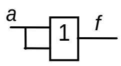
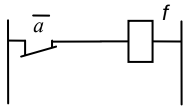
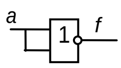
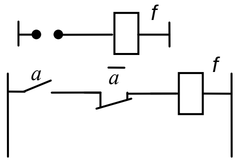
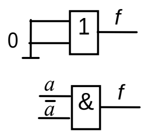
С увеличением количества входных переменных резко возрастает количество функций, поэтому для изучения, анализа и синтеза функций многих переменных используются функции одной и двух переменных и принцип суперпозиции:

$$f(a; b; c; d) = f\,(U; V)$$

$$U = \varphi(a; b) \quad V = \varphi(c; d).$$

1.3 Логические функции одной переменной

Функции одной переменной можно представить в виде следующей таблицы:

Функция	Таблица истинности		Формулы	Содержание логической функции	Контактная схема	Бесконтактная схема
	а 1	0				
Единичная	1	1	f = 1 f = a + $\bar{a}$	Имеет значение 1 независимо от значения а		
Повторение	1	0	f = a	Функция повторяет значение аргумента		
Инверсия, отрицание, дополнение	0	1	f = $\bar{a}$	Функция имеет значение, обратное значению аргумента		
Нулевая	0	0	f = 0 f = a*$\bar{a}$	Имеет значение 0, независимо от значения а		

1.4 Логические функции двух переменных.

Общее количество логических функций двух переменных составляет:

$$M = 2^{2^2} = 16 \, .$$

Рассмотрим логические функции двух переменных.

1. Нулевая.

Таблица истинности

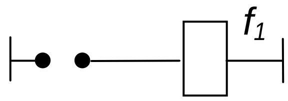
a	b	x
0	0	0
1	0	0
0	1	0
1	1	0

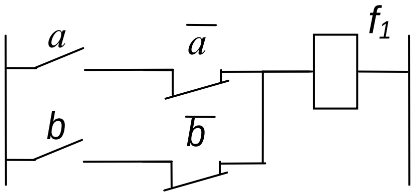
Символическое обозначение: $f = 0$

Содержание логической функции: Функция имеет значение ноль, независимо от значения аргументов функции.

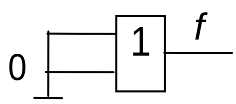
Структурная формула: $f_1 = 0 = a \cdot \bar{a} + b \cdot \bar{b}$.

Контактная схема:





Бесконтактная схема:



2. Операция Пирса (стрелка Пирса, операция ИЛИ-НЕ).

Таблица истинности

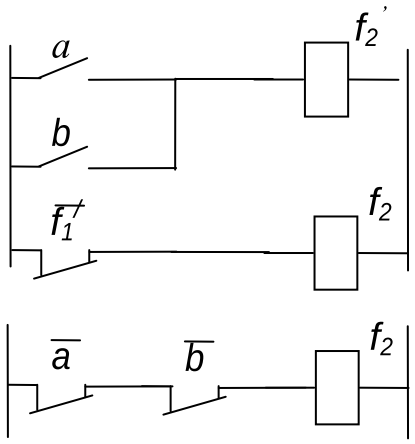
a	b	x
0	0	1
1	0	0
0	1	0
1	1	0

Символическое обозначение: $f = a \downarrow b$

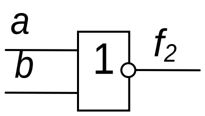
Содержание логической функции: Функция имеет значение единица, когда и переменная a и переменная b имеют значение ноль.

Структурная формула: $f_2 = \overline{a+b} = \bar{a} \cdot \bar{b}$.

Контактная схема:



Бесконтактная схема: логический элемент ИЛИ-НЕ.



3. Запрет *b*.

Таблица истинности

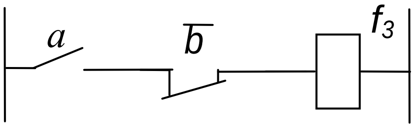
a	*b*	*x*
0	0	0
1	0	1
0	1	0
1	1	0

Символическое обозначение: *f* = a←b

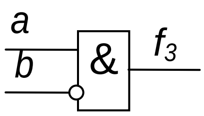
Содержание логической функции: Функция имеет значение, совпадающее со значением *a*, если переменная *b* имеет значение "0" и функция равна нулю f=0 если b=1.

Структурная формула: $f_3 = a \cdot \bar{b}$.

Контактная схема:



Бесконтактная схема:



4. Запрет *a*.

Таблица истинности

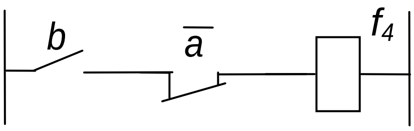
a	*b*	*x*
0	0	0
1	0	0
0	1	1
1	1	0

Символическое обозначение: $f = b \leftarrow a$

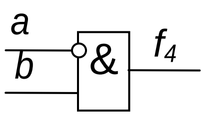
Содержание логической функции: Функция имеет значение, совпадающее со значением *b*, если переменная *a* имеет значение "0" и функция равна нулю f = 0 если a = 1.

Структурная формула: $f_4 = b \cdot \bar{a}$.

Контактная схема:



Бесконтактная схема:



5. Конъюнкция, логическое произведение.

Таблица истинности

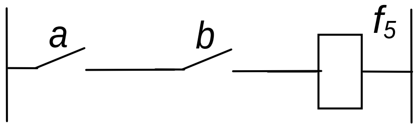
a	*b*	*x*
0	0	0
1	0	0
0	1	0
1	1	1

Символическое обозначение: $f = b \cdot a$

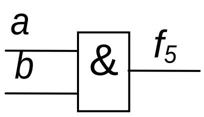
Содержание логической функции: Функция имеет значение единица когда и *a* и *b* имеют значения единица.

Структурная формула: $f_5 = a \cdot b$.

Контактная схема:



Бесконтактная схема: Логический элемент И.



6. Повторение *a*.

Таблица истинности

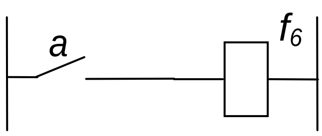
a	b	x
0	0	0
1	0	1
0	1	0
1	1	1

Символическое обозначение: $f = a$

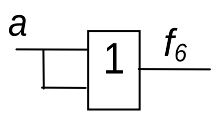
Содержание логической функции: Функция повторяет значение *a*, независимо от значения *b*.

Структурная формула: $f_6 = a \cdot (b + \bar{b}) = a$.

Контактная схема:



Бесконтактная схема:



7. Повторение *b*.

Таблица истинности

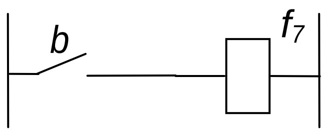
a	*b*	*x*
0	0	0
1	0	0
0	1	1
1	1	1

Символическое обозначение: $f = b$

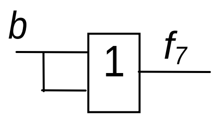
Содержание логической функции: Функция повторяет значение *b*, независимо от значения *a*.

Структурная формула: $f_7 = b \cdot (a + \bar{a}) = b$.

Контактная схема:



Бесконтактная схема:



8. **Инверсия *a***.

Таблица истинности

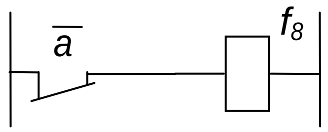
a	*b*	*x*
0	0	1
1	0	0
0	1	1
1	1	0

Символическое обозначение: $f = \bar{a}$

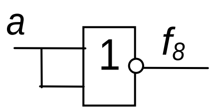
Содержание логической функции: Функция принимает значение, обратное значению *a*, вне зависимости от значения *b*.

Структурная формула: $f_8 = \bar{a} \cdot (b + \bar{b}) = \bar{a}$.

Контактная схема:



Бесконтактная схема:



9. Инверсия *b*.

Таблица истинности

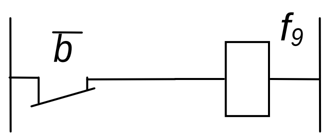
a	b	x
0	0	1
1	0	1
0	1	0
1	1	0

Символическое обозначение: f = b

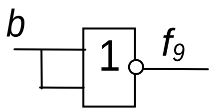
Содержание логической функции: Функция принимает значение, обратное значению *b*, вне зависимости от значения *a*.

Структурная формула: $f_9 = \bar{b} \cdot (a + \bar{a}) = \bar{b}$.

Контактная схема:



Бесконтактная схема:



10. Эквивалентность (равнозначность).

Таблица истинности

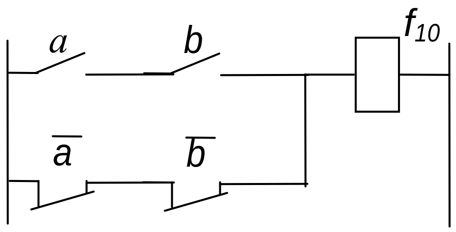
a	*b*	*x*
0	0	1
1	0	0
0	1	0
1	1	1

Символическое обозначение: f = a~b, f = a≡b

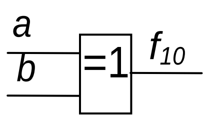
Содержание логической функции: Функция имеет значение единица, когда обе переменные имеют одинаковые значения.

Структурная формула: $f_{10} = \bar{a} \cdot \bar{b} + a \cdot b$.

Контактная схема:



Бесконтактная схема: логический элемент эквивалентность.



11. Неэквивалентность (неравнозначность, исключающее ИЛИ).

Таблица истинности

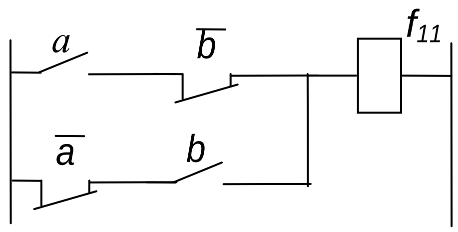
a	*b*	*x*
0	0	0
1	0	1
0	1	1
1	1	0

Символическое обозначение: f = a$\not\equiv$b

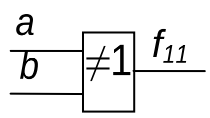
Содержание логической функции: Функция имеет значение единица, когда переменные имеют различные значения.

Структурная формула: $f_{11} = \bar{a} \cdot b + a \cdot \bar{b}$.

Контактная схема:



Бесконтактная схема: логический элемент исключающее ИЛИ.



12. **Операция Шеффера (штрих Шеффера, операция И-НЕ).**

Таблица истинности

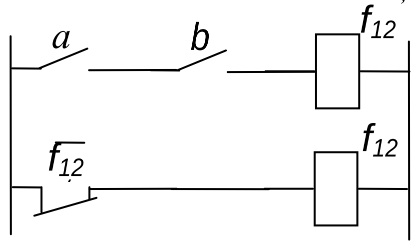
a	b	x
0	0	1
1	0	1
0	1	1
1	1	0

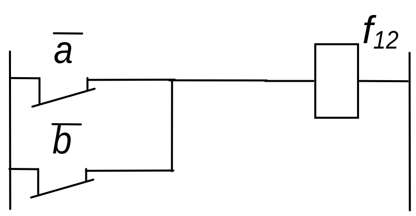
Символическое обозначение: f = a|b

Содержание логической функции: Функция имеет значение ноль, когда и переменная a и переменная b имеют значение единица.

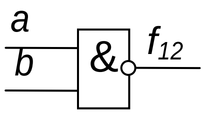
Структурная формула: $f_{12} = \overline{a \cdot b} = \bar{a} + \bar{b}$.

Контактная схема:





Бесконтактная схема: логический элемент И-НЕ.



13. Дизъюнкция (логическое сложение, операция ИЛИ).

Таблица истинности

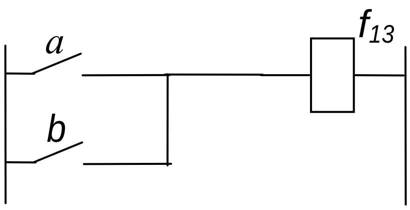
a	*b*	*x*
0	0	0
1	0	1
0	1	1
1	1	1

Символическое обозначение: f = a+b

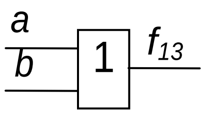
Содержание логической функции: Функция имеет значение ноль, когда и переменная a и переменная b имеют значение ноль.

Структурная формула: $f_{13} = a + b$.

Контактная схема:



Бесконтактная схема: логический элемент ИЛИ.



1.5 **Аксиомы алгебры логики**

Аксиомы алгебры логики являются основными соотношениями, позволяющими преобразовывать логические функции. Все аксиомы алгебры логики можно разделить на две группы:

1. Устанавливающие связь между значениями переменных.
2. Устанавливающие связь между переменными.

Универсальным методом доказательства и проверки логических соотношений является сопоставление значений функций на всех наборах входных переменных.

Логические функции равносильны, если на любом входном наборе их значения совпадают.

Для проверки эквивалентности логических функций необходимо определить их значения на всех входных наборах и сравнить эти значения. Если значения совпадают, то функции будут эквивалентными.

Кроме этого, для доказательства логических выражений могут использоваться равносильные преобразования, позволяющие получить эквивалентные соотношения.

К первой группе относятся следующие аксиомы.

1. Логическая переменная может принимать лишь одно из двух возможных значений:

$$a = 0 \quad \text{если} \quad a \neq 1$$

$$a = 1 \quad \text{если} \quad a \neq 0$$

2. Существуют такие «0» и «1», что

$$\bar{0} = 1; \qquad \bar{1} = 0$$

3. Для основных логических операций соотношения значений переменных имеют следующий вид:

$$0 \cdot 0 = 0 \qquad 1 + 1 = 1$$

$$0 + 0 = 0 \qquad 1 \cdot 1 = 1$$

$$0 \cdot 1 = 0 \qquad 0 + 1 = 1.$$

Как видно, каждая аксиома состоит из двух частей, что соответствует правилу инверсии. Данное правило заключается в том, что любая аксиома может быть преобразована в другую аксиому заменой «0» и «1» и взаимной заменой операций дизъюнкция и конъюнкция.

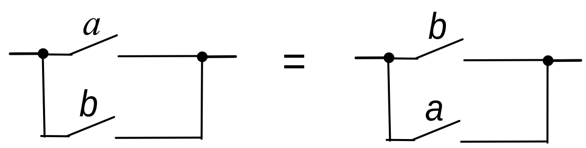
Ко второй группе (устанавливающие связь между переменными) относятся следующие аксиомы.

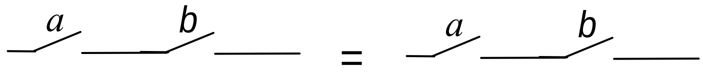
1. Аксиома коммутативности (перемещения)

Результат выполнения операций логического сложения и умножения не зависит от порядка следования переменных.

$$a + b = b + a$$
$$a \cdot b = b \cdot a$$

Для контактной реализации получим





Данная аксиома справедлива практически для всех логических функций двух переменных за исключением импликации и запрета.

2. Аксиома ассоциативности (сочетания)

Результат выполнения операций логического сложения и умножения не зависит от порядка следования скобок в выражении.

$$a + (b + c) = (b + a) + c = a + b + c$$
$$a \cdot (b \cdot c) = (b \cdot a) \cdot c = a \cdot b \cdot c$$

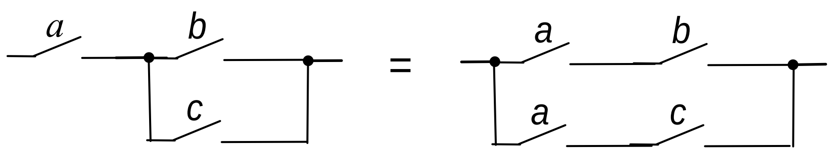
3. Аксиома дистрибутивности

Данная аксиома имеет два варианта.

Первый вариант для операции дизъюнкции.

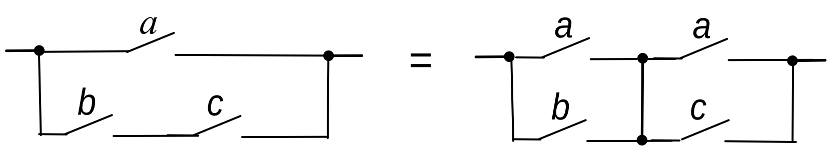
Общая переменная может быть вынесена за скобки.

$$a \cdot (b + c) = a \cdot b + a \cdot c$$



Второй вариант для операции конъюнкция (раскрытие скобок).

$$(a+b)\cdot(a+c)=a+b\cdot c$$



Для проверки аксиомы раскроем скобки и выполним преобразования

$$(a+b)\cdot(a+c)=a\cdot a+a\cdot c+a\cdot b+b\cdot c=a+a\cdot c+a\cdot b+b\cdot c=$$
$$=a(1+c+b)+b\cdot c=a\cdot 1+b\cdot c=a+b\cdot c$$

На основе рассмотренных аксиом выводятся все теоремы, выражающие основные законы алгебры логики. Их ещё называют системой равносильных преобразований функций.

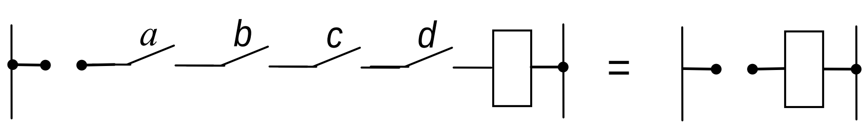
1.6 Законы алгебры логики

I. Закон нулевого множества:

$$0+a=a$$
$$0\cdot a=0$$
$$0\cdot a\cdot b\cdot c\cdot d\cdot e\cdot ....=0$$

Конъюнкция любого числа переменных обращается в 0, если хотя бы одна переменная принимает значение ноль, независимо от значения других переменных.

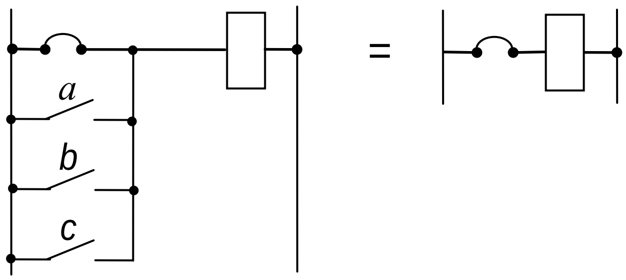
Сущность данного закона поясняет следующая схема.



II. Закон универсального множества:

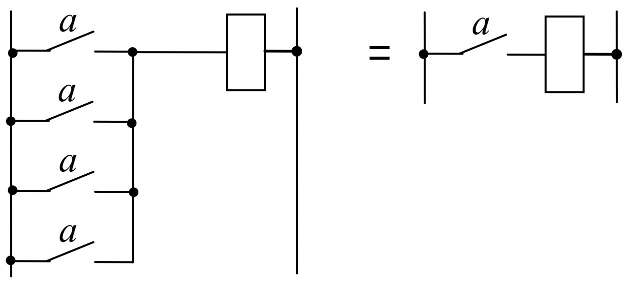
$$1 + a = 1$$
$$1 \cdot a = \grave{a} \quad .$$
$$1 + a + b + c + d + e + \ldots. = 1$$

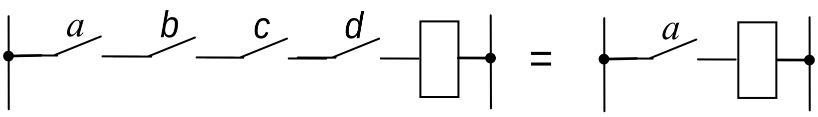
Дизъюнкция любого числа переменных обращается в единицу, если хотя бы одна из переменных принимает значение единица, независимо от значений других переменных.



III. Закон сохранения степени:

$$a + a = a$$
$$a + a + a + a + \ldots = a$$
$$a \cdot a = a$$
$$a \cdot a \cdot a \cdot a \cdot a \cdot \grave{a} \cdot \ldots. = a$$





IV. Закон двойной инверсии:

$$\bar{\bar{\grave{a}}} = \grave{a}\,.$$

V. Законы дополнительности.

– Логического противоречия:

$$\grave{a} \cdot \bar{\grave{a}} = 0\,;$$

конъюнкция любой переменной и её инверсии равна нулю.

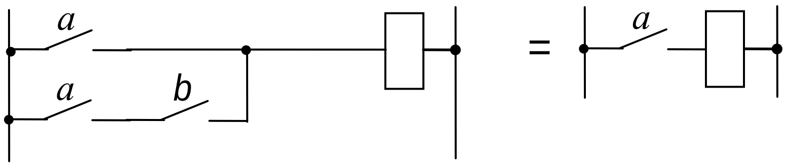
– Исключённого третьего:

$$\grave{a}+\overline{\grave{a}}=1;$$

дизъюнкция любой переменной и её инверсии равна 1.

VI. Законы поглощения.

$$a+a\cdot b=a$$
$$a+a\cdot b+a\cdot c+a\cdot d+\ldots=a\,.$$

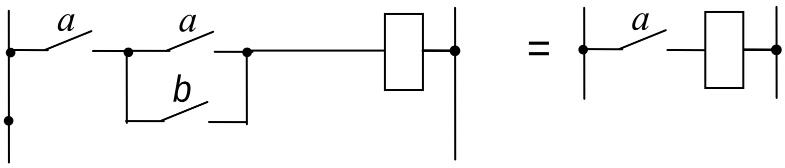


Для доказательства соотношения вынесем переменную *a* за скобки и воспользуемся законом универсального множества.

$$a+a\cdot b=a\cdot(1+b)=a\cdot 1=a\,.$$

Второй вариант закона поглощения:

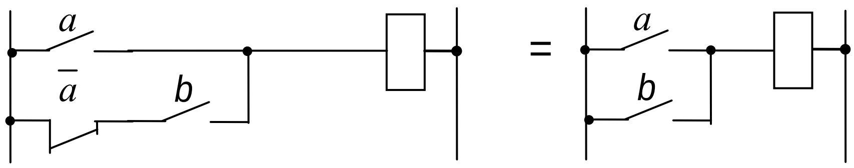
$$a\cdot(\grave{a}+b)=a$$
$$a\cdot(a+b)\cdot(a+c)\cdot\left(a+d\right)+\ldots=a\,.$$



Для доказательства соотношения раскроем скобки, применим закон сохранения степени. После этого вынесем переменную а за скобки и применим закон универсального множества:

$$a\cdot(\grave{a}+b)=a\cdot a+a\cdot b=a+a\cdot b=a\cdot(1+b)=a\cdot 1=a\,.$$

VII. $$a+\bar{a}\cdot b=a+b$$

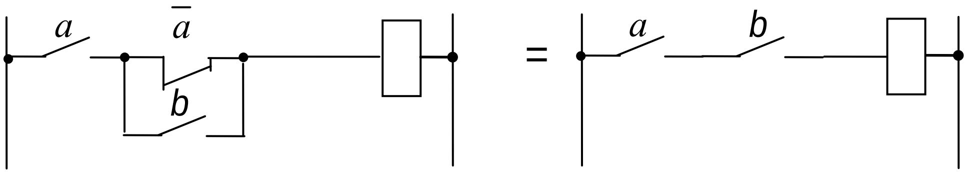


Для доказательства соотношения выполним следующие преобразования:

$$a+\bar{a}\cdot b=a\cdot(b+\bar{b})+\bar{a}\cdot b=a\cdot b+a\cdot\bar{b}+\bar{a}\cdot b=a\cdot b+a\cdot\bar{b}+a\cdot b+\bar{a}\cdot b=$$
$$=a\cdot(b+\bar{b})+b\cdot(a+\bar{a})=a\cdot 1+b\cdot 1=a+b$$

Вторая форма выражения:

$$a\cdot(\bar{a}+b)=a\cdot b$$

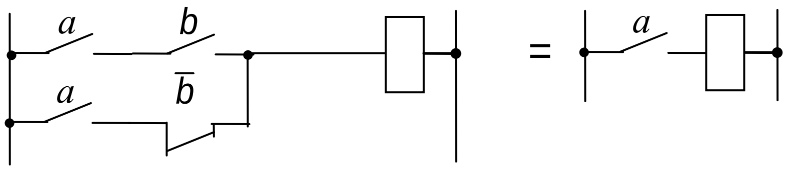


Для доказательства выполним преобразования:

$$a\cdot(\bar{a}+b)=a\cdot\bar{a}+a\cdot b=0+a\cdot b=a\cdot b\,.$$

VIII. Законы склеивания:

$$a\cdot b+a\cdot\bar{b}=a$$

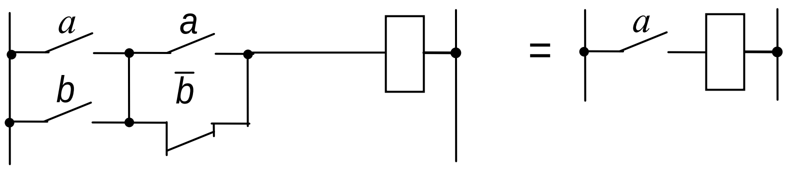


Для доказательства выполним преобразования:

$$a\cdot b+a\cdot\bar{b}=a\cdot(b+\bar{b})=a\cdot 1=a$$

Вторая форма закона:

$$(a+b)\cdot(a+\bar{b})=a\,.$$



Для доказательства раскроем скобки:

$$(a+b)\cdot(a+\bar{b})=a\cdot a+a\cdot\bar{b}+a\cdot b+b\cdot\bar{b}=a\cdot a+a\cdot\bar{b}+a\cdot b+0=$$
$$=a\cdot a+a\cdot\bar{b}+a\cdot b=a\cdot(1+\bar{b}+b)=a\cdot 1=a\,.$$

IX. Закон обобщённого склеивания:

$$a\cdot b+\bar{a}\cdot c+b\cdot c=a\cdot b+\bar{a}\cdot c$$
$$(a+b)\cdot(\bar{a}+c)\cdot(b+c)=(a+b)\cdot(\bar{a}+c)$$

Данные соотношения доказываются через подстановку значений переменных.

X. Законы де Моргана.

$$\overline{a+b} = \bar{a} \cdot \bar{b}$$

$$\overline{a+b+\tilde{n}+d+e+\ldots} = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} \cdot \bar{e} \cdot \ldots$$

т.е. инверсия дизъюнкции есть конъюнкция инверсий.

Второй вариант:

$$\overline{a \cdot b} = \bar{a} + \bar{b}$$

$$\overline{a \cdot b \cdot \tilde{n} \cdot d \cdot e \ldots} = \bar{a} + \bar{b} + \bar{c} + \bar{d} + \bar{e} + \ldots$$

т.е. инверсия конъюнкции есть дизъюнкция инверсий.

Обобщение законов де Моргана вывел американский инженер Шеннон:

$$\overline{f(a,b,c,d,e,\ldots,\times,+)} = f(\bar{a},\bar{b},\bar{c},\bar{d},\bar{e},\ldots,+,\times).$$

Инверсия любой функции получается заменой каждой переменной её инверсией и одновременно взаимной заменой знаков операций конъюнкция и дизъюнкция.

XI. Теорема разложения.

Можно показать, что любую логическую функцию можно представить в виде:

$$\begin{aligned} f(a,b,c,d\ldots) &= a \cdot f(1,b,c,d\ldots) + \bar{a} \cdot f(0,b,c,d\ldots) \\ f(a,b,c,d\ldots) &= [a + f(0,b,c,d\ldots)] \cdot [\bar{a} + f(1,b,c,d\ldots)] \end{aligned}.$$

Для доказательства данной теоремы воспользуемся методом подстановки. Примем $a = 1$, следовательно $\bar{\grave{a}} = 0$, и подставим в выражения:

$$\begin{aligned} f(1,b,c,d\ldots) &= 1 \cdot f(1,b,c,d\ldots) + 0 \cdot f(0,b,c,d\ldots) = 1 \cdot f(1,b,c,d\ldots) + 0 = \\ &= 1 \cdot f(1,b,c,d\ldots) = f(1,b,c,d\ldots) \end{aligned}.$$

Для второго выражения получим:

$$\begin{aligned} f(1,b,c,d\ldots) &= [1 + f(0,b,c,d\ldots)] \cdot [0 + f(1,b,c,d\ldots)] = 1 \cdot [0 + f(1,b,c,d\ldots)] = \\ &= [0 + f(1,b,c,d\ldots)] = f(1,b,c,d\ldots) \end{aligned}.$$

Полученные равносильные выражение показывают, что теорема верна.

1.7 Элементарная конъюнкция и дизъюнкция. Нормальные формы функций

При проектировании дискретных систем управления для описания принципа их работы используется математический аппарат Булевой алгебры.

Булева алгебра – это алгебра, содержащая две бинарных операции (конъюнкция и дизъюнкция), и одну унарную операцию (инверсия). При этом переменные могут принимать только два значения.

Элементарной конъюнкцией называется конъюнкция <u>любого</u> числа различных переменных функции, входящих с инверсией или без не более одного раза.

Пример: если имеется логическая функция четырёх переменных *f(a,b,c,d)*, то для неё элементарные конъюнкции могут иметь вид:

$$a \cdot b \cdot c \cdot \overline{d};\ \ a \cdot \overline{b} \cdot c;\ \ \overline{a} \cdot c;\ \ b.$$

Как видно из примера, элементарная конъюнкция может содержать четыре переменные функции, три, две или одну переменную.

Элементарной дизъюнкцией называется дизъюнкция <u>любого</u> числа различных переменных функции, входящих с инверсией или без не более одного раза.

Пример: для логической функции четырёх переменных *f(a,b,c,d)* элементарные дизъюнкции могут иметь вид:

$$a + b + c + \overline{d};\ \ a + \overline{b} + c;\ \ \overline{a} + c;\ \ b.$$

Для записи логических выражений в Булевой алгебре наиболее часто используется две стандартных формы записи, которые называются нормальными формами.

Дизъюнктивной нормальной формой функции (ДНФ) называется дизъюнкция любого числа элементарных конъюнкций.

$$f(a,b,c,d)_{\text{ÄÍÔ}} = a \cdot b \cdot c \cdot \overline{d} + a \cdot \overline{b} \cdot c + \overline{a} \cdot c + b$$

Конъюнктивной нормальной формой функции (КНФ) называется конъюнкция любого числа элементарных дизъюнкций.

$$f(a,b,c,d) = (a + b + c + \bar{d}) \cdot (a + \bar{b} + c) \cdot (\bar{a} + c) \cdot b.$$

Любая логическая функция может быть записана в виде равносильных логических уравнений с использование различных логических функций двух переменных. От любой формы записи можно перейти к ДНФ или КНФ.

Для того чтобы записать произвольно заданную функцию в виде дизъюнктивной нормальной формы (ДНФ) необходимо:

1. Выразить все операции через операции дизъюнкция, конъюнкция и инверсия.
2. Избавиться от инверсии над целыми выражениями, использую закон де Моргана.
3. Раскрыть скобки, используя закон дистрибутивности:

$$(a + b) \cdot (a + c) = a + b \cdot c.$$

4. Применить законы поглощения.

Пример: преобразовать логическую функцию, заданную в произвольной форме, в форму ДНФ. Логическая функция имеет следующий вид:

$$f(a,b,c) = [a \to \overline{(b \cdot \bar{c} \sim \bar{a})}] + b \cdot a.$$

Раскрываем операцию эквивалентности ($a \sim b = \bar{a} \cdot \bar{b} + a \cdot b$):

$$f(a,b,c) = [a \to \overline{(b \cdot \bar{c} \sim \bar{a})}] + b \cdot a = [a \to \overline{(\overline{b \cdot \bar{c}} \cdot \bar{\bar{a}} + b \cdot \bar{c} \cdot \bar{a})}] + b \cdot a.$$

Избавляемся от общей инверсии с помощью закона де Моргана:

$$\begin{aligned} f(a,b,c) &= [a \to \overline{(\overline{b \cdot \bar{c}} \cdot \bar{\bar{a}} + b \cdot \bar{c} \cdot \bar{a})}] + b \cdot a = [a \to (\overline{\overline{b \cdot \bar{c}} \cdot a}) \cdot (\overline{b \cdot \bar{c} \cdot \bar{a}})] + b \cdot a = \\ &= [a \to (b \cdot \bar{c} + \bar{a}) \cdot (\bar{b} + c + a)] + b \cdot a \end{aligned}.$$

Раскрываем скобки и выполняем преобразования:

$$\begin{aligned} f(a,b,c) &= [a \to (b \cdot \bar{c} + \bar{a}) \cdot (\bar{b} + c + a)] + b \cdot a = [a \to (\underbrace{b \cdot \bar{c} \cdot \bar{b}}_{=0} + \underbrace{b \cdot \bar{c} \cdot c}_{=0} + b \cdot \bar{c} \cdot a + \\ &+ \bar{a} \cdot \bar{b} + \bar{a} \cdot c + \underbrace{\bar{a} \cdot a}_{=0})] + b \cdot a = [a \to (b \cdot \bar{c} \cdot a + \bar{a} \cdot \bar{b} + \bar{a} \cdot c)] + b \cdot a \end{aligned}$$

Раскрываем операцию импликации ($a \to b = \bar{a} + b$):

$$f(a,b,c)=[a\rightarrow(b\cdot\overline{c}\cdot a+\overline{a}\cdot\overline{b}+\overline{a}\cdot c)]+b\cdot a=\overline{a}+b\cdot\overline{c}\cdot a+\overline{a}\cdot\overline{b}+\overline{a}\cdot c+b\cdot a=$$
$$=\overline{a}\cdot\underbrace{(1+\overline{b}+c)}_{=1}+b\cdot a\cdot\underbrace{(\overline{c}+1)}_{=1}=\overline{a}\cdot 1+b\cdot a\cdot 1=\overline{a}+b\cdot a=\overline{a}+b$$

Для того чтобы записать произвольно заданную функцию в виде конъюнктивной нормальной формы (КНФ) необходимо:

5. Выразить все операции через операции дизъюнкция, конъюнкция и инверсия.
6. Избавиться от инверсии над целыми выражениями, использую закон де Моргана.
7. Используя закон дистрибутивности получить конъюнкцию дизъюнкций:

$$a+b\cdot c=(a+b)\cdot(a+c).$$

8. Применить законы поглощения.

1.8 Совершенные нормальные формы логических функций

Для записи логических выражений по таблицам истинности используются совершенные нормальные формы функций.

Конституентом единицы (минтермом) называется конъюнкция всех переменных функции, взятых с инверсией или без не более одного раза.

Пример: если имеется логическая функция четырёх переменных *f(a,b,c,d)*, то для неё конституенты единицы могут иметь вид:

$$a\cdot b\cdot c\cdot\overline{d};\ a\cdot\overline{b}\cdot c\cdot d;\ \overline{a}\cdot\overline{b}\cdot c\cdot d....$$

Конституентом нуля (макстермом) называется дизъюнкция всех переменных функции, взятых с инверсией или без не более одного раза.

Пример: для логической функции четырёх переменных *f(a,b,c,d)* конституенты нуля могут иметь вид:

$$a+b+c+\overline{d};\ a+\overline{b}+c+d;\ \overline{a}+c+b+d.....$$

Конституенты нуля и единицы могут быть составлены для каждого входного набора функции.

Конституент единицы (минтерм) составляется как конъюнкция всех переменных функции, причём если в данном наборе переменная имеет значения единицы, то в минтерм входит сама переменная, если переменная имеет значение ноль, то в минтерм входит инверсия переменной.

Конституент нуля (макстерм) составляется как дизъюнкция всех переменных функции, причём если в данном наборе переменная имеет значения единицы, то в макстерм входит инверсия переменной, если переменная имеет значение ноль, то в макстерм входит сама переменная.

Пример: для функции трёх переменных $f(a,b,c)$ конституеты нуля и единицы для всех наборов будут иметь следующий вид:

Входные наборы			*Минтерм*	*Макстерм*
a	b	c		
0	0	0	$\overline{a}\cdot\overline{b}\cdot\overline{c}$	$a+b+c$
0	0	1	$\overline{a}\cdot\overline{b}\cdot c$	$a+b+\overline{c}$
0	1	0	$\overline{a}\cdot b\cdot\overline{c}$	$a+\overline{b}+c$
0	1	1	$\overline{a}\cdot b\cdot c$	$a+\overline{b}+\overline{c}$
1	0	0	$a\cdot\overline{b}\cdot\overline{c}$	$\overline{a}+b+c$
1	0	1	$a\cdot\overline{b}\cdot c$	$\overline{a}+b+\overline{c}$
1	1	0	$a\cdot b\cdot\overline{c}$	$\overline{a}+\overline{b}+c$
1	1	1	$a\cdot b\cdot c$	$\overline{a}+\overline{b}+\overline{c}$

Из примера видно, что если в минтерм подставить значения соответствующего входного набора, то он будет равен 1. Аналогично, если в макстерм подставить значения соответствующего входного набора, то он будет равен 0. Поэтому они называются конституентами единицы и нуля.

Существует один вид дизъюнктивной нормальной формы и один вид конъюнктивной нормальной формы, в которых функция может быть записана только единственным образом. Они называются *совершенными нормальными формами* логических функций.

Совершенная дизъюнктивная нормальная форма (СДНФ) функции представляет собой дизъюнкцию конституентов единицы для тех наборов, на которых функция принимает значения единицы.

Совершенная конъюнктивная нормальная форма (СКНФ) функции представляет собой конъюнкцию конституентов нуля для тех наборов, на которых функция принимает значения ноль.

Рассмотрим примеры записи нормальных форм функций по заданным таблицам истинности.

Пример: по заданной таблице истинности записать аналитические выражения функции трёх переменных $f(a,b,c)$ в виде СДНФ и СКНФ. Таблица истинности функции имеет следующий вид:

Номер набора	a	b	c	x
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	0

Для записи функции в виде СДНФ необходимо записать дизъюнкцию конституентов единицы для тех наборов, на которых функция принимает значения 1. Функция принимает значение 1 на 2, 4, 6 и 7 наборах. Конституенты единицы для данных наборов имеют вид:

2 набор - $\bar{a} \cdot \bar{b} \cdot c$;

4 набор - $\bar{a} \cdot b \cdot c$;

6 набор - $a \cdot \bar{b} \cdot c$;

7 набор - $a \cdot b \cdot \bar{c}$.

Совершенная дизъюнктивная нормальная форма функции будет иметь вид:

$$f(a,b,c)_{\text{ÑÄÍÔ}} = \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}.$$

Для записи функции в виде СКНФ необходимо записать конъюнкцию конституентов нуля для тех наборов, на которых функция принимает значения 0. Функция принимает значение 0 на 1, 3, 5 и 8 наборах. Конституенты нуля для данных наборов имеют вид:

1 набор - $a + b + c$;

3 набор - $a + \bar{b} + c$;

5 набор - $\bar{a} + b + c$;

8 набор - $\bar{a} + \bar{b} + \bar{c}$.

Совершенная дизъюнктивная нормальная форма функции будет иметь вид:

$$f(a,b,c)_{\text{ÑÊÍÔ}} = (a + b + c) \cdot (a + \bar{b} + c) \cdot (\bar{a} + b + c) \cdot (\bar{a} + \bar{b} + \bar{c}).$$

Совершенные нормальные формы функций обладают следующими свойствами.

1. СДНФ данной функции содержит столько слагаемых, сколько единиц содержится в таблице истинности.
2. СКНФ данной функции содержит столько множителей, сколько нулей содержится в таблице истинности.
3. Для любой функции сумма членов СДНФ и СКНФ равна числу входных наборов.
4. Существует единственная функция, не имеющая СКНФ - единичная.
5. Существует единственная функция, не имеющая СДНФ - нулевая.
6. Для того чтобы функция в СДНФ имела значение единицы необходимо, чтобы хотя бы один конституент единицы на данном наборе принимал значение единицы.

7. Для того чтобы функция в СКНФ имела значение ноль необходимо, чтобы хотя бы один конституент нуля на данном наборе принимал значение ноль.

В некоторых случаях при проектировании дискретных управляющих устройств возникает необходимость по совершенной форме построить карту Карно. Для этого необходимо по конституентам совершенной формы определить входные наборы и проставить в карте Карно на этих наборах соответствующие значения (для СДНФ -1, для СКНФ – 0).

Пример. По заданной СДНФ функции построить карту Карно.

$$f(a,b,c,d)_{\text{ÑÄÍÔ}} = \bar{a}\cdot\bar{b}\cdot c\cdot d + \bar{a}\cdot b\cdot c\cdot\bar{d} + a\cdot\bar{b}\cdot c\cdot\bar{d} + a\cdot b\cdot\bar{c}\cdot d + a\cdot b\cdot\bar{c}\cdot\bar{d} + a\cdot\bar{b}\cdot\bar{c}\cdot d$$

Так СДНФ функции содержит 6 слагаемых, следовательно в карте Карно будет шесть единиц. По конституентам определим наборы, на которых функция принимает значение 1:

конституент	*a*	*b*	*c*	*d*
$\bar{a}\cdot\bar{b}\cdot c\cdot d$	0	0	1	1
$\bar{a}\cdot b\cdot c\cdot\bar{d}$	0	1	1	0
$a\cdot\bar{b}\cdot c\cdot\bar{d}$	1	0	1	0
$a\cdot b\cdot\bar{c}\cdot d$	1	1	0	1
$a\cdot b\cdot\bar{c}\cdot\bar{d}$	1	1	0	0
$a\cdot\bar{b}\cdot\bar{c}\cdot d$	1	0	0	1

Строим карту Карно и расставляем значения функции 1 на определённых наборах. На остальных наборах расставляем значение 0.

b

a

0	1	0	0
1	1	0	0
0	0	0	1
1	0	1	0

d

c

От записи логической функции в нормальной форме можно перейти к записи в совершенной нормальной форме.

Для перехода от ДНФ к СДНФ необходимо в каждой конъюнкции восстановить недостающую переменную, умножив эту конъюнкцию на 1 = $\grave{a}+\bar{a}$, где a – недостающая переменная. После этого раскрываются скобки и сокращаются повторяющиеся конституенты.

Пример. Преобразовать заданную ДНФ к СДНФ и построить карту Карно.

$$f(a,b,c,d)_{\text{ÄÍÔ}} = a\cdot b\cdot c+\bar{a}\cdot b\cdot d+a\cdot\bar{b}\cdot\bar{c}+\bar{b}\cdot\bar{c}\cdot d\,.$$

Восстанавливаем в слагаемых недостающие переменные.

$$f(a,b,c,d)_{\text{ÄÍÔ}} = a\cdot b\cdot c\cdot(d+\bar{d})+\bar{a}\cdot b\cdot d\cdot(c+\bar{c})+a\cdot\bar{b}\cdot\bar{c}\cdot(d+\bar{d})+\bar{b}\cdot\bar{c}\cdot d\cdot(a+\bar{a})$$

Раскрываем скобки и сокращаем повторяющиеся конституенты.

$$\begin{aligned}f(a,b,c,d)_{\text{ÄÍÔ}} &= a\cdot b\cdot c\cdot(d+\bar{d})+\bar{a}\cdot b\cdot d\cdot(c+\bar{c})+a\cdot\bar{b}\cdot\bar{c}\cdot(d+\bar{d})+\bar{b}\cdot\bar{c}\cdot d\cdot(a+\bar{a})=\\&=a\cdot b\cdot c\cdot d+a\cdot b\cdot c\cdot\bar{d}+\bar{a}\cdot b\cdot d\cdot c+\bar{a}\cdot b\cdot d\cdot\bar{c}+\underline{a\cdot\bar{b}\cdot\bar{c}\cdot d}+a\cdot\bar{b}\cdot\bar{c}\cdot\bar{d}+\underline{\bar{b}\cdot\bar{c}\cdot d\cdot a}+\\&+\bar{b}\cdot\bar{c}\cdot d\cdot\bar{a}=a\cdot b\cdot c\cdot d+a\cdot b\cdot c\cdot\bar{d}+\bar{a}\cdot b\cdot d\cdot c+\bar{a}\cdot b\cdot d\cdot\bar{c}+a\cdot\bar{b}\cdot\bar{c}\cdot d+a\cdot\bar{b}\cdot\bar{c}\cdot\bar{d}+\\&+\bar{b}\cdot\bar{c}\cdot d\cdot\bar{a}\end{aligned}.$$

По полученной СДНФ строим карту Карно.

b

a

1	0	0	0
1	0	1	1
0	1	1	0
0	1	0	0

d

c

Для того чтобы от КНФ функции перейти к СКНФ необходимо в дизъюнкции, в которых не хватает переменных прибавить ноль, выразив его через закон логического противоречия ($\grave{a}\cdot\bar{a}=0$, где a – недостающая переменная). После этого необходимо преобразовать полученную дизъюнкцию, пользуясь законом дистрибутивности ($(a+b)\cdot(a+c)=a+b\cdot c$) и привести подобные члены.

Пример. Преобразовать заданную КНФ к СКНФ и построить карту Карно.

$$f(a,b,c,d)_{\text{ÊÍÔ}} = (a+b+c)\cdot(a+\overline{b}+d)\cdot(a+c+\overline{d})$$

Восстанавливаем в дизъюнкциях недостающие переменные:

$$f(a,b,c,d)_{\text{ÊÍÔ}} = (a+b+c+d\cdot\overline{d})\cdot(a+\overline{b}+d+c\cdot\overline{c})\cdot(a+c+\overline{d}+b\cdot\overline{b}).$$

Применяем закон дистрибутивности и приводим подобные члены:

$$f(a,b,c,d)_{\text{ÑÊÍÔ}} = (a+b+c+d)\cdot(\underline{a+b+c+\overline{d}})\cdot(a+\overline{b}+d+c)\cdot(a+\overline{b}+d+\overline{c})\cdot$$
$$\cdot(\underline{a+c+\overline{d}+b})\cdot(a+c+\overline{d}+\overline{b}) = (a+b+c+d)\cdot(a+b+c+\overline{d})\cdot(a+\overline{b}+d+c)\cdot\ .$$
$$\cdot(a+\overline{b}+d+\overline{c})\cdot(a+c+\overline{d}+\overline{b})$$

Строим карту Карно.

b

a

1	1	0	0
1	1	0	0
1	1	1	1
1	1	0	1

d

c

1.9 **Применение теоремы разложения к нормальным формам функции**

Первое выражение для теоремы разложения имеет вид:

$$f(a,b,c,d\ldots) = a\cdot f(1,b,c,d\ldots) + \overline{a}\cdot f(0,b,c,d\ldots).$$

Пусть имеется функция, имеющая вид $a\cdot f(a,b,c,d\ldots)$. Применим к ней теорему разложения:

$$a\cdot f(a,b,c,d...) = a\cdot[a\cdot f(1,b,c,d...) + \bar{a}\cdot f(0,b,c,d...)] =$$
$$\underbrace{a\cdot a}_{=a}\cdot f(1,b,c,d...) + \underbrace{a\cdot \bar{a}}_{=0}\cdot f(0,b,c,d...) = a\cdot f(1,b,c,d...) + 0\cdot f(0,b,c,d...) = .$$
$$= a\cdot f(1,b,c,d...) + 0 = a\cdot f(1,b,c,d...)$$

Таким образом, получаем соотношение:

$$a\cdot f(a,b,c,d...) = a\cdot f(1,b,c,d...).$$

Данное соотношение наиболее часто применяется для упрощения релейно-контакторных схем.

Если в релейно-контакторной схеме имеется отдельный, последовательно соединённый с остальной частью схемы контакт, то в остальной части схемы все одноимённые контакты можно закоротить, а инверсные контакты заменить разрывом цепи.

Пример. Упростить релейно-контакторную схему, используя теорему разложения. Схема имеет следующий вид.

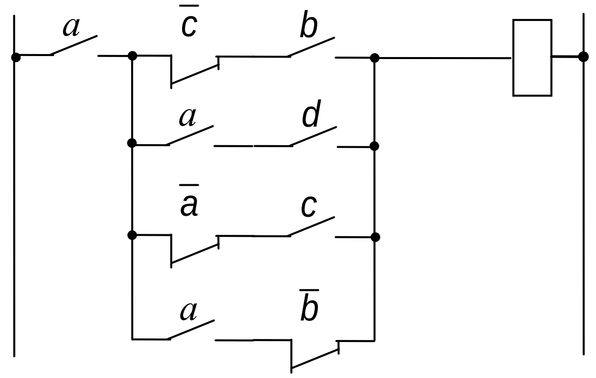


Рисунок 1.2 – Релейно-контакторная схема управляющего устройства

По схеме запишем аналитические выражения функции, учитывая, что последовательное соединение контактов соответствует операции конъюнкция, а параллельное соединение контактов – операции дизъюнкции.

$$f(a,b,c,d) = a\cdot(b\cdot\bar{c} + a\cdot d + \bar{a}\cdot c + a\cdot\bar{b}) = a\cdot f'(a,b,c,d).$$

Применим к данному выражению полученное соотношение:

$$a\cdot f'(a,b,c,d) = a\cdot f'(1,b,c,d) = a\cdot(b\cdot\bar{c} + 1\cdot d + 0\cdot c + 1\cdot\bar{b}) =$$
$$= a\cdot(b\cdot\bar{c} + d + 0 + \bar{b}) = a\cdot(b\cdot\bar{c} + d + \bar{b}).$$

На основании полученного выражения строим упрощённую схему.

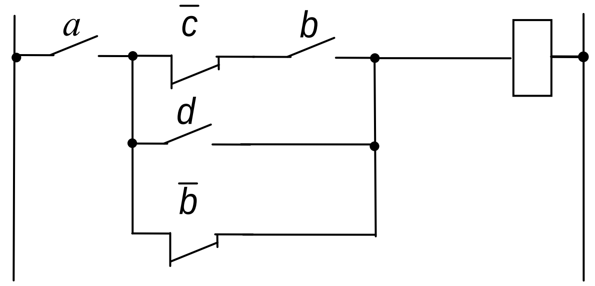


Рисунок 1.3 – Преобразованная релейно-контакторная схема управляющего устройства

Рассмотрим второе выражение теоремы разложения. Пусть имеется логическая функция вида $a+f(a,b,c,d...)$. Применив второе выражение теоремы разложения, получим:

$$a+f(a,b,c,d...)=a+[a+f(0,b,c,d...)]\cdot[\bar{a}+f(1,b,c,d...)]=$$
$$=[a+a+f(0,b,c,d...)]\cdot[\underbrace{a+\bar{a}}_{=1}+f(1,b,c,d...)]=[a+f(0,b,c,d...)]\cdot$$
$$\cdot[\underbrace{1+f(1,b,c,d...)}_{=1}]=[a+f(0,b,c,d...)]\cdot 1=a+f(0,b,c,d...)$$

Таким образом, получаем соотношение:

$$a+f(a,b,c,d...)=a+f(0,b,c,d...).$$

Если в релейно-контакторной схеме имеется отдельный, параллельно соединённый с остальной частью схемы контакт, то в остальной части схемы все одноимённые контакты можно заменить разрывом цепи, а инверсные контакты закоротить.

Пример. Упростить релейно-контакторную схему, используя второе выражение теоремы разложения. Схема имеет следующий вид.

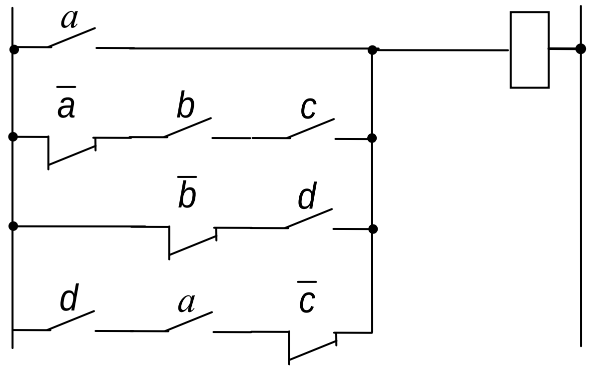


Рисунок 1.4 – Релейно-контакторная схема управляющего устройства

Запишем аналитическое выражение:

$$f(a,b,c,d) = a + \bar{a} \cdot b \cdot c + \bar{b} \cdot d + a \cdot d \cdot \bar{c} = a + f'(a,b,c,d...)$$

Преобразуем полученное выражение

$$a + f'(a,b,c,d...) = a + f'(0,b,c,d...) = a + 1 \cdot b \cdot c + \bar{b} \cdot d + 0 \cdot d \cdot \bar{c} = \\ = a + b \cdot c + \bar{b} \cdot d + 0 = a + b \cdot c + \bar{b} \cdot d.$$

На основании полученного выражения строим упрощённую схему.

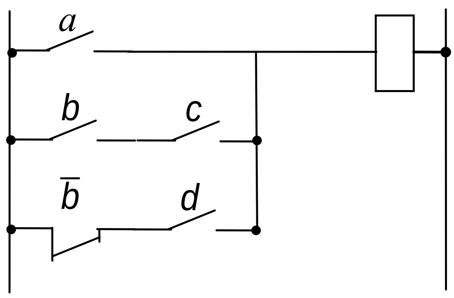


Рисунок 1.5 – Преобразованная релейно-контакторная схема управляющего устройства

Преобразования схем можно выполнять, не записывая аналитические выражения. Для этого необходимо закоротить или заменить разрывом цепи соответствующие контакты.

1.10 **Базис функции Шеффера**

Развитие технологии изготовления элементов дискретной автоматики привело к созданию новых типов элементов и поставило задачи синтеза автоматов с учётом особенностей этих элементов. В настоящее время промышленностью выпускается много типов элементов в интегральном исполнении, наибольшее распространение среди которых получили элементы, реализующие функцию Шеффера (И-НЕ).

При реализации устройства на элементах одного типа необходимо записать логические выражения, описывающие принцип действия данного устройства, в базисе соответствующей операции.

Базисом называется запись логического выражения с использованием ограниченного количества логических операций (обычно одной).

При переходе к базису операции Шеффера для записи выражения используется только операция Шеффера:

$$f = a \mid b = \overline{a \cdot b}.$$

Таблица истинности данной операции имеет следующий вид:

a	b	f
0	0	1
1	0	1
0	1	1
1	1	0

Рассмотрим основные свойства этой функции.

1. Связь с переменными и константами:

$$\begin{array}{ll} a \mid a = \overline{a} & a \mid \overline{a} = 1 \\ a \mid 1 = \overline{a} & a \mid 0 = 1 \end{array}.$$

2. Связь с дизъюнкцией:

$$a + b = \overline{\overline{a + b}} = \overline{\overline{a} \cdot \overline{b}} = \overline{a} \mid \overline{b} = (a \mid a) \mid (b \mid b) = (a \mid 1) \mid (b \mid 1).$$

3. Связь с конъюнкцией:

$$a \cdot b = \overline{\overline{a \cdot b}} = \overline{a \mid b} = (a \mid b) \mid (a \mid b) = (a \mid b) \mid 1.$$

В алгебре, построенной на функции Шеффера справедлив только переместительный закон: a | b | c = b | c | a.

Преобразование логического выражения в базис Шеффера в общем виде происходит с использованием теоремы разложения:

$$f(a,b,c)=\overline{\overline{f(a,b,c)}}=\overline{\overline{a\cdot f(1,b,c)+\bar{a}\cdot f(0,b,c)}}=\overline{\overline{a\cdot f(1,b,c)}\cdot\overline{\bar{a}\cdot f(0,b,c)}}=$$
$$=[a\mid f(1,b,c)]\mid[\bar{a}\mid f(0,b,c)].$$

Таким образом, окончательно можем записать:

$$f(a,b,c)=[a\mid f(1,b,c)]\mid[\bar{a}\mid f(0,b,c)].$$

Рассмотрим несколько примеров по преобразованию логических функций и реализации принципиальных схем в базисе операции Шеффера.

Пример 1. Преобразовать заданную функцию в базис Шеффера.

$$f(a,b,c)=a\cdot\bar{b}+\bar{a}\cdot c+b\cdot\bar{c}+a\cdot b\cdot c$$

Для преобразования возьмём двойную инверсию и преобразуем с помощью закона де-Моргана.

$$\overline{\overline{f(a,b,c)}}=\overline{\overline{a\cdot\bar{b}+\bar{a}\cdot c+b\cdot\bar{c}+a\cdot b\cdot c}}=\overline{\overline{a\cdot\bar{b}}*\overline{\bar{a}\cdot c}*\overline{b\cdot\bar{c}}*\overline{a\cdot b\cdot c}}=$$
$$=(a\mid\bar{b})\mid(\bar{a}\mid c)\mid(b\mid\bar{c})\mid(a\mid b\mid c).$$

Из рассмотренного примера можно сделать следующий вывод.

Для того чтобы перейти от функции, заданной в ДНФ, в базис Шеффера, необходимо в исходной функции заключить в скобки все конъюнкции и заменить все операции дизъюнкция и конъюнкция на операцию Шеффера.

Пример 2. Преобразовать функцию

$$f(a,b,c)=\overline{\bar{a}\cdot b+\bar{b}\cdot c+a\cdot b\cdot c+\bar{a}\cdot\bar{c}}$$

в базис Шеффера и построить функциональную схему.

При преобразовании воспользуемся предыдущим правилом:

$$f(a,b,c)=\overline{\bar{a}\cdot b+\bar{b}\cdot c+a\cdot b\cdot c+\bar{a}\cdot\bar{c}}=\overline{(\bar{a}\mid b)\mid(\bar{b}\cdot c)\mid(a\mid b\mid c)\mid(\bar{a}\mid\bar{c})}=$$
$$=[(\bar{a}\mid b)\mid(\bar{b}\cdot c)\mid(a\mid b\mid c)\mid(\bar{a}\mid\bar{c})]\mid 1=[(\bar{a}\mid b)\mid(\bar{b}\cdot c)\mid(a\mid b\mid c)\mid(\bar{a}\mid\bar{c})]\mid$$
$$\mid[(\bar{a}\mid b)\mid(\bar{b}\cdot c)\mid(a\mid b\mid c)\mid(\bar{a}\mid\bar{c})].$$

На основании полученного выражения можем построить функциональную схему.

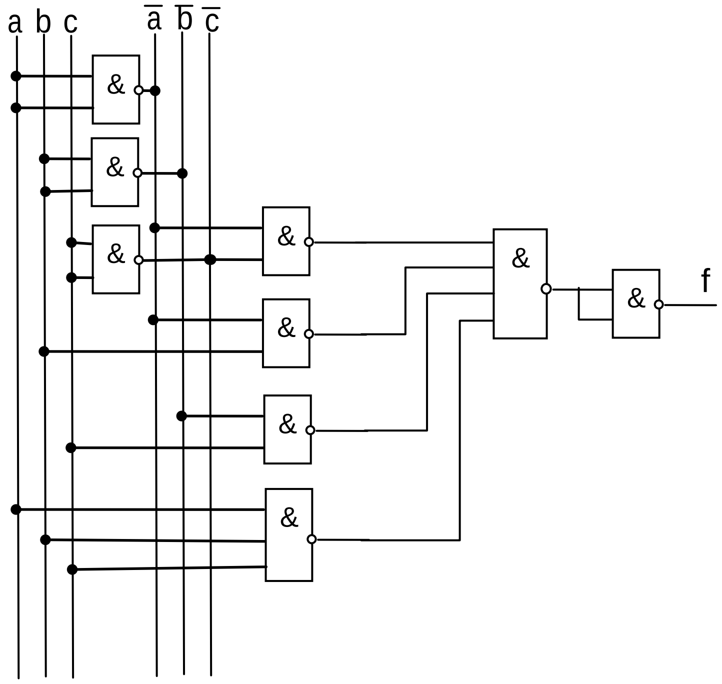


Рисунок 1.6 – Функциональная схема управляющего устройства на бесконтактных элементах И-НЕ

Пример 3. Преобразовать функцию, заданную в КНФ в базис операции Шеффера.

$$f(a,b,c) = (\bar{a}+b)\cdot(b+\bar{c})\cdot(a+c)$$

$$f(a,b,c) = \overline{\overline{(\bar{a}+b)\cdot(b+\bar{c})\cdot(a+c)}} = \overline{\overline{(\bar{a}+b)}+\overline{(b+\bar{c})}+\overline{(a+c)}} = \overline{a\cdot\bar{b}+\bar{b}\cdot c+\bar{a}\cdot\bar{c}} =$$

$$= \overline{(a|\bar{b})|(\bar{b}|c)|(\bar{a}|\bar{c})} = [(a|\bar{b})|(\bar{b}|c)|(\bar{a}|\bar{c})]|[(a|\bar{b})|(\bar{b}|c)|(\bar{a}|\bar{c})]$$

Для того, чтобы перейти от функции, заданной в произвольной форме к функции в базисе операции Шеффера необходимо:

1. Выразить все операции через операции конъюнкция, дизъюнкция и инверсия и избавится при этом от скобок.
2. В получившейся ДНФ заключить все конъюнкции в скобки и заменить знаки операций конъюнкция и дизъюнкция на штрих Шеффера
3. Если исходная функция имела общую инверсию, то эту инверсию можно реализовать, используя выражения:

$$a \mid a = \bar{a} \qquad a \mid 1 = \bar{a}.$$

Логическую операцию И-НЕ реализуют интегральные микросхемы типа ЛА. Каждый из корпусов интегральных схем (ИС) типа ЛА содержит от двух до четырёх логических элементов, а микросхемы ЛА2 и ЛА19 содержат по одному логическому элементу И-НЕ на восемь и на двенадцать входов соответственно.

Условные обозначения и цоколёвка микросхем типа ЛА представлены на рисунке 1.2.

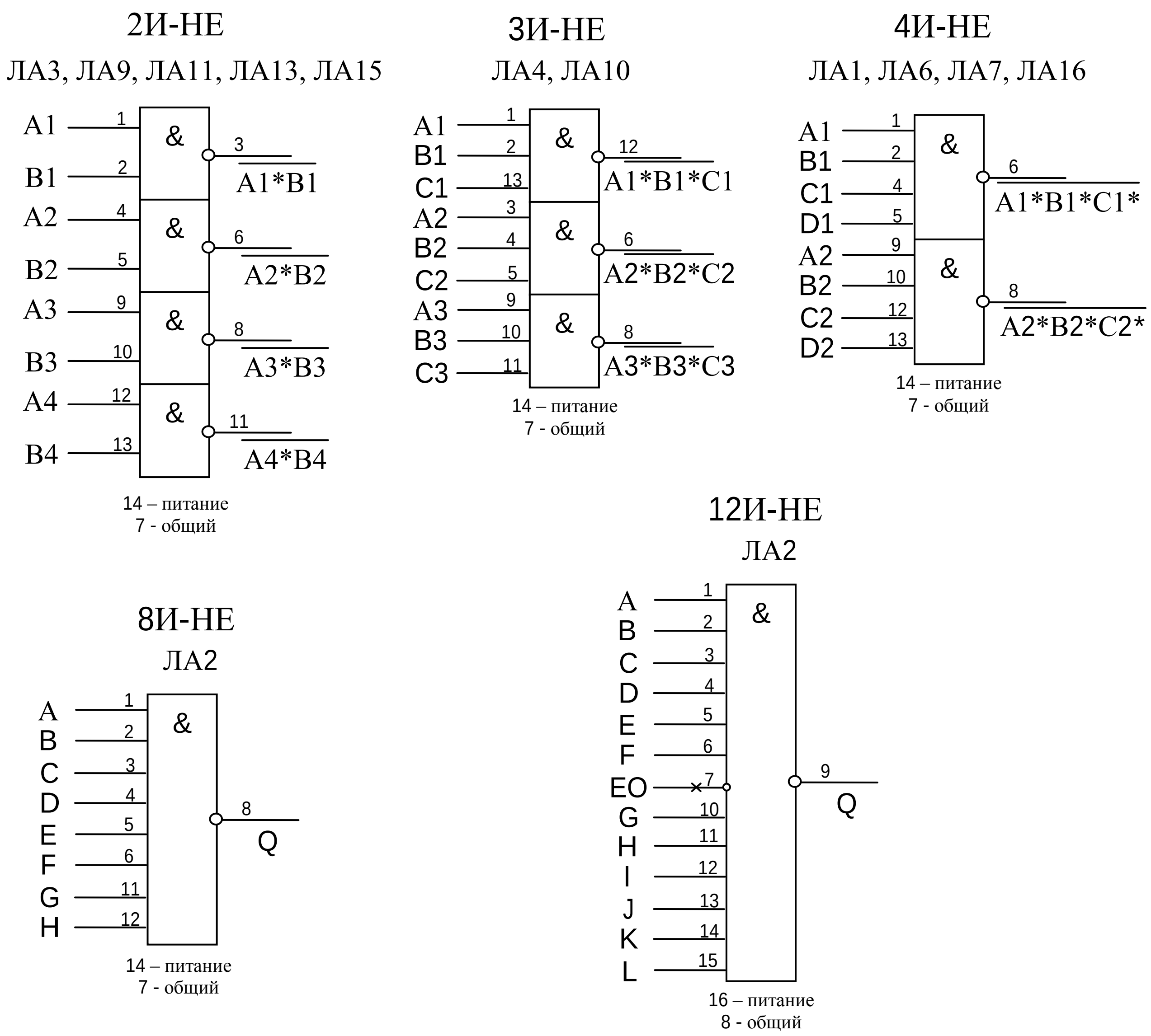


Рисунок 1.7 - Условные обозначения и цоколёвка микросхем типа ЛА.

EO – дополнительный вход, дающий разрешение по выходу (EO = 1 обеспечивает 3 состояние Z с очень большим выходным сопротивлением).

Микросхемы ЛА7…ЛА11, ЛА13, ЛА18 имеют выходы с открытым коллектором. Для формирования выходного перепада напряжения к выходу такого элемента необходимо подключить нагрузку. В качестве нагрузки могут выступать сегменты индикаторов, лампы накаливания, светодиоды, обмотки реле.

1.11 **Базис функции Пирса**

В настоящее время кроме элементов И-НЕ широкое распространение получили элементы ИЛИ-НЕ, выполняющие операцию Пирса. Это связано с применением для построения логических элементов полевых транзисторов.

Операция Пирса записывается следующим образом:

$$f = a \downarrow b = \overline{a + b}.$$

Таблица истинности данной операции имеет следующий вид:

a	b	f
0	0	1
1	0	0
0	1	0
1	1	0

Рассмотрим основные свойства этой функции.

1. Связь с переменными и константами:

$$\begin{array}{ll} a \downarrow a = \bar{a} & \qquad a \downarrow \bar{a} = 0 \\ a \downarrow 0 = \bar{a} & \qquad a \downarrow 1 = 0 \end{array}.$$

2. Связь с дизъюнкцией:

$$a + b = \overline{\overline{a + b}} = \overline{a \downarrow b} = (a \downarrow b) \downarrow (a \downarrow b) = (a \downarrow b) \downarrow 0.$$

3. Связь с конъюнкцией:

$$a \cdot b = \overline{\overline{a \cdot b}} = \overline{\bar{a} + \bar{b}} = \bar{a} \downarrow \bar{b} = (a \downarrow a) \downarrow (b \downarrow b) = (a \downarrow 0) \,|\, (b \downarrow 0).$$

Преобразование логического выражения в базис Пирса в общем виде происходит с использованием второго выражения теоремы разложения:

$$f(a,b,c) = \overline{\overline{f(a,b,c)}} = \overline{\overline{[a + f(0,b,c)] \cdot [\overline{a} + f(1,b,c)]}} = \overline{\overline{[a + f(0,b,c)]} + \overline{[\overline{a} + f(1,b,c)]}} =$$
$$= [a \downarrow f(0,b,c)] \downarrow [\overline{a} \downarrow f(1,b,c)]$$

Таким образом, окончательно можем записать:

$$f(a,b,c) = [a \downarrow f(0,b,c)] \downarrow [\overline{a} \downarrow f(1,b,c)].$$

Рассмотрим несколько примеров по преобразованию логических функций и реализации принципиальных схем в базисе операции Пирса.

Пример 1. Преобразовать заданную функцию в базис Пирса.

$$f(a,b,c) = (a + \overline{b}) \cdot (b + \overline{c}) \cdot (a + b + c).$$

Для преобразования возьмём двойную инверсию и преобразуем с помощью закона де-Моргана.

$$f(a,b,c) = \overline{\overline{(a + \overline{b}) \cdot (b + \overline{c}) \cdot (a + b + c)}} = \overline{\overline{(a + \overline{b})} + \overline{(b + \overline{c})} + \overline{(a + b + c)}} =$$
$$= (a \downarrow \overline{b}) \downarrow (b \downarrow \overline{c}) \downarrow (a \downarrow b \downarrow c).$$

Из рассмотренного примера можно сделать следующий вывод.

Для того чтобы перейти от функции, заданной в КНФ, к функции, заданной в базисе операции Пирса, достаточно в исходной функции заменить все операции дизъюнкция и конъюнкция на операцию Пирса.

Пример 2. Преобразовать функцию

$$f(a,b,c) = a \cdot \overline{b} + \overline{a} \cdot c + b \cdot \overline{c}$$

в базис Пирса и построить функциональную схему.

$$f(a,b,c) = \overline{\overline{a \cdot \overline{b} + \overline{a} \cdot c + b \cdot \overline{c}}} = \overline{\overline{a \cdot \overline{b}} * \overline{\overline{a} \cdot c} * \overline{b \cdot \overline{c}}} = \overline{(\overline{a} + b) \cdot (a + \overline{c}) \cdot (\overline{b} + c)} =$$
$$= \overline{(\overline{a} \downarrow b) \downarrow (a \downarrow \overline{c}) \downarrow (\overline{b} \downarrow c)} = [(\overline{a} \downarrow b) \downarrow (a \downarrow \overline{c}) \downarrow (\overline{b} \downarrow c)] \downarrow [(\overline{a} \downarrow b) \downarrow (a \downarrow \overline{c}) \downarrow (\overline{b} \downarrow c)]$$

На основании полученного выражения можем построить функциональную схему, представленную на рисунке 1.8.

Логическую операцию ИЛИ-НЕ реализуют интегральные микросхемы типа ЛЕ. Каждый из корпусов интегральных схем (ИС) типа ЛЕ содержит от двух до четырёх логических элементов. Условные обозначения и цоколёвка микросхем типа ЛЕ представлены на рисунке 1.9.

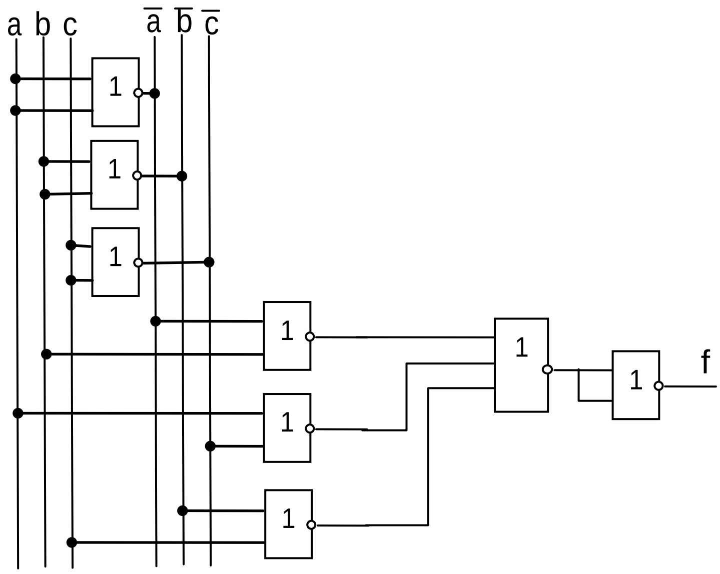


Рисунок 1.8 – Функциональная схема управляющего устройства на бесконтактных элементах ИЛИ-НЕ

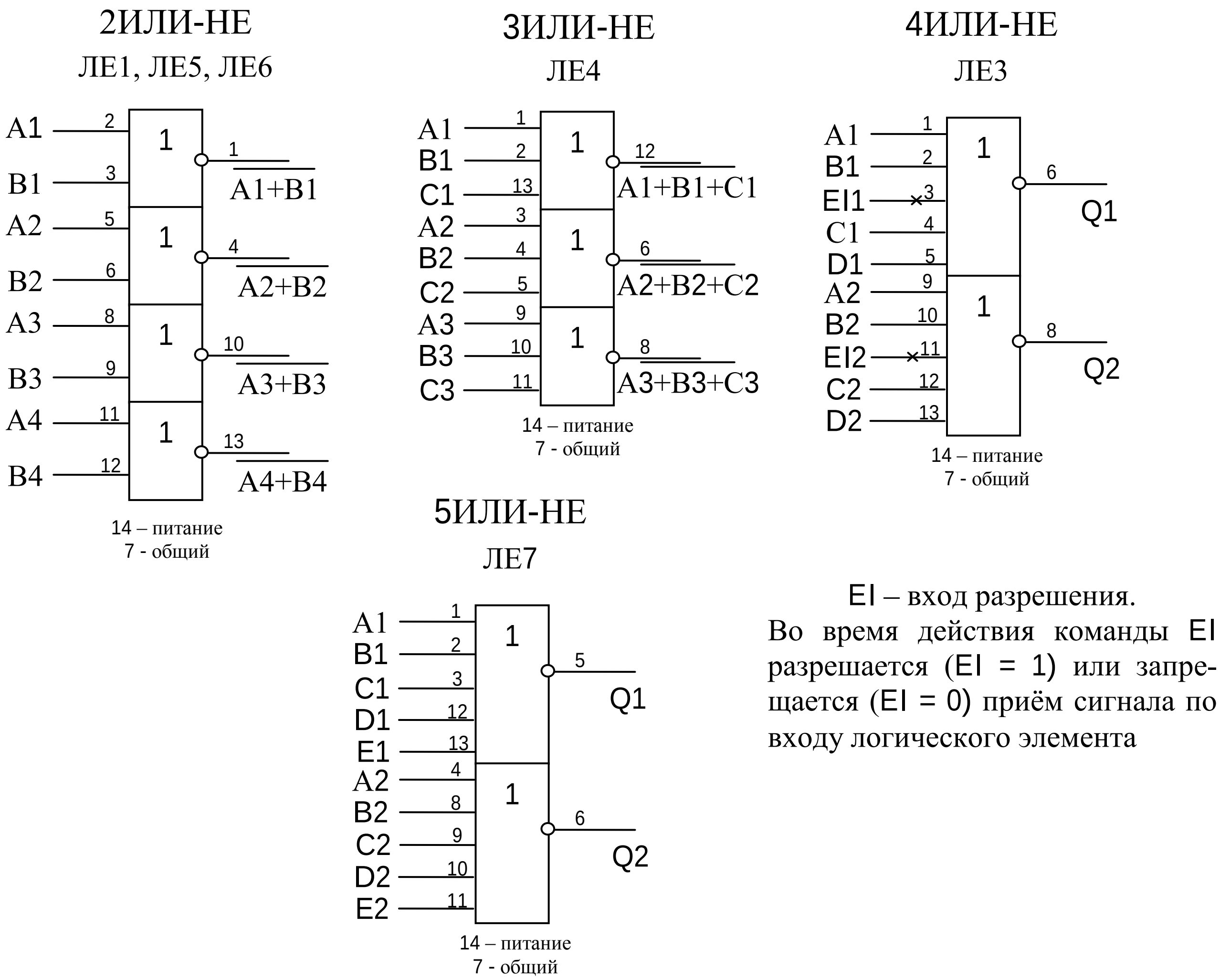


Рисунок 1.9 - Условные обозначения и цоколёвка микросхем типа ЛЕ

1.12 **Минимизация функций алгебры логики.**

1.11.1 Постановка задачи

При инженерном проектировании дискретных управляющих устройств обычно возникает задача оптимизации структуры автомата, т. е. получения экономичной и надёжной технической реализации. Это означает, что из двух схем автоматов следует выбирать ту, которая содержит меньшее число элементов, а при одинаковом числе элементов – ту, суммарное число входов используемых элементов которой будет наименьшее. Решение задачи оптимизации структуры автомата связано с проблемой минимизации функции алгебры логики, которая описывает принцип его действия.

Минимизация – это процесс нахождения такого эквивалентного выражения функции алгебры логики, которое содержит минимальное число вхождений переменных.

Минимизация может иметь и другие цели: получение выражения с минимальным числом инверсных переменных, либо с минимальным числом вхождений какой-либо одной переменной и др.

Большинство методов минимизации ориентированы на получение функции в виде дизъюнктивной нормальной формы (ДНФ).

Для минимизации функций алгебры логики используются алгебраические и графические методы.

Алгебраические методы:

1. Метод, основанный на применении теоремы Квайна.
2. Метод, основанный на применении теоремы Блэка.

Графический метод:

1. Метод минимизации с помощью карты Карно.

Рассмотрим эти методы подробно.

1.12.2 **Минимизация логических функций с помощью карты Карно**

При использовании этого метода функция задаётся в виде координатной карты состояний, известной также как карта Карно.

Карта содержит 2^n клеток по числу наборов значений входных переменных. Каждая клетка определяется координатами столбца и строки и соответствует определённому набору. Поскольку карта строится на плоскости, то все переменные разбиваются на две группы так, что одна группа определяет координаты строки, а другая – столбца. В клетки карты Карно проставляется значение функции на данном наборе.

Например, пусть имеется карта Карно следующего вида:

1	0	0	1	0	0	0	1
1	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

По карте Карно можно записать логическое выражение функции в виде СДНФ:

$$f(a,b,c,d,e)_{СДНФ} = a\cdot\overline{b}\cdot\overline{c}\cdot d\cdot\overline{e} + a\cdot b\cdot\overline{c}\cdot d\cdot\overline{e} + a\cdot\overline{b}\cdot\overline{c}\cdot d\cdot e + a\cdot b\cdot\overline{c}\cdot d\cdot e + \overline{a}\cdot\overline{b}\cdot\overline{c}\cdot\overline{d}\cdot\overline{e}.$$

Как видно, выражение содержит достаточно большое количество переменных (25 переменных).

Для того чтобы найти минимизированное выражение функции в виде ДНФ необходимо:

1. Все единицы карты Карно охватить минимальным числом контуров. При этом, контуры должны быть максимальны по размеру, включать $2^к$ клеток (1, 2, 4, 8, 16, 32, 64 и т. д.) и быть симметрич-

ными относительно осей симметрии всей карты Карно, половины карты Карно, четверти и т. д.

2. Каждый контур даёт в общее выражение минимизированной функции конъюнкцию тех переменных, которые в данном контуре не меняют своего значения. Причём, если в контуре переменная в наборах равна 1, то в конъюнкцию записывается сама переменная, если равна 0 – то записывается её инверсия.
3. Минимизированное выражение получается как дизъюнкция конъюнкций, соответствующих каждому контуру.

Для того чтобы найти минимизированное выражение функции в виде КНФ необходимо:

1. Все нули карты Карно охватить минимальным числом контуров. К контурам предъявляются аналогичные требования.
2. Каждый контур даёт в общее выражение минимизированной функции дизъюнкцию тех переменных, которые в данном контуре не меняют своего значения. Причём, если в контуре переменная в наборах равна 0, то в дизъюнкцию записывается сама переменная, если равна 1 – то записывается её инверсия.
3. Минимизированное выражение получается как конъюнкция дизъюнкций, соответствующих каждому контуру.

Одни и те же наборы можно использовать в разных контурах неограниченное число раз. Контуры могут состоять из $2^к$ подконтуров (1, 2, 4 и т.д.), расположенных симметрично относительно осей симметрии карты Карно.

Используя рассмотренные правила, запишем минимизированное выражение в виде ДНФ по карте Карно, представленной выше.

$$f(a,b,c,d,e)_{ДНФ} = a \cdot \bar{c} \cdot d + \bar{b} \cdot \bar{c} \cdot d \cdot \bar{e} = d \cdot \bar{c} \cdot (a + \bar{b} \cdot \bar{e}).$$

Полученное выражение содержит 5 вхождение переменных и значительно проще полученного выражения в форме СДНФ.

Запишем выражение в виде КНФ. Для этого охватим контурами все нули, расположенные в карте Карно.

c c

b

a

1	0	0	1	0	0	0	1	d
1	0	0	1	0	0	0	0	d e
0	0	0	0	0	0	0	0	e
0	0	0	0	0	0	0	0	

$$f(a,b,c,d,e)_{КНФ} = d \cdot \bar{c} \cdot (a+\bar{e}) \cdot (a+\bar{b}) .$$

Рассмотрим ещё несколько примеров.

Пример. Записать минимизированное выражение в виде ДНФ и КНФ для логической функции, заданной следующей картой Карно.

c c

b

a

1	0	1	1	0	0	1	1	d
1	0	1	1	0	0	0	0	d e
0	0	0	0	0	0	0	0	e
1	0	1	1	0	0	0	1	

$$f(a,b,c,d,e)_{ДНФ} = a \cdot b \cdot d + a \cdot \bar{c} \cdot d + a \cdot b \cdot \bar{e} + \bar{b} \cdot \bar{c} \cdot \bar{e} + \bar{a} \cdot \bar{b} \cdot d \cdot \bar{e} =$$
$$= a \cdot d \cdot (b+\bar{c}) + \bar{e} \cdot (a \cdot b + \bar{b} \cdot (\bar{c} + \bar{a} \cdot d))$$
$$f(a,b,c,d,e)_{КНФ} = (a+\bar{b}) \cdot (d+\bar{e}) \cdot (a+\bar{e}) \cdot (\bar{a}+b+\bar{c}) \cdot (a+\bar{c}+d) .$$

Пример. Минимизировать функцию, заданную в виде ДНФ с помощью карты Карно и реализовать устройство управления на элементах И-НЕ и ИЛИ-НЕ.

$$f(a,b,c,d) = a \cdot \bar{b} \cdot d + b \cdot \bar{c} \cdot d + \bar{a} \cdot c \cdot d + \bar{a} \cdot \bar{b} \cdot d .$$

Для построения карты Карно запишем заданную функцию в виде СДНФ. Для этого в каждом слагаемом восстановим недостающую переменную, домножив на единицу.

$$f(a,b,c,d) = a\cdot\bar{b}\cdot d\cdot(c+\bar{c}) + b\cdot\bar{c}\cdot d\cdot(a+\bar{a}) + \bar{a}\cdot c\cdot d\cdot(b+\bar{b}) + \bar{a}\cdot\bar{b}\cdot d\cdot(c+\bar{c}) =$$
$$= a\cdot\bar{b}\cdot d\cdot c + a\cdot\bar{b}\cdot d\cdot\bar{c} + b\cdot\bar{c}\cdot d\cdot a + b\cdot\bar{c}\cdot d\cdot\bar{a} + \bar{a}\cdot c\cdot d\cdot b + \bar{a}\cdot c\cdot d\cdot\bar{b} +$$
$$+ \bar{a}\cdot\bar{b}\cdot d\cdot c + \bar{a}\cdot\bar{b}\cdot d\cdot\bar{c} = a\cdot\bar{b}\cdot d\cdot c + a\cdot\bar{b}\cdot d\cdot\bar{c} + b\cdot\bar{c}\cdot d\cdot a + b\cdot\bar{c}\cdot d\cdot\bar{a} + \bar{a}\cdot c\cdot d\cdot b +$$
$$+ \bar{a}\cdot\bar{b}\cdot d\cdot c + \bar{a}\cdot\bar{b}\cdot d\cdot\bar{c}$$

Полученная СДНФ содержит семь конституэнтов, следовательно в карте Карно будет семь единиц. Восстановим карту Карно.

b

a

0	0	0	0
1	1	1	1
1	0	1	1
0	0	0	0

d

c

По карте Карно записываем минимизированное выражение

$$f(a,b,c,d) = \bar{a}\cdot d + \bar{b}\cdot d + \bar{c}\cdot d = d\cdot(\bar{a}+\bar{b}+\bar{c})$$

1.12.3 Алгебраический метод минимизации с использованием теоремы Квайна

Метод основан на применении теоремы Квайна.

Т е о р е м а: Если в СДНФ функции алгебры логики произвести все возможные операции неполного склеивания , а затем все возможные операции элементарного поглощения, то полученная форма функции будет сокращённой.

Логическая операция вида:

$$A\cdot x + A\cdot\bar{x} = A$$

называется операцией полного склеивания по переменной x, а вида

$$A \cdot x + A \cdot \overline{x} = A \cdot x + A \cdot \overline{x} + A$$

операцией неполного склеивания по переменной x.

Рассмотрим пример применения теоремы Квайна для минимизации логической функции.

Пример. Выполнить минимизацию логической функции, заданной таблицей истинности, с помощью теоремы Квайна.

Таблица истинности.

a	b	c	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

По таблице истинности записываем СДНФ логической функции:

$$f(a,b,c) = a \cdot b \cdot c + \overline{a} \cdot b \cdot c + \overline{a} \cdot \overline{b} \cdot c + a \cdot \overline{b} \cdot \overline{c}$$

Выполним все возможные операции неполного склеивания.

$$f(a,b,c) = \underbrace{a \cdot b \cdot c + \underline{\underline{\overline{a} \cdot b \cdot c}}}_{1} + \underline{\underline{\overline{a} \cdot \overline{b} \cdot c}} + a \cdot \overline{b} \cdot \overline{c} = a \cdot b \cdot c + \overline{a} \cdot b \cdot c + b \cdot c + \overline{a} \cdot \overline{b} \cdot c +$$

$$+ \overline{a} \cdot b \cdot c + \overline{a} \cdot c + a \cdot \overline{b} \cdot \overline{c} = b \cdot c + \overline{a} \cdot c + a \cdot \overline{b} \cdot \overline{c}$$

Дизъюнктивная нормальная форма (ДНФ) функции называется тупиковой, если ни один её член не может быть удалён без нарушения эквивалентности её исходной функции. К элементарным конъюнкциям тупиковой ДНФ операция склеивания неприменима.

Тупиковая ДНФ называется минимальной, если она содержит наименьшее число вхождений переменных по сравнению с другими тупиковыми формами этой функции.

В общем случае полученная сокращённая ДНФ может не быть тупиковой и не быть минимальной. Поэтому необходимо применить ещё закон обобщённого склеивания к сокращённой ДНФ, получить все возможные тупиковые формы и выбрать из них минимальную.

Эта операция может быть формализована с помощью импликантной матрицы Квайна. Она представляет собой таблицу, столбцы которой соответствуют исходным конституентам, строки – импликантам сокращённой ДНФ. В клетках на пересечении строк и столбцов отмечают вхождение данного импликанта в данный конституент. Минимальная ДНФ записывается как дизъюнкция тех импликантов, которые в минимальном количестве покрывают все исходные конституенты.

Рассмотрим пример.

Пример. Выполнить минимизацию логической функции, заданной таблицей истинности.

Таблица истинности.

a	b	c	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

По таблице истинности записываем СДНФ логической функции:

$$f(a,b,c) = a\cdot b\cdot c + a\cdot b\cdot\bar{c} + \bar{a}\cdot\bar{b}\cdot\bar{c} + a\cdot\bar{b}\cdot\bar{c}$$

Выполним все возможные операции неполного склеивания.

$$f(a,b,c) = \underbrace{a\cdot b\cdot c + \overbrace{a\cdot b\cdot\bar{c}}^{3}}_{1} + \underbrace{\bar{a}\cdot\bar{b}\cdot\bar{c} + \overbrace{a\cdot\bar{b}\cdot\bar{c}}^{3}}_{2} = a\cdot b\cdot c + a\cdot b\cdot\bar{c} + a\cdot b + .$$

$$+\bar{a}\cdot\bar{b}\cdot\bar{c} + a\cdot\bar{b}\cdot\bar{c} + \bar{b}\cdot\bar{c} + a\cdot b\cdot\bar{c} + a\cdot\bar{b}\cdot\bar{c} + a\cdot\bar{c} = a\cdot b + \bar{b}\cdot\bar{c} + a\cdot\bar{c}$$

Строим матрицу Квайна.

	$a \cdot b \cdot c$	$a \cdot b \cdot \bar{c}$	$\bar{a} \cdot \bar{b} \cdot \bar{c}$	$a \cdot \bar{b} \cdot \bar{c}$
$a \cdot b$	X	X		
$\bar{b} \cdot \bar{c}$		X		X
$a \cdot \bar{c}$			X	X

Минимальное покрытие всех конституентов функции обеспечивают первый и третий импликанты. Можем записать минимальную ДНФ.

$$f(a,b,c) = a \cdot b + \bar{b} \cdot \bar{c}$$

Для формального определения минимальной ДНФ по матрице Квайна записывают формулу покрытия. Формула покрытия записывается следующим образом. Каждая импликанта обозначается новым символом. Для каждого конституента записывается дизъюнкция тех импликантов, которые входят в данный конституент. Полученные дизъюнкции связываются через конъюнкцию. К формуле покрытия применимы все законы поглощения, что позволяет получить не только все возможные варианты тупиковых форм, но и минимальную форму.

Для нашего примера обозначим импликанты буквами латинского алфавита. Можем записать формулу покрытия:

F = A*(A + B)*C*(B + C) = A*C

Как видим, результат совпадает с полученным ранее.

1.12.4 Алгебраический метод минимизации с использованием теоремы Блэка

Если минимизируемая логическая функция задана в виде ДНФ, то можно развернуть её в СДНФ и применить любой метод минимизации, соответсвующий этой форме. Однако, избежать процедуры развёртывания можно, если воспользоваться методом Блэка.

Т е о р е м а: Если в произвольной ДНФ произвести все возможные операции обобщённого склеивания и элементарного поглощения, то будет получена сокращённая ДНФ.

Порядок минимизации.

1. В заданной ДНФ выполняют все возможные операции обобщённого склеивания:

   $$a\cdot b+\bar{a}\cdot c=a\cdot b+\bar{a}\cdot c+b\cdot c$$

   Полученное выражение снова преобразовывают с помощью операции обобщённого склеивания до тех пор, пока получение новых членов с помощью этого закона станет невозможно.
2. Применяя в полученном выражении операцию поглощения, находят сокращённую нормальную форму.

Пример. Минимизировать логическую функцию, заданную в виде ДНФ.

$$f(a,b,c)=\overbrace{a}^{2}+\underbrace{\bar{b}\cdot c+\overbrace{\bar{a}\cdot b\cdot c}^{2}}_{1}=\bar{b}\cdot c+\bar{a}\cdot b\cdot c+\underbrace{\bar{a}\cdot c+a}_{3}+\bar{a}\cdot b\cdot c+b\cdot c=$$

$$=\bar{b}\cdot c+\bar{a}\cdot b\cdot c+\underbrace{\bar{a}\cdot c+a+c}+\bar{a}\cdot b\cdot c+b\cdot c=a+c$$

1.13 Понятие о булевых матрицах

Среди множества дискретных устройств выделяется класс автоматов, построенных на контактных реле или вентилях, структура которых не может быть непосредственно описана в виде формулы с использованием операций булевой алгебры. Примером таких устройств может служить мостиковая структура, представленная на рисунке 1.10.

Составить уравнения алгебры логики для данной структуры достаточно затруднительно. В тоже время применение мостиковой структуры позволяет значительно упростить схему.

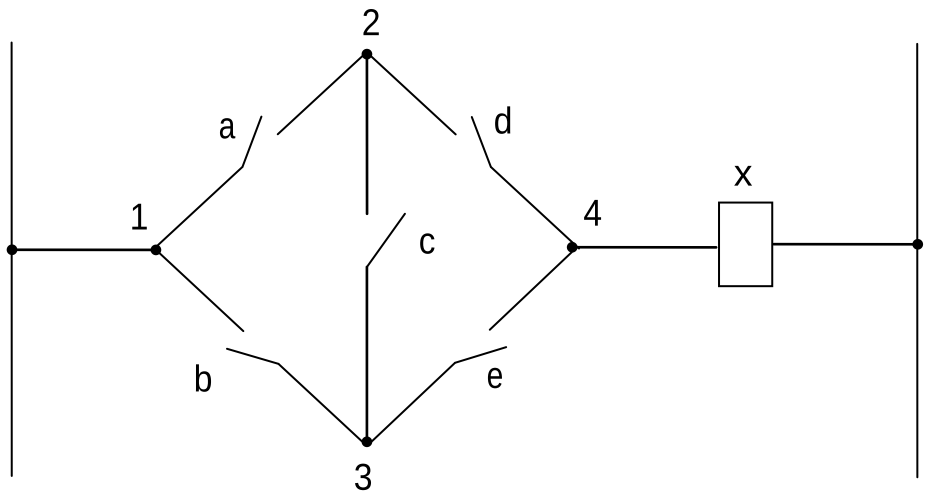


Рисунок 1.10 – Мостиковая структура

Для описания таких автоматов применяются булевы матрицы.

Булева матрица содержит столько строк и столбцов, сколько узлов содержит схема автомата. Элемент A_{ij} матрицы соответствует алгебраическому выражению функции цепи, соединяющей непосредственно узел i с узлом j. В общем случае это может быть константа 0 или 1, одна переменная или её инверсия, или любое алгебраическое выражение.

Для схемы, представленной на рисунке, булева матрица будет иметь вид.

$$M = \begin{vmatrix} 1 & a & b & 0 \\ a & 1 & c & d \\ b & c & 1 & e \\ 0 & d & e & 1 \end{vmatrix}$$

По булевой матрице можно определить алгебраическое выражение функций всех цепей в схеме от узла i к узлу j. Для этого достаточно вычислить определитель $|A_{ij}|$ булевой подматрицы, получаемой вычёркиванием i-го столбца и j-той строки. В нашем случае определим логическое выражение проводимости между узлами 1-4.

$$A_{ij} = \begin{vmatrix} a & b & 0 \\ 1 & c & d \\ c & 1 & e \end{vmatrix}$$

При вычислении булева определителя используются две операции – конъюнкция и дизъюнкция. Способ вычисления такой же, как и в обычной алгебре, только все члены положительны и связываются операцией дизъюнкции. Вычислим определитель $A_{1,4}$ разложением матрицы по элементам первой строки.

$$A_{1,4} = a * \begin{vmatrix} c & d \\ 1 & e \end{vmatrix} + b * \begin{vmatrix} 1 & d \\ c & e \end{vmatrix} = a \cdot c \cdot e + a \cdot d + b \cdot e + b \cdot c \cdot d$$

Обычно разложение целесообразно проводить по той строке или столбцу, который содержит наименьшее число единиц.

Пример. Записать выражение для цепей проводимости между узлами 1-5 мостиковой структуры, представленной на рисунке 1.11.

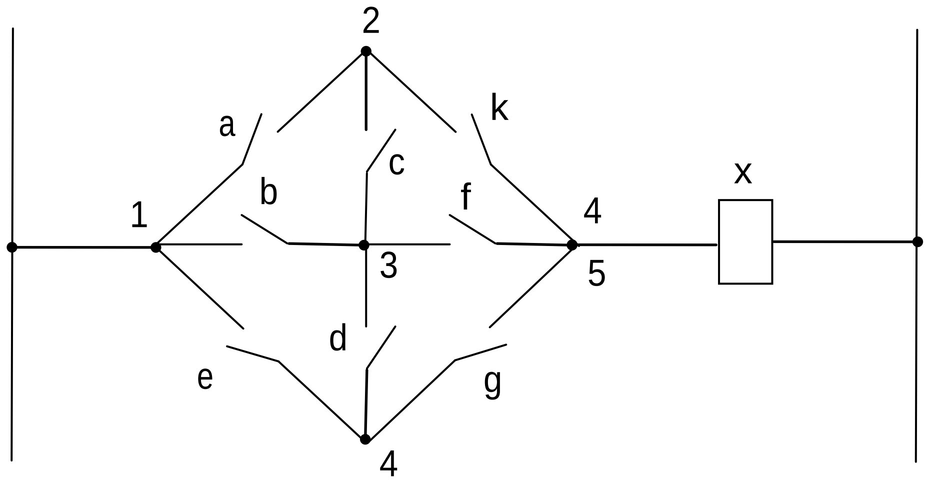


Рисунок 1.11 – Мостиковая структура

Построим булеву матрицу.

$$M = \begin{matrix} 1 & a & b & e & 0 \\ a & 1 & c & 0 & k \\ b & c & 1 & d & f \\ e & 0 & d & 1 & g \\ 0 & k & f & g & 1 \end{matrix}.$$

Для записи логического выражения определим определитель подматрицы вида:

$$A_{1,5}=\begin{array}{cccc} a & b & e & 0 \\ 1 & c & 0 & k \\ c & 1 & d & f \\ 0 & d & 1 & g \end{array} = a*\begin{vmatrix} c & 0 & k \\ 1 & d & f \\ d & 1 & g \end{vmatrix} + b*\begin{vmatrix} 1 & 0 & k \\ c & d & f \\ 0 & 1 & g \end{vmatrix} + e*\begin{vmatrix} 1 & c & k \\ c & 1 & f \\ 0 & d & g \end{vmatrix} =$$

$$= a\cdot(c\cdot d\cdot g + k + c\cdot f) + b\cdot(d\cdot g + c\cdot k + f) + e\cdot(g + c\cdot d\cdot k + d\cdot f)$$

Пример. По заданному выражению построить принципиальную схему.

$$f(a,b,c,d,e,f) = d + e\cdot f + e\cdot b\cdot c + a\cdot c + a\cdot b\cdot f = d + e\cdot(f + b\cdot c) + a\cdot(c + b\cdot f)$$

По данному выражению можем получить следующие варианты схем

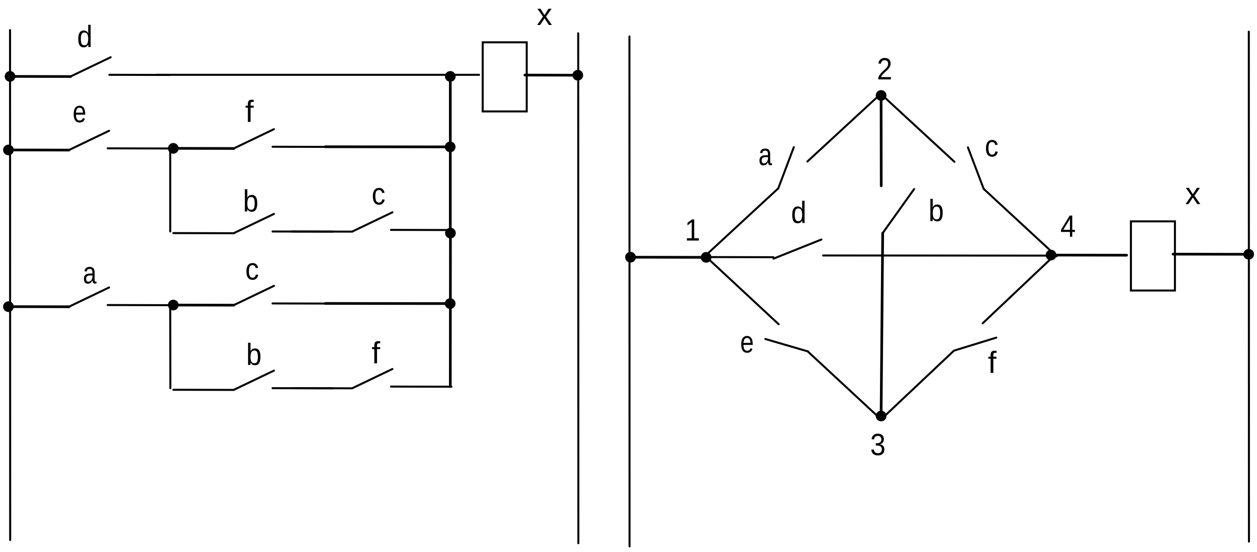


Рисунок 1.12 – Структурная схема устройства управления.

Как видно из рисунка вторая схема содержит значительно меньше контактов (в первой схеме – 9 контактов, во второй – 6 контактов).

Таким образом, получаемое при синтезе устройство целесообразно проанализировать с точки зрения возможности получения мостиковой структуры для существенного упрощения схемы.

2. СИНТЕЗ КОМБИНАЦИОННЫХ АВТОМАТОВ. СТРУКТУРНЫЙ АНАЛИЗ КОМБИНАЦИОННЫХ АВТОМАТОВ

2.1 Порядок синтеза комбинационных автоматов

Комбинационный автомат (однотактная схема) – это устройство релейного действия с n входами и m выходами, у которого при идеальных (безинерционных) элементах состояние выходов (сигналы на выходах) в некоторый момент времени однозначно определяется состоянием входов (сигналами на входах) в тот же момент времени.

Комбинационный автомат (КА) может быть представлен в виде многополюсника следующего вида:

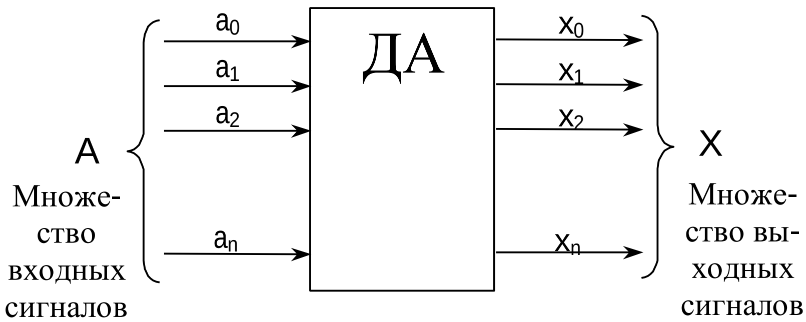


Рисунок 2.1 – Функциональная схема комбинационного автомата.

Так как сигналы на выходах КА определяются только значениями входных сигналов, следовательно, комбинационный автомат однозначно описывается формулами алгебры логики.

Однако, релейный комбинационный автомат выполнен не на идеальных релейных элементах и обладает некоторой инерционностью. Инерционность элементов связана с наличием различных задержек при переключении логических элементов или контактов схемы. Это приводит к тому, что после изменения сигналов на входах сигналы на выходе, соответствующие новому состоянию, появляются не сразу, а с некоторой задержкой, причём в переходный период возможно появление на выходе некоторых промежуточных значений сигналов. Такой процесс получил название состязаний.

В задачу синтеза комбинационных автоматов входит построение схемы – структуры автомата по заданным условиям его работы и при заданном наборе элементов, причём, как правило, ставится условие получить по возможности простую (с минимальным числом элементов) схему и проверить её на состязания.

Задание комбинационного автомата сводится к заданию тех функций, которые он должен реализовать. Число функций равно числу выходов автомата. Для задания комбинационного автомата используют те же способы, что и для задания функций алгебры логики, которые были рассмотрены выше (табличный, координатный, числовой, аналитический).

Для построения схемы комбинационного автомата необходимо записать логические функции, связывающие каждый выход автомата с входными переменными:

$$\begin{cases} x_0 = f_0(a_0, a_1, a_2 ... a_n); \\ x_1 = f_1(a_0, a_1, a_2 ... a_n); \\ ... \\ x_m = f_m(a_0, a_1, a_2 ... a_n). \end{cases}$$

Порядок синтеза комбинационных автоматов.

1. Абстрактный синтез
   1.1 Составление словесной формулировки условий работы схемы. Можно использовать любой другой способ задания автомата (табличный, графический и др.).
   1.2 Формализация условий работы автомата (обозначение переменных, составление таблицы истинности или карты Карно).
   1.3 Минимизация автомата, т. е. получение минимизированного алгебраического выражения.

2. Структурный синтез.

   2.1 Выбор базисных элементов, аппаратов, в случае необходимости кодирование переменных.

   2.2 Преобразование логической функции в заданный базис.

   2.3 Построение структурной схемы.

   2.4 Построение принципиальной схемы.

   2.5 Анализ полученной схемы на правильность работы и возможность возникновения состязаний.

Рассмотрим пример синтеза комбинационного автомата.

Пример. Синтезировать схему сравнения двух трёхразрядных двоичных чисел по выражению:

$$A > B \quad \text{где} \quad A = a_2a_1a_0,\ B = b_2b_1b_0$$

Дадим словесное описание условий работы автомата.

Автомат можно представить следующим образом:

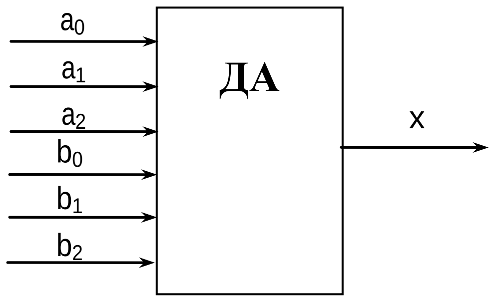


Рисунок 2.2 – Функциональная схема устройства сравнения

Условия работы автомата описываются следующими выражениями

$$\begin{cases} x = 1 & \text{если} \quad A > B \\ x = 0 & \text{если} \quad A \le B \end{cases}.$$

Для формального описания автомата составим таблицу истинности. Так как входных переменных 6, то число входных наборов будет равно

$$N = 2^6 = 64.$$

Для уменьшения размера, таблицу истинности составим в следующем виде:

	a_2	a_1	a_0	b_2	b_1	b_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
A_1	0	0	0	0	0	0	0	1	1	1	1	1	1	1
A_2	0	0	1	0	0	1	0	0	1	1	1	1	1	1
A_3	0	1	0	0	1	0	0	0	0	1	1	1	1	1
A_4	0	1	1	0	1	1	0	0	0	0	1	1	1	1
A_5	1	0	0	1	0	0	0	0	0	0	0	1	1	1
A_6	1	0	1	1	0	1	0	0	0	0	0	0	1	1
A_7	1	1	0	1	1	0	0	0	0	0	0	0	0	1
A_8	1	1	1	1	1	1	0	0	0	0	0	0	0	0

В данной таблице значение X_1 соответствует сравнению первого набора A_1 числа А со всеми значениями числа В, значение X_2 соответствует сравнению второго набора и т. д.

Для записи логического выражения на основании таблицы истинности построим карту Карно.

a_2 a_2 a_1 a_0

	001 1	101 5	111 7	011 3	010 2	110 6	100 4	000 0
000 0	1	1	1	1	1	1	1	0
100 4	0	1	1	0	0	1	0	0
110 6	0	0	1	0	0	0	0	0
010 2	0	1	1	1	0	1	1	0
011 3	0	1	1	0	0	1	1	0
111 7	0	0	0	0	0	0	0	0
101 5	0	0	1	0	0	1	0	0
001 1	0	1	1	1	1	1	1	0

b_2 b_1 b_2 b_0

Запишем логическое выражение для выходной переменной.

$$
\begin{aligned}
x &= a_2 \cdot \overline{b_2} + a_1 \cdot \overline{b_1} \cdot \overline{b_2} + a_1 \cdot a_2 \cdot \overline{b_1} + a_0 \cdot a_2 \cdot \overline{b_0} \cdot \overline{b_1} + a_0 \cdot \overline{b_0} \cdot \overline{b_1} \cdot \overline{b_2} + a_0 \cdot a_1 \cdot \overline{b_0} \cdot \overline{b_2} + \\
&+ a_0 \cdot a_1 \cdot a_2 \cdot \overline{b_0} = a_2 \cdot \overline{b_2} + a_1 \cdot \overline{b_1} \cdot (\overline{b_2} + a_2) + a_0 \cdot \overline{b_0} \cdot (a_2 \cdot \overline{b_1} + \overline{b_1} \cdot \overline{b_2} + a_1 \cdot \overline{b_2} + \\
&+ a_1 \cdot a_2) = a_2 \cdot \overline{b_2} + a_1 \cdot \overline{b_1} \cdot (\overline{b_2} + a_2) + a_0 \cdot \overline{b_0} \cdot (\overline{b_1} \cdot (a_2 + \overline{b_2}) + a_1 \cdot (\overline{b_2} + a_2)) = \\
&= a_2 \cdot \overline{b_2} + a_1 \cdot \overline{b_1} \cdot (\overline{b_2} + a_2) + a_0 \cdot \overline{b_0} \cdot ((\overline{b_1} + a_1) \cdot (a_2 + \overline{b_2}))
\end{aligned}
.
$$

На основании полученного выражения строим схему автомата на контактных элементах.

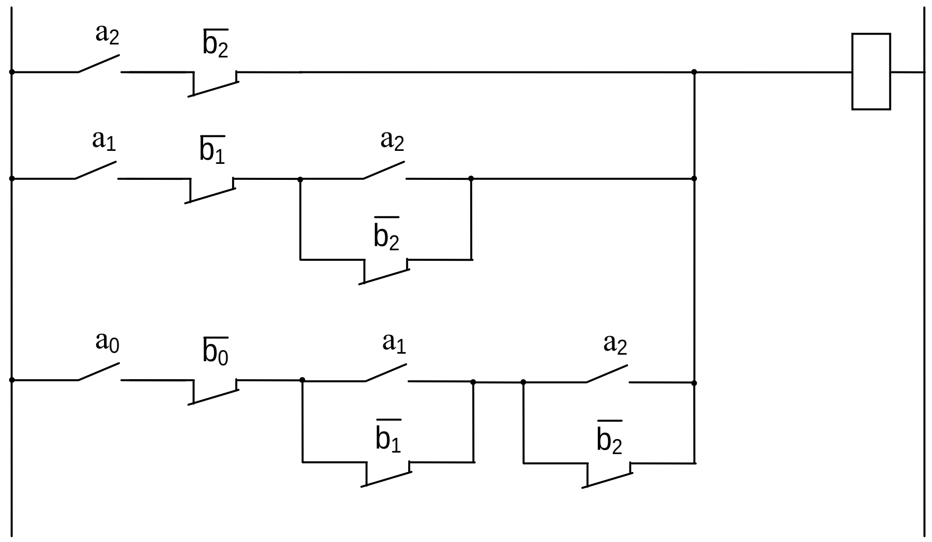


Рисунок 2.3 – Схема устройства сравнения двух чисел на контактных элементах

Построим схему на бесконтактных элементах И-НЕ. Для этого преобразуем выражение в базис Шеффера.

$$
\begin{aligned}
x &= \overline{\overline{a_2 \cdot \overline{b_2} + a_1 \cdot \overline{b_1} \cdot \overline{b_2} + a_1 \cdot a_2 \cdot \overline{b_1} + a_0 \cdot a_2 \cdot \overline{b_0} \cdot \overline{b_1} + a_0 \cdot \overline{b_0} \cdot \overline{b_1} \cdot \overline{b_2} + a_0 \cdot a_1 \cdot \overline{b_0} \cdot \overline{b_2} + a_0 \cdot a_1 \cdot a_2 \cdot \overline{b_0}}} = \\
&= (a_2 \mid \overline{b_2}) \mid (a_1 \mid \overline{b_1} \mid \overline{b_2}) \mid (a_1 \mid a_2 \mid \overline{b_1}) \mid (a_0 \mid a_2 \mid \overline{b_0} \mid \overline{b_1}) \mid (a_0 \mid \overline{b_0} \mid \overline{b_1} \mid \overline{b_2}) \mid (a_0 \mid a_1 \mid \overline{b_0} \mid \overline{b_2}) \mid \\
&\mid (a_0 \mid a_1 \mid a_2 \mid \overline{b_0})
\end{aligned}
$$

Схема на бесконтактных элементах И-НЕ представлена на рисунке 2.4

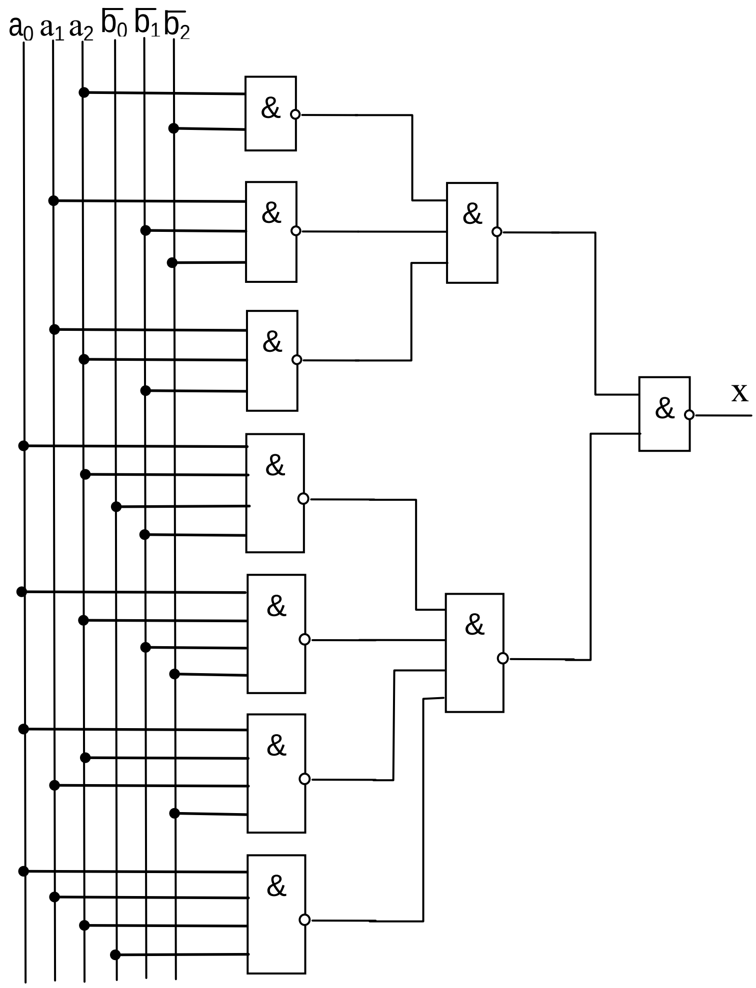


Рисунок 2.4 – Схема устройства сравнения двух чисел на бесконтактных элементах И-НЕ

2.2 Учёт недоопределённости комбинационного автомата при его задании и синтезе

Комбинационные автоматы с n входами в общем случае имеют 2^n наборов или состояний, различающихся комбинациями значений двухзначного сигнала (0 и 1) на этих входах. Каждому из этих состояний на каждом выходе x_i автомата соответствует определённое значение (0 или 1) выходного сигнала, т. е. каждый выход автомата реализует некоторую булеву функцию от входных переменных:

$$x_i = f_i(a_1, \ldots, a_n).$$

При формулировании требований к работе автомата по условиям его взаимодействия с внешними устройствами не для всех 2^n состояний входов бывают определены значения выходных сигналов. Для некоторых состояний бывает несущественно состояние сигнала на том или ином выходе.

Комбинационный автомат называется полностью определённым, если для всех входных наборов определены значения всех выходных переменных.

Если хотя бы на одном входном наборе не определено или безразлично значение выходного сигнала, то комбинационный автомат называется не полностью определённым.

Это приводит к тому, что все входные наборы распадаются на три подмножества:

1. Обязательные наборы α , при которых на данном выходе сигнал принимает значение единица.
2. Запрещённые наборы β , при которых на данном выходе сигнал принимает значение ноль.
3. Условные наборы.

Условные наборы – это не используемые или безразличные наборы, т. е. на этих наборах значение функции не существенно или может быть доопределено с дальнейшим изменением условий задачи. Для однозначного задания автомата в этом случае необходимо записать два подмножества:

$$f = (\alpha 1, \alpha 2 \ldots / \beta 1, \beta 2 \ldots)$$
$$f = (\alpha 1, \alpha 2 \ldots / \gamma 1, \gamma 2 \ldots)$$

С учётом условных наборов можем получить несколько неравносильных алгебраических выражений для данной функции. Каждый значок ~ можем заменить либо на ноль, либо на единицу. Из полученных выражений выбираем наиболее простое, минимальное, но с учётом базисных элементов реализации.

Пример. Построить структурную схему на бесконтактных элементах комбинационного автомата, заданного следующей картой Карно.

0	~	0	0	1	1	0	0
0	1	1	0	0	0	0	0
0	~	~	0	1	~	~	1
0	0	0	0	1	~	0	0

Запишем выражение минимизированной функции в виде ДНФ, заменив, где это необходимо для получения максимальных контуров, знак ~ на 1.

$$f(a,b,c,d,e)_{ДНФ} = a\cdot c\cdot e + \bar{a}\cdot b\cdot \bar{e} + \bar{a}\cdot d\cdot e\,.$$

Для построения структурной схемы, полученное выражение преобразуем в базис Шеффера.

$$f(a,b,c,d,e)_{ДНФ} = (a\,|\,c\,|\,e)\,|\,(\bar{a}\,|\,b\,|\,\bar{e})\,|\,(\bar{a}\,|\,d\,|\,e)$$

На основании полученного выражения можем построить структурную схему на бесконтактных элементах И-НЕ.

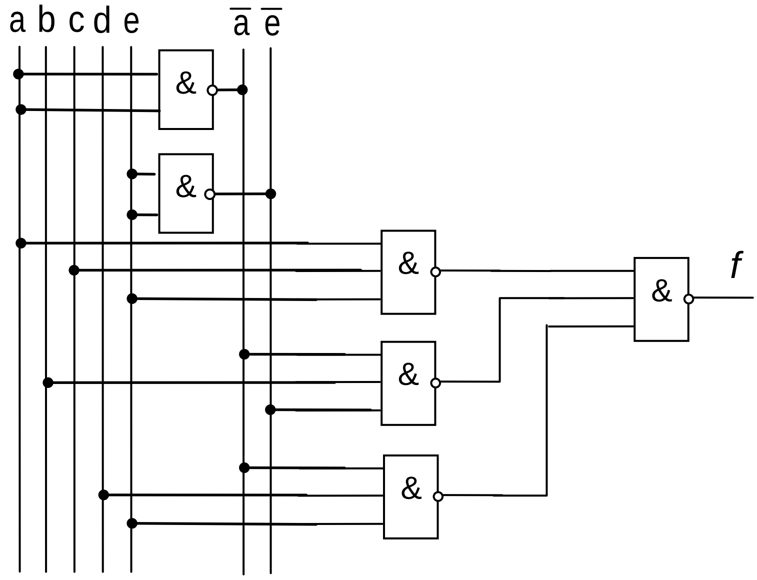


Рисунок 2.5 – Структурная схема устройства управления на элементах И-НЕ

Запишем выражение минимизированной функции в виде КНФ, заменив, где это необходимо для получения максимальных контуров, знак ~ на 0.

0	~	0	0	1	1	0	0
0	1	1	0	0	0	0	0
0	~	~	0	1	~	~	1
0	0	0	0	1	~	0	0

$$f(a,b,c,d,e)_{КНФ} = (\bar{a}+e)\cdot(\bar{a}+c)\cdot(b+e)\cdot(a+d+\bar{e}) =$$
$$= (\bar{a}+e\cdot c)\cdot(b+e)\cdot(a+d+\bar{e})$$

Преобразуем полученное выражение в базис операции Пирса.

$$f(a,b,c,d,e)_{КНФ} = (\bar{a}+e\cdot c)\cdot(b+e)\cdot(a+d+\bar{e}) =$$
$$= (\bar{a}\downarrow(\overline{\overline{e\cdot c}}))\downarrow(b\downarrow e)\downarrow(a\downarrow d\downarrow\bar{e}) = (\bar{a}\downarrow(\overline{\bar{e}+\bar{c}}))\downarrow(b\downarrow e)\downarrow(a\downarrow d\downarrow\bar{e}) =$$
$$= (\bar{a}\downarrow(\bar{e}\downarrow\bar{c}))\downarrow(b\downarrow e)\downarrow(a\downarrow d\downarrow\bar{e})$$

На основании полученного выражения построим структурную схему устройства управления на элементах ИЛИ-НЕ.

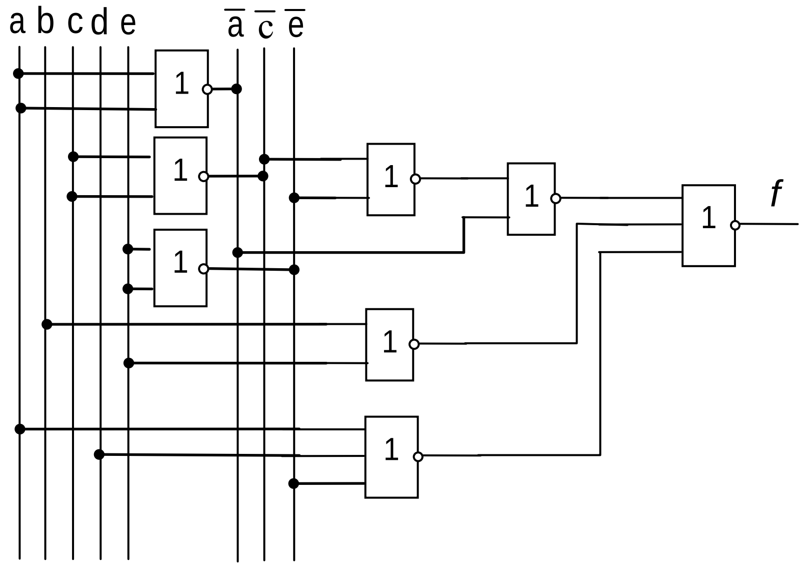


Рисунок 2.6 - Структурная схема устройства управления на элементах ИЛИ-НЕ

Структурная схема устройства управления на элементах И-НЕ содержит 6 логических элементов и 19 связей между ними. Структурная схема на элементах ИЛИ-НЕ содержит 8 логических элементов и 21 связь между ними. С точки зрения простоты реализации и надёжности целесообразно использовать структурную схему на элементах И-НЕ.

2.3 Минимизация системы уравнений, описывающей комбинационный автомат со многими входами

На практике чаще всего встречаются автоматы, имеющие m выходов и описывающиеся системой функций f_1, $f_2, \ldots f_m$. Таким образом, возникает задача оптимального синтеза всего автомата и, следовательно, совместной минимизации всех функций системы. Существует несколько подходов к решению этой задачи. Наиболее простой метод – применение многовыходной функции.

Порядок минимизации.

1. Проводится минимизация каждой функции отдельно $f_1^*, f_2^*, \ldots f_m^*$.
2. Составляется выражение для многовыходной функции

$$Ф = (f_1^* z_1 + f_2^* z_2 + \ldots + f_m^* z_m)$$

где z – буква ярлык, не носящая смысла логической переменной.

3. Раскрываются выражения всех функций, причём все слагаемые маркируются буквами ярлыками.

$$Ф = f_{11}^* z_1 + f_{12}^* z_1 + f_{13}^* z_1 + f_{21}^* z_2 + f_{22}^* z_2 + \ldots + f_{m1}^* z_m + \ldots$$

4. Выделяются общие части различных функций

$$f_{12}^* = f_{21}^*$$

и представляются в виде:

$$f_{12}^* z_1 + f_{21}^* z_2 = F_1(z_1 + z_2)$$

где: $z_1 + z_2$ означает только принадлежность F_1 и к первой и ко второй выходной функции. При выполнении данного пункта необходимо учитывать тип реализуемого автомата (контактный или бесконтактный).

Пример. Построить структурную схему устройства управления с четырьмя входами и двумя выходами, заданного следующими картами Карно

b

a

*x*1

1	0	1	1
0	0	0	0
0	1	0	0
0	0	0	0

d

c

b

a

*x*2

1	0	1	1
0	1	0	0
0	0	0	0
0	1	0	0

d

c

Запишем минимизированные выражения логических функций:

$$x_1 = \overline{a}\cdot\overline{c}\cdot\overline{d} + \overline{b}\cdot\overline{c}\cdot\overline{d} + a\cdot b\cdot c\cdot d =$$
$$= \overline{c}\cdot\overline{d}\cdot(\overline{a}\cdot+\overline{b}) + a\cdot b\cdot c\cdot d$$

$$x_2 = \overline{a}\cdot\overline{c}\cdot\overline{d} + \overline{b}\cdot\overline{c}\cdot\overline{d} + a\cdot b\cdot\overline{c}\cdot d +$$
$$+ a\cdot b\cdot c\cdot\overline{d} = \overline{c}\cdot\overline{d}\cdot(\overline{a}\cdot+\overline{b}) +$$
$$+ a\cdot b\cdot(\overline{c}\cdot d + c\cdot\overline{d})$$

Выполним совместную минимизацию полученных функций.

Запишем выражение для многовыходной функции.

$$F = x_1z_1 + x_2z_2 = [\overline{c}\cdot\overline{d}\cdot(\overline{a}\cdot+\overline{b}) + a\cdot b\cdot c\cdot d]z_1 + [\overline{c}\cdot\overline{d}\cdot(\overline{a}\cdot+\overline{b}) + a\cdot b\cdot(\overline{c}\cdot d + c\cdot\overline{d})]z_2 =$$
$$= \overline{c}\cdot\overline{d}\cdot(\overline{a}\cdot+\overline{b})z_1 + a\cdot b\cdot c\cdot d\ z_1 + \overline{c}\cdot\overline{d}\cdot(\overline{a}\cdot+\overline{b})z_2 + a\cdot b\cdot(\overline{c}\cdot d + c\cdot\overline{d})z_2 =$$
$$= \overline{c}\cdot\overline{d}\cdot(\overline{a}\cdot+\overline{b})(z_1 + z_2) + a\cdot b\cdot c\cdot d\ z_1 + a\cdot b\cdot(\overline{c}\cdot d + c\cdot\overline{d})z_2$$

На основании полученного выражения построим схему в бесконтактном варианте в базисе И, ИЛИ, НЕ.

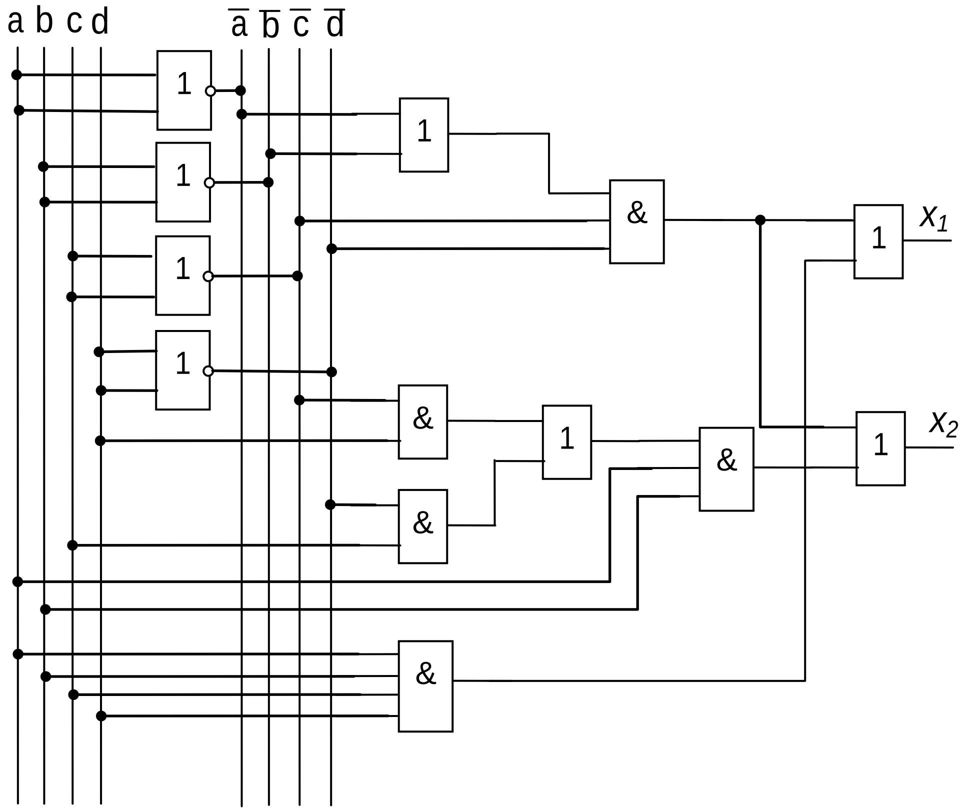


Рисунок 2.7 – Реализация устройства управления с двумя выходами на бесконтактных элементах И, ИЛИ, НЕ

Из схемы видно, что для бесконтактной реализации, как правило, целесообразно выделять общие части для различных выходов, связанные с этими выходами дизъюнктивно.

Рассмотрим контактный вариант реализации.

Пунктиром выделено неправильно (можно исправить, если при полярном питании вставить диоды).

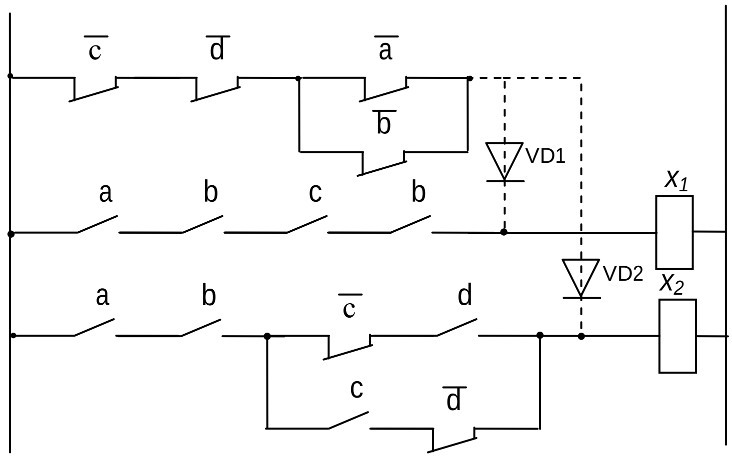


Рисунок 2.8 - Реализация устройства управления с двумя выходами на контактных элементах

Из схемы видно, что дизъюнктивно связанная общая часть при контактной реализации в общем случае не может быть вынесена. Эту часть можно реализовать только при полярном питании, используя диодную развязку. Конъюнктивно связанную общую часть здесь выделить невозможно.

Пример. Построить структурную схему устройства управления, заданного следующими картами Карно

b

a

x_1

0	1	0	0
0	0	0	0
0	1	0	0
0	0	0	0

d

c

b

a

x_2

0	0	0	0
0	1	0	0
0	0	0	0
0	1	0	0

d

c

Запишем минимизированные выражения логических функций:

$$x_1 = a \cdot b \cdot \overline{c} \cdot \overline{d} + a \cdot b \cdot c \cdot d$$

$$x_2 = a \cdot b \cdot c \cdot \overline{d} + a \cdot b \cdot \overline{c} \cdot d$$

Выражение для многовыходной функции:

$$F = (a \cdot b \cdot \overline{c} \cdot \overline{d} + a \cdot b \cdot c \cdot d) z_1 +$$
$$+ (a \cdot b \cdot c \cdot \overline{d} + a \cdot b \cdot \overline{c} \cdot d) z_2 = a \cdot b \cdot (\overline{c} \cdot \overline{d}\ z_1 +$$
$$+ c \cdot d\ z_1 + c \cdot \overline{d}\ z_2 + \overline{c} \cdot d\ z_2)$$

На основании полученного выражения можем составить контактную схему.

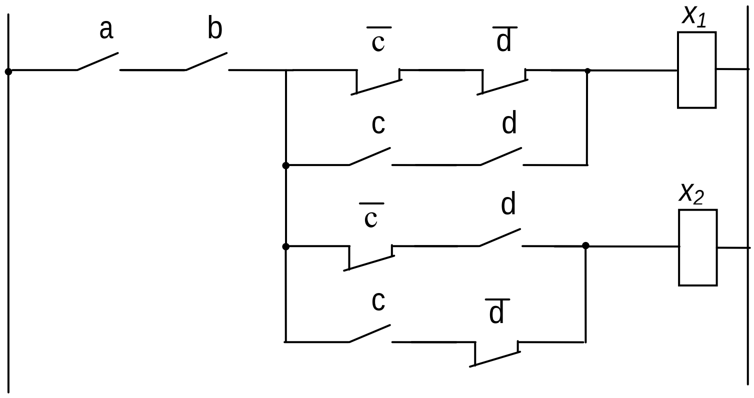


Рисунок 2.9 – Контактный вариант схемы управления

Таким образом, в данном варианте упрощение достигнуто за счёт конъюнктивно связанной общей части. Из рассмотренных примеров можно сделать следующий вывод.

При синтезе многовыходных функций при реализации для контактной схемы необходимо выделять конъюнктивно связанную общую часть, а при реализации бесконтактных схем – дизъюнктивно связанную общую часть.

При реализации в других базисах возможны выделения обоих частей для бесконтактных схем.

2.4 Применение диодов в контактных схемах

Применение диодов в контактных схемах с целью уменьшения контактов возможно при использовании полярных источников питания.

$$f = a*b + a*c + b*c = a*(b + c) + bc$$

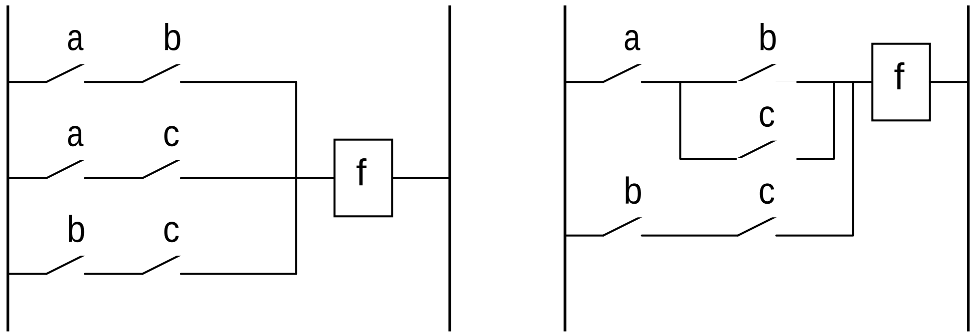


Рисунок 2.10 – Контактная схема устройства управления без применения диодов

Если использовать полярный источник питания и диоды, получим схему:

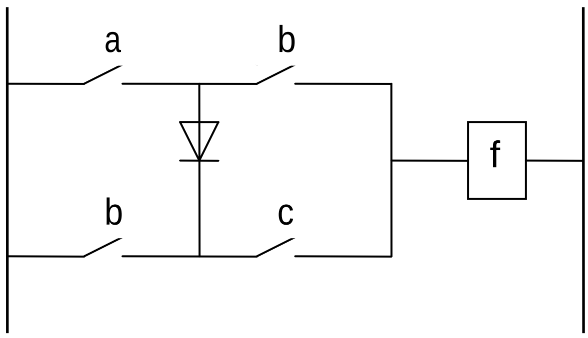


Рисунок 2.11 – Контактная схема устройства при полярном источнике питания с применением диодов

Применение диода позволяет уменьшить количество контактов, исключая возникновение ложных цепей за счёт односторонней проводимости.

3. ДИСКРЕТНЫЕ АВТОМАТЫ С ПАМЯТЬЮ

3.1 Общие понятия

В отличие от рассмотренных в предыдущей главе комбинационных автоматов, существует другой класс автоматов, в которых при одинаковых входных воздействиях на выходе автомата могут возникнуть различные выходные сигналы.

Состояние выхода такого автомата зависит не только от того, какие сигналы присутствуют на его входах в данный момент времени, но и от того, какие последовательности сигналов поступали на входы автомата в предшествующие моменты времени. Примером такого устройства может служить автомат, заданный следующей таблицей включения:

a	b	x
0	0	0
1	0	1
0	0	1

У данного устройства на одинаковых входных наборах на выходе формируются различные выходные сигналы.

Такие логические устройства должны иметь информацию о предыдущей работе схемы. Данная информация формируется устройством с помощью внутренних сигналов элементов памяти. Можно сказать, что автомат помнит свою предысторию и хранит её в памяти. Поэтому такие автоматы называют дискретными автоматами с памятью. Функциональная схема такого устройства представлена на рисунке 3.1.

На вход устройства поступают входные сигналы a_1 - a_n, значения которых образуют множество входных сигналов A. На выходе устройства формируются выходные сигналы $x_1 – x_m$, значения которых образуют множество выходных сигналов X. Для запоминания последовательности поступления входных сигналов используются внутренние сигналы элементов памяти $p_1 –$ p_k, образующие множество внутренних сигналов P.

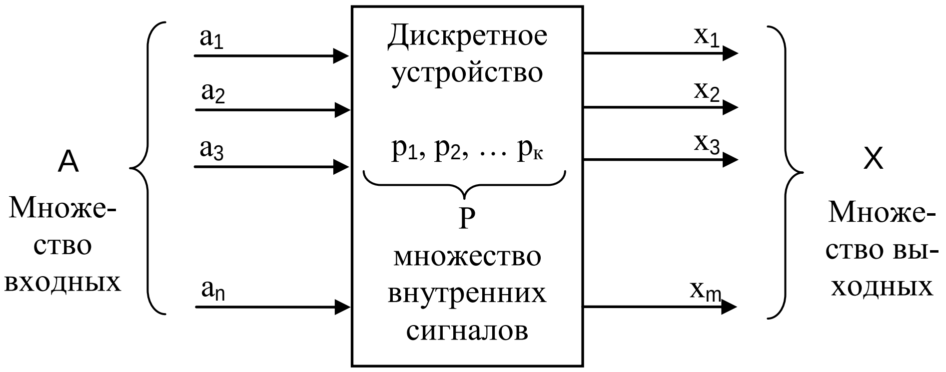


Рисунок 3.1 – Функциональная схема дискретного автомата с памятью.

Внутреннюю структуру дискретного автомата с памятью можно представить в виде совокупности комбинационной схемы и элементов памяти.

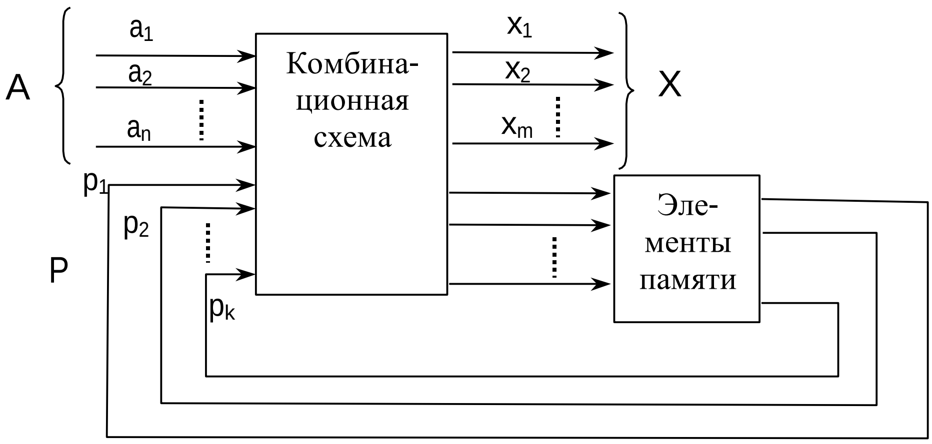


Рисунок 3.2 – Структурная схема дискретного автомата с памятью.

Входы дискретного автомата описывается входным алфавитом, буквами которого являются упорядоченные n - значные наборы входных сигналов.

Внутренние состояния дискретного автомата описываются алфавитом состояний, буквами которого являются упорядоченные k – значные наборы значений внутренних сигналов.

Выходы системы описываются алфавитом выхода, буквами которого являются упорядоченные m – значные наборы выходных переменных.

Из рисунка видно, что для описания дискретного автомата с памятью необходимо определить зависимость выходных сигналов от входных и внутренних сигналов, а так же зависимость сигналов, переключающих память от входных и внутренних.

Следовательно, в отличие от комбинационной схемы, работа дискретного автомата описывается двумя функциями – функцией выходов и функцией переходов.

Функция выходов – зависимость значений сигналов на каждом выходе от значения входных сигналов А и внутренних сигналов с элементов памяти Р в данный момент времени:

$$X_t = f(A_t; P_t)$$

Функция переходов – зависимость значений внутренних сигналов от значения входных сигналов в данный момент времени и внутренних сигналов автомата в предыдущий момент времени.

$$P_t = f(A_t; P_{t-1})$$

Следовательно, можно дать следующее определение дискретного автомата с памятью.

Дискретным автоматом (ДА) называется система с входным алфавитом А, алфавитом состояний Р, выходным алфавитом X и двумя характеристическими функциями.

Дискретный автомат – это математическая модель управляющего устройства.

Различают две основные модели дискретного автомата:

Дискретный автомат первого рода (автомат Мили)

$$X_t = f(A_t; P_t)$$

$$P_t = f(A_t; P_{t-1}).$$

Дискретный автомат второго рода (автомат Мура)

$$X_t = f(P_t)$$

$$P_t = f(A_t; P_{t-1}).$$

Основное различие между этими типами автоматов состоит в характере функции выхода; в первом типе автомата она зависит от состояния входа и внутреннего состояния, а во втором – только от внутреннего состояния.

Любое управляющее устройство может быть представлено обеими моделями. Модель Мили будет иметь более сложную комбинационную схему выходов, но меньше элементов памяти. У модели Мура – наоборот. Кроме этого часто используется совмещённая модель, когда часть выходных сигналов формируется по модели Мура, а часть – по модели Мили.

Работа дискретного автомата описывается функцией дискретного времени, единицей которого является такт.

Такт – это промежуток реального времени, в течение которого в схеме не происходят изменения. В зависимости от способа формирования единиц времени автоматы делятся на синхронные и асинхронные.

В асинхронном автомате начало и конец любого такта определяется изменением значения хотя бы одного входного или внутреннего сигнала.

В синхронном автомате интервалы времени формируются с помощью тактового генератора. При этом при каждом новом тактовом импульсе даже неизменная входная информация воспринимается как новая информация.

3.2 Способы задания дискретных автоматов

Способ задания автомата зависит от состава объектов, которые его определяют. В случае автомата с памятью – это конечные множества A, P, X, которые задаются простым перечислением их элементов, и функции переходов и выходов, которые являются формализацией словесного описания алгоритма функционирования автомата. На практике часто не удаётся перейти непосредственно от словесной формулировки условий работы автомата к его функциям переходов и выходов. В таких случаях применяют некоторый начальный язык задания автомата.

Начальные языки задания указывают связь только между входными и выходными сигналами. Функция переходов в этом случае явно не задаётся. К таким языкам можно отнести:

1. Линейные диаграммы
2. Первоначальные таблицы включения
3. Тактограммы
4. Логические схемы алфавитов.

Эти языки целесообразно использовать для начального задания дискретного автомата наряду со словесной формулировкой.

Языки, позволяющие задать функции переходов и выходов в виде однозначного отображения множества входных сигналов и внутренних сигналов в множество выходных сигналов и множество внутренних сигналов называются стандартными языками или автоматными языками.

Автоматные языки ставят в соответствие каждой паре вход-состояние пару выход-новое состояние:

$$A_i P_i \rightarrow X_i P_{i+1}.$$

Автоматные языки представлены в виде:

1. Таблиц выходов и переходов
2. Автоматных графов
3. Матриц переходов и выходов.

Рассмотрим подробнее автоматные языки задания.

I. <u>Задание автомата таблицами выходов и переходов.</u>

Таблица переходов – это таблица, строки которой соответствуют исходным состояниям, столбцы соответствуют входным наборам, в клетках на пересечении строк и столбцов проставляется состояние, в которое переходит автомат из данного исходного состояния, определяемого строкой, под действием данной входной последовательности, определяемой столбцом.

Общий вид таблицы переходов следующий:

	A1	A2	A2	…
P1	P1	P2	P3	
P2	P3	P1	P2	
P3	P2	P3	P1	
…				

где:

A1, A2, A3 … A_2^n – входные наборы;

P1, P2, P3 … $P_2^к$ – наборы значений внутренних переменных.

При начальном задании автомата вместо наборов значений внутренних переменных для обозначения внутренних состояний используются символьные обозначения (обычно используются цифры с 0).

Таблица выходов – это таблица, строки которой соответствуют исходным состояниям, столбцы соответствуют входным наборам, в клетках на пересечении строк и столбцов проставляется состояние выхода, формируемого в данном исходном состоянии под действием входной последовательности, определяемой столбцом.

Внешний вид таблицы выходов будет зависеть от выбранной модели описания дискретного автомата с памятью. Общий вид таблицы выходов для различных моделей представлен ниже.

Модель Мили

	A1	A2	A2	…
P1	X1	X3	X2	
P2	X3	X2	X1	
P3	X2	X1	X3	
…				

Модель Мура

	X
P1	X1
P2	X2
P3	X3

Часто таблицу переходов и выходов совмещают и пользуются совмещённой автоматной таблицей.

Для получения совмещённой таблицы при описании автомата по модели Мура к таблице переходов добавляют столбец выходов.

	A1	A2	A2	…	X
P1	P1	P2	P3		X1
P2	P3	P1	P2		X2
P3	P2	P3	P1		X3
…					

Для получения совмещённой автоматной таблицы для модели Мили необходимо в каждую клетку таблицы переходов, через запятую, дописать состояние выхода.

	A1	A2	A2	…
P1	P1, X1	P2, X3	P3, X2	
P2	P3, X3	P1, X2	P2, X1	
P3	P2, X2	P3, X1	P1, X3	
…				

Рассмотрим несколько примеров задания дискретного автомата с памятью.

Пример. Выполнить задание дискретного автомата с памятью, заданного следующей линейной диаграммой включения.

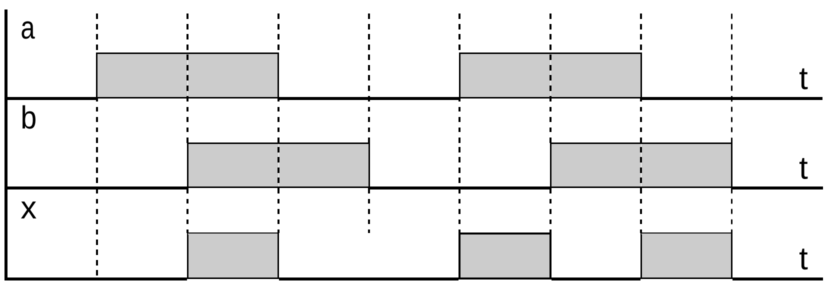


Рисунок 3.3 – Линейная диаграмма включения дискретного автомата

Для полного и точного описания устройства составим избыточную автоматную таблицу. В данной таблице любое изменение входного или выходного сигнала будет приводить к переходу в новое состояние. Следовательно, каждый такт работы устройства будет соответствовать определенному состоянию.

Так как для данного устройства последовательность смены сигналов задана и известна, следовательно, достаточно просто определить такты работы и количество внутренних состояний в автоматной таблице. Из линейной диаграммы видно, что тактов работы будет восемь. Строим таблицы переходов и выходов для разных моделей с восьмью внутренними состояниями.

Таблица переходов

	a		b	
0	1			0
1	1	2		
2		2	3	
3			3	4
4	5			4
5	5	6		
6		6	7	
7			7	0

Таблица выходов (модель Мили)

	a		b	
0	0			0
1	0	~		
2		1	~	
3			0	0
4	~			0
5	1	~		
6		0	~	
7			1	~

Таблица выходов (модель Мура)

	x
0	0
1	0
2	1
3	0
4	0
5	1
6	0
7	1

Построим совмещённые автоматные таблицы по модели Мура и модели Мили.

Совмещённые автоматные таблицы выходов и переходов

модель Мили

		b		
	a			
0	1,0			0,0
1	1,0	2,~		
2		2,1	3,~	
3			3,0	4,0
4	5,~			4,0
5	5,1	6,~		
6		6,0	7,~	
7			7,1	0,~

модель Мура

		b			x
	a				
0	1			0	0
1	1	2			0
2		2	3		1
3			3	4	0
4	5			4	0
5	5	6			1
6		6	7		0
7			8	0	1

Пример. Построить совмещённые автоматные таблицы по моделям Мили и Мура для устройства «Последовательный сумматор».

Данное устройство производит последовательное суммирование двух чисел в двоичном коде. Функциональная схема устройства представлена на рисунке 3.4

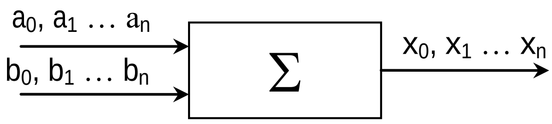


Рисунок 3.4 – Функциональная схема последовательного сумматора.

Принцип действия устройства рассмотрим на примере суммирования двух чисел.

		1	0	1	1
		0	1	1	0
	1	0	0	0	1
		↓	↓	↓	
память		1	1	1	

При суммировании 0 и 1 на выходе устройства формируется 1, в памяти сохраняется 0. При суммировании двух 1 на выходе устройства формируется 0, в память заносится 1, которая учитывается при суммировании следующих разрядов.

При работе данного устройства последовательность поступления сигналов на входе не задана. На вход может поступить любая из четырёх комбинаций входных переменных. Поэтому, при построении автоматной таблицы необходимо рассмотреть переход из исходного состояния в последующее при поступлении всех допустимых входных последовательностей.

Количество внутренних состояний автомата при описании по модели Мили будет определяться количеством комбинаций значений внутренних сигналов. В нашем случае это две комбинации (0 или 1). Совмещённая автоматная таблица последовательного сумматора по модели Мили будет иметь вид:

	b a			
0	0,1	1,0	0,1	0,0
1	1,0	1,1	1,0	0,1

Для модели Мура количество внутренних состояний автомата будет определяться количеством комбинаций значений внутренних и выходных сигналов. Для последовательного сумматора это четыре комбинации (00, 01, 10 и 11, где первая переменная – значение выхода, вторая – значение в памяти). Совмещённая автоматная таблица последовательного сумматора по модели Мура будет иметь вид:

		b a				x
память 0	0	1	2	1	0	0
память 0	1	1	2	1	0	1
память 1	2	2	3	2	1	0
память 1	3	2	3	2	1	1

II. <u>Задание автомата автоматными графами.</u>

То же самое задание автомата можно представить в виде автоматного графа. Вершины автоматного графа соответствуют состояниям автомата, дуги – переходам из состояния в состояние. Для дискретного автомата Мура у вершин графа проставляются номера состояний и соответствующие им выходные наборы, над дугами отмечаются входные наборы, под действием которых осуществляется данный переход (рисунок 3.5).

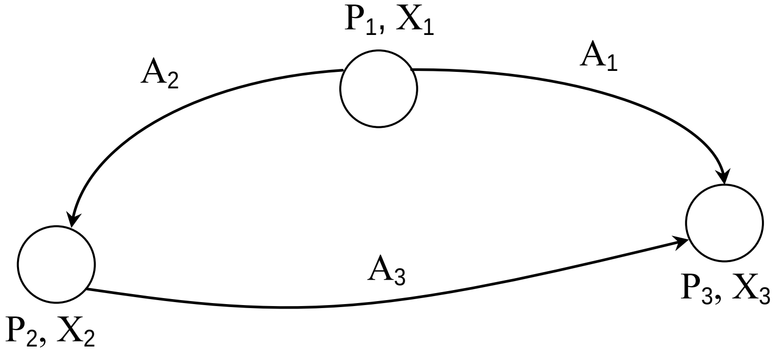


Рисунок 3.5 – Общий вид автоматного графа при описании по модели Мура

Для дискретного автомата Мили у вершин проставляются номера состояний, а над дугами проставляются входные наборы, под действием которых осуществился переход и выходы, формируемые при данном переходе (рисунок 3.6).

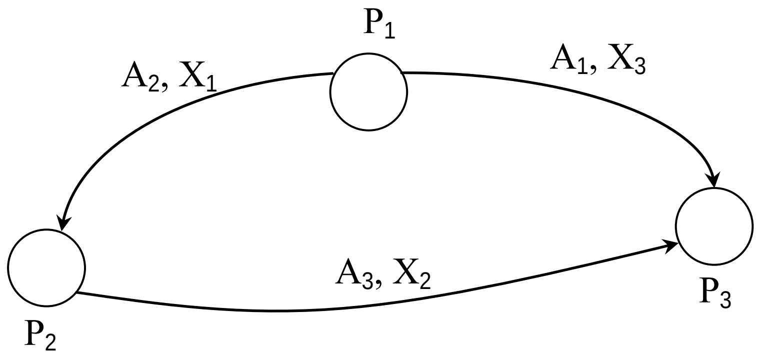


Рисунок 3.6 – Общий вид автоматного графа при описании по модели Мили

Рассмотрим пример построения автоматного графа.

Пример. Построить автоматный граф для устройства «Последовательный сумматор».

Построим автоматный граф по модели Мили. Так как в автоматной таблице для модели Мили два состояния, следовательно, автоматный граф будет содержать две вершины. Автоматный граф представлен на рисунке 3.7.

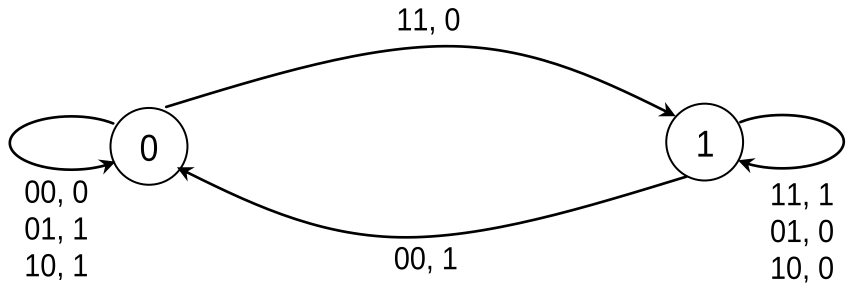


Рисунок 3.7 – Автоматный граф по модели Мили для устройства «Последовательный сумматор»

Автоматный граф по модели Мура будет содержать четыре вершины (по количеству состояний в автоматной таблице), и представлен на рисунке 3.8.

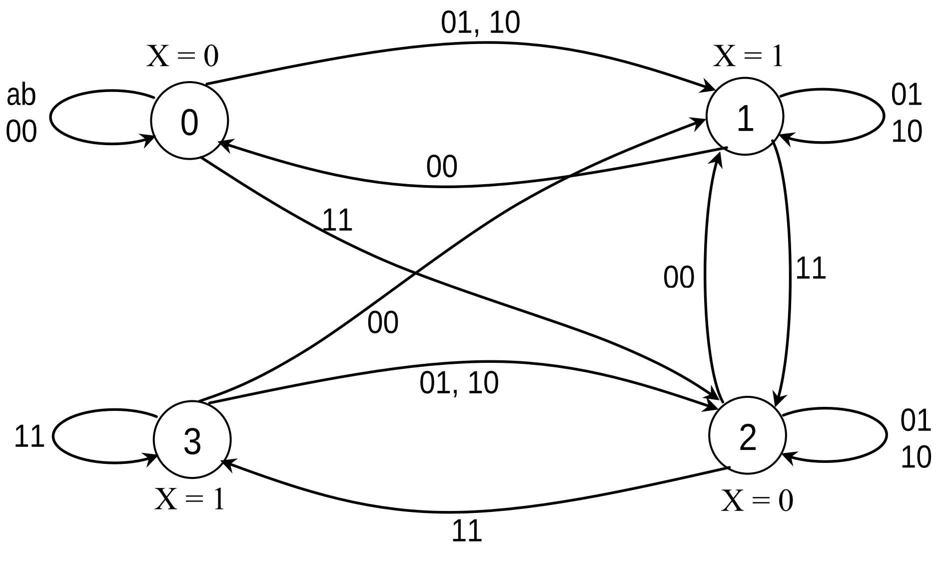


Рисунок 3.8 – Автоматный граф по модели Мура для устройства «Последовательный сумматор»

III. Задание дискретного автомата матрицей выходов и переходов.

При описании дискретного автомата матрицей переходов и выходов строки и столбцы матрицы соответствуют исходным состояниям. Элементами матрицы для модели Мура являются дизъюнктивно связанные входные наборы, под действием которых осуществляется переход из одного состояния в другое. Выходы при этом описываются матрицей-столбцом.

$$P = \begin{array}{cccc} & P_1 & P_2 & \dots & P_{2^k} \\ P_1 & A_2 & A_1 & \dots & \\ P_2 & A_2 + A_3 & A_1 & \dots & \\ \dots & \dots & \dots & \dots & \\ P_{2^k} & A_i & A_j & \dots & \end{array} \qquad X = \begin{array}{c|c|} P_1 & X_1 \\ P_2 & X_2 \\ \dots & \dots \\ P_{2^K} & X_{2^K} \end{array}$$

Для модели Мили элементами матрицы являются пары вход-выход. В случае необходимости пары связываются дизъюнктивно.

$$P = \begin{array}{ccccc} & P_1 & P_2 & \dots & P_{2^k} \\ P_1 & \frac{A_2}{X_1} & \frac{A_1}{X_2} & \dots & \\ P_2 & \frac{A_2}{X_3} + \frac{A_3}{X_2} & \frac{A_1}{X_1} & \dots & \\ \dots & \dots & \dots & \dots & \\ P_{2^k} & \frac{A_i}{X_J} & \frac{A_j}{X_I} & \dots & \end{array}$$

Матрицы переходов обладают меньшей наглядностью, чем автоматные графы. Однако они дают возможность формализовать ряд операций по преобразованию автоматов. Матрицы удобны при осуществлении автоматического синтеза автоматов с помощью ЭВМ.

Пример. Описать дискретный автомат «Последовательный сумматор» с помощью матриц выходов и переходов.

Матрица переходов для модели Мили:

$$P = \begin{vmatrix} \frac{\bar{a} \cdot \bar{b}}{0} + \frac{\bar{a} \cdot b}{1} + \frac{a \cdot \bar{b}}{1} & \frac{a \cdot b}{0} \\ \frac{\bar{a} \cdot \bar{b}}{1} & \frac{\bar{a} \cdot b}{0} + \frac{a \cdot \bar{b}}{0} + \frac{a \cdot b}{1} \end{vmatrix}.$$

Для модели Мура:

$$P = \begin{vmatrix} \bar{a}\cdot\bar{b} & a\cdot\bar{b}+\bar{a}\cdot b & a\cdot b & \sim \\ \bar{a}\cdot\bar{b} & a\cdot\bar{b}+\bar{a}\cdot b & a\cdot b & \sim \\ \sim & \bar{a}\cdot\bar{b} & a\cdot\bar{b}+\bar{a}\cdot b & a\cdot b \\ \sim & \bar{a}\cdot\bar{b} & a\cdot\bar{b}+\bar{a}\cdot b & a\cdot b \end{vmatrix} \times \begin{vmatrix} 0 \\ 1 \\ 0 \\ 1 \end{vmatrix}.$$

Из рассмотренных примеров видно, что любой дискретный автомат с памятью может быть представлен как моделью Мили, так и моделью Мура и возможен переход от одной модели к другой. Для перехода от модели Мили к модели Мура необходимо:

1. Каждой паре состояние-выход в таблице автомата Мили присвоить новый номер состояния автомата Мура
2. Составить таблицу автомата Мура, заменяя пары состояние-выход на новые номера состояний автомата Мура
3. Ввести в таблицу все полученные новые состояния и приписать каждому состоянию соответствующий выход.

Для перехода от автомата Мура к автомату Мили необходимо:

1. Каждому номеру состояния в таблице приписать через запятую соответствующий этому состоянию выход
2. Убрать столбец выходов
3. Заменить одинаковые строки одной строкой, присвоив совмещаемым состояниям новый номер состояния автомата Мили.

3.3 **Полностью и не полностью определённые автоматы**

Рассмотренные способы задания автомата (таблицами, графами и матрицами) полностью описывали автомат и были в этом отношении эквивалентны. При этом, задавая автомат, мы нигде не оговаривали никаких ограничений условий его функционирования. Однако в общем случае такие условия существуют, и с ними следует считаться.

Входная последовательность, удовлетворяющая каким-либо дополнительным условиям, связанным с функционированием автомата, называется допустимой.

Совокупность входных последовательностей может быть ограничена по условиям работы автомата. Кроме того ограничения на входные последовательности могут быть связаны с внутренним состоянием автомата. На практике возможны случаи, когда для автомата, находящегося в момент времени t в состоянии Pi, следующее его состояние $Pj(t + 1)$ и состояние выхода $X(t)$ либо хотя бы одна из этих компонент определена не для всякого входного воздействия Ak. В таких случаях говорят, что для данного автомата входное воздействие Ak в состоянии Pi запрещено.

Если на множество допустимых входных последовательностей не наложено никаких ограничений и, следовательно, любая входная последовательность является для автомата допустимой, то такой автомат называется полностью определённым.

Если хотя бы для одного состояния автомата на допустимую входную последовательность наложено ограничение, он называется неполностью определённым.

Для обозначения неопределённых внутренних состояний и неопределённых состояний выхода используется знак тильда.

3.4 Постановка задачи синтеза дискретных автоматов с памятью

Синтез управляющего устройства в общем случае состоит из трёх этапов:

1. Этап предварительного или блочного синтеза

   На этом этапе, на основании словесной формулировки или другого предварительного способа задания сложного устройства выполняется разбиение этого устройства на блоки, и намечаются информационные связи между блоками.

2. Абстрактный синтез автомата

2.1. Составление словесной формулировки условий работы блока или устройства. Составление первоначальной таблицы включения, линейной диаграммы или соответствующего другого описания на начальном языке описания работы автомата.

Задачи первого этапа синтеза исключительно важны, поскольку при выполнении данного этапа необходимо решить две задачи:

1) Можно ли построить автомат с конечным числом входов, выходов и внутренних состояний, который выполнял бы поставленные задачи.

На сегодняшний день данная задача окончательно не решена, т. е. не определены окончательные и необходимые условия, при выполнении которых по заданным условиям можно построить автомат.

2) Выбор способа формулировки условий работы, т. е. языка задания автомата.

На первый взгляд, достаточно перечислить все входные и соответствующие им выходные последовательности. Это просто сделать, если последовательности конечны. Однако на практике иногда нельзя заранее установить длительность работы проектируемого устройства, следовательно, необходимо рассчитывать работу устройства на бесконечную входную последовательность. Здесь уже применить простое перечисление не возможно.

2.2. Синтез автомата на автоматном языке

В случае синтеза на автоматном языке осуществляется переход от словесной формулировки и задания автомата на начальном языке к автоматному описанию. При этом строится избыточная автоматная таблица частичного автомата. В этой автоматной таблице каждое состояние автомата соответствует такту, т. е. в каждой строке таблицы заполнены только две клетки – приход в данное состояние и выход из данного состояния.

Результатом второго этапа абстрактного синтеза является задание автомата одним из стандартных способов, например в виде таблиц переходов и выходов.

2.3. Минимизация памяти автомата

Этот этап проводится уже с меньшей степенью абстракции. Здесь автомат рассматривается состоящим из двух частей: логики и памяти (рисунок 3.9).

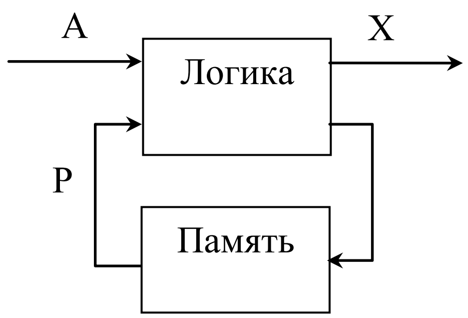


Рисунок 3.9 – Структура автомата при минимизации

Объём памяти связан с числом внутренних состояний автомата. При уменьшении числа внутренних состояний автомата уменьшается количество элементов памяти, но при этом полученный автомат должен быть эквивалентен исходному автомату.

Автомат A называется эквивалентным автомату A', если для каждого состояния автомата A найдётся по крайней мере одно эквивалентное состояние автомата A'.

Состояния pi и pj эквивалентны, если автомат, находясь в этих состояниях, под действием любой входной последовательности вырабатывает одинаковые выходные последовательности.

Таким образом, минимизация заключается в замене исходного автомата эквивалентным автоматом с наименьшим числом состояний.

Результатом третьего этапа абстрактного синтеза является таблица переходов и выходов автомата, выполняющего поставленные условия работы и имеющего вместе с тем наименьшее возможное число внутренних состояний.

Полученная на втором этапе абстрактного синтеза минимизированная таблица переходов и выходов служит исходной для третьего этапа синтеза, который характеризуется дальнейшим снижением уровня абстракции.

3. Структурный синтез автомата

Абстрактный автомат заменяется структурным автоматом и производится его компоновка из более простых автоматов.

3.1. Кодирование состояний входа, выхода и внутренних состояний.

Осуществляется переход от символов (букв) входа, выхода, состояния к сигналам, соответствующим этим символам. При этом каждому внутреннему состоянию автомата присваивается свой, индивидуальный неповторяющийся код.

При кодировании необходимо решить задачу упрощения структурной схемы автомата и задачу устранения критических состязаний элементов памяти.

В результате кодирования от таблицы переходов, которая давала лишь общую картину функционирования автомата, не раскрывая его внутренней структуры, осуществляется переход к таблице состояний. В данной таблице полностью отражается зависимость между воздействиями на внешних входах и входах, по которым поступает информация с элементов памяти, и сигналами на внешних выходах и тех выходах, с которых поступает информация в память автомата. Это позволяет перейти к синтезу комбинационного автоматного эквивалента.

3.2. Синтез комбинационной части автомата.

На этом этапе записываются аналитические выражения для выходных сигналов и внутренних сигналов.

3.3. Преобразование полученных выражений в заданный базис.

3.4. Построение по полученным выражениям структурной схемы.

3.5. Переход от структурной схемы к принципиальной схеме.

При этом учитывается питание логических элементов, согласуются входные сигналы системы и элементов, учитывается нагрузочная способность элементов, предусматривается подавление помех.

3.6. Анализ правильности функционирования полученной схемы устройства управления и, в случае необходимости, преобразование её с целью обеспечения надёжности функционирования.

3.5 Минимизация памяти автомата

Минимизация заключается в нахождении совместимых состояний в автоматной таблице и замене совместимых состояний одним состоянием нового эквивалентного автомата.

Совместимыми называются состояния pi и pj, для которых справедлива следующая закономерность: не зависимо от того, в каком из этих состояний находится автомат, на любую допустимую входную последовательность он формирует одинаковые выходные последовательности.

Группа попарно совместимых состояний образует группу совместимости.

Группы совместимости, покрывающие все состояния исходного автомата, образуют класс совместимости.

Таким образом, отношение совместимости представляет собой такое разбиение исходных состояний, блоками которого являются группы совместимости.

Для получения минимального автомата необходимо построить минимальный класс совместимости, т. е. такой, который содержит минимум групп совместимости, группы максимальны по размеру и данное разбиение покрывает все состояния исходного автомата. Поиск совместимых состояний сводится к поиску совместимых строк автоматной таблицы и замене их одной строкой.

При поиске совместимых состояний в автоматной таблице используют два признака:

1) Из обоих состояний под действием любой допустимой входной последовательности осуществляется переходы в совместимые или непротиворечивые состояния.
2) Этим состояниям должны соответствовать непротиворечивые или совместимые выходы.

При этом, в зависимости от того, как накладываются переходы, два состояния могут быть условно совместимыми.

Два состояния называются условно совместимыми, если они совместимы при условии совместимости состояний, в которые автомат переходит из этих состояний.

Из множества методов минимизации наибольшее применение находит метод Ангера и Пола на основе треугольной таблицы совместимости.

<u>Порядок минимизации:</u>

1. Строится треугольная таблица, в которой различаются совместимые, несовместимые и условно-совместимые пары состояний.
2. Просматривается таблица, и формируются максимальные группы совместимости.
3. Строится таблица покрытия и определяется минимальный класс совместимости.
4. В случае необходимости выполняется редукция классов совместимости, т. е. из групп совместимости удаляются повторяющиеся состояния.
5. Строится минимизированная таблица эквивалентного автомата.

Рассмотрим примеры минимизации различных устройств.

Пример. Выполнить минимизацию устройства «Генератор запрета».

Функциональная схема устройства и линейная диаграмма включения представлены на рисунке 3. 10.

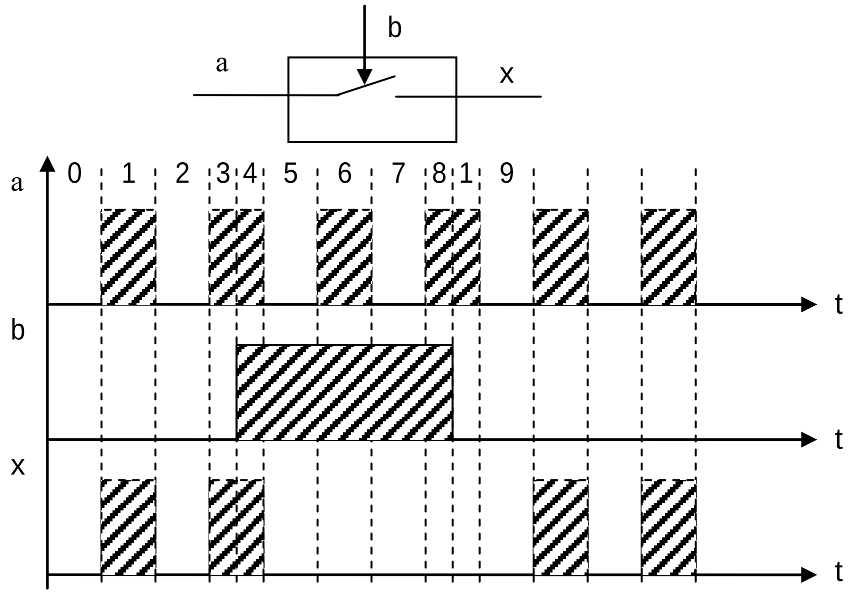


Рисунок 3.10 – Линейная диаграмма включения «Генератора запрета»

Необходимо, чтобы сигнал на выходе x появлялся при наличии сигнала a и при отсутствии сигнала запрета b. При этом импульсы на выходе не должны искажаться.

Составим первоначальную автоматную таблицу выходов и переходов асинхронного автомата по модели Мура.

	a		b		x
0	1			0	0
1	1			2	1
2	3			2	0
3	3	4			1
4		4	5		1
5		6	5		0
6		6	7		0
7		8	7		0
8	9	8			0
9	9			0	0

1. Составим треугольную таблицу. В данной таблице по горизонтали проставляются состояния от нулевого до предпоследнего, по вертикали – от последнего до первого (исключая нулевое состояние). Каждая клетка треугольной таблицы соответствует паре состояний. По автоматной таблице просматриваем все пары состояний в порядке, указанным в треугольной таблице справа на лево, снизу вверх и отмечаем совместимые, несовместимые и условно совместимые состояния.

1	X								
2	1,3 V	X							
3	X	V	X						
4	X	V	X	V					
5	V	X	V	X	X				
6	V	X	V	X	X	5,7 V			
7	V	X	V	X	X	6,8 V	6,8 V		
8	1,9 X	X	3,9 X	X	X	6,8 V	V	V	
9	1,9 X	X	3,9 2,0 X	X	X	V	V	V	V
	0	1	2	3	4	5	6	7	8

X – несовместимое состояние
V – совместимое состояние
3,9 – условно совместимое состояние

2. По треугольной таблице выписываем группы совместимости. Для этого просматриваем треугольную таблицу и объединяем в группы попарно совместимые между собой состояния.

- (8,9);
- (7,8,9);
- (6,7,8,9);
- (5,6,7,8,9);
- (5,6,7,8,9) (3,4);
- (2,5,6,7) (5,6,7,8,9) (3,4);
- (2,5,6,7) (5,6,7,8,9) (1,3,4);
- (0,2,5,6,7) (5,6,7,8,9) (1,3,4);

A B C

3. Строим таблицу покрытия и определяем минимальный класс совместимости. Строки таблицы покрытия соответствуют группам совместимости, столбцы – исходным состояниям. На пересечении строк (групп совместимости) со столбцами (исходными состояниями) отмечается вхождение данного состояния в данную группу совместимости.

	0	1	2	3	4	5	6	7	8	9
0,2	X		X							
5,6,7,8,9						X	X	X	X	X
1,3,4		X		X	X					

На основании таблицы покрытия составляется формула покрытия. Для этого для каждого состояния записываем дизъюнкцию тех групп совместимости, в которые входит данное состояние. Дизъюнкции для каждого состояния связываются конъюнктивно. К полученной формуле применяются законы алгебры логики, раскрываются все скобки и получают в результате дизъюнктивно связанные классы совместимости. Из полученных классов необходимо выбрать тот, который содержит наименьшее количество групп совместимости. Это и будет минимальный класс совместимости.

$$F = A*C*A*C*C*B*B*B*B*B = A*B*C;$$

В нашем случае имеем один класс совместимости, он же минимальный.

4. Выполним редукцию групп совместимости.

Состояния 5,6,7 входят в две группы совместимости. Необходимо удалить эти состояния из одной из групп, оставив в другой. При удалении этих состояний получившиеся групп совместимости должны быть такими, чтобы условия совместимости для них не нарушались.

Удалим состояния 5,6,7 из группы совместимости А. Проверяем, не нарушились ли условия совместимости в полученных группах. Для всех состояний условия совместимости выполняются. Таким образом, получим редуцированный класс совместимости:

(0,2) (5,6,7,8,9) (1,3,4)

Если удалить повторяющиеся состояния из группы B, то группа A становится несовместимой, так как нарушаются условия совместимости (условие 6,8 для пар состояний 5,6 5,7).

5. Строим минимизированную автоматную таблицу.

Для этого каждой группе совместимости присваиваем новый номер состояния эквивалентного автомата:

0,2 – 0

1,3,4 – 1

5,6,7,8,9 – 2

В исходной таблице старые номера заменяем на новые и строки с одинаковыми номерами объединяем в одну строку путём наложения.

<table>
<tr><td rowspan="2"></td><td></td><td colspan="2">b</td><td></td><td rowspan="2">x</td></tr>
<tr><td colspan="2">a</td><td></td><td></td></tr>
<tr><td>0</td><td>1</td><td></td><td></td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td></td><td></td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td></td><td></td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td></td><td></td><td>1</td></tr>
<tr><td>1</td><td></td><td>1</td><td>2</td><td></td><td>1</td></tr>
<tr><td>2</td><td></td><td>2</td><td>2</td><td></td><td>0</td></tr>
<tr><td>2</td><td></td><td>2</td><td>2</td><td></td><td>0</td></tr>
<tr><td>2</td><td></td><td>2</td><td>2</td><td></td><td>0</td></tr>
<tr><td>2</td><td>2</td><td>2</td><td></td><td></td><td>0</td></tr>
<tr><td>2</td><td>2</td><td></td><td></td><td>0</td><td>0</td></tr>
</table>

<table>
<tr><td rowspan="2"></td><td></td><td colspan="2">b</td><td></td><td rowspan="2">x</td></tr>
<tr><td colspan="2">a</td><td></td><td></td></tr>
<tr><td>0</td><td>1</td><td></td><td></td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>2</td><td>0</td><td>1</td></tr>
<tr><td>2</td><td>2</td><td>2</td><td>2</td><td>0</td><td>0</td></tr>
</table>

Выполним минимизацию данного устройства при его описании по модели Мили. Составим автоматную таблицу по модели Мили. По автоматной таблице просматриваем все состояния и заполняем треугольную таблицу.

	a		b	
0	1,~			0,0
1	1,1			2,~
2	3,~			2,0
3	3,1	4,1		
4		4,1	5,~	
5		6,0	5,0	
6		6,0	7,0	
7		8,0	7,0	
8	9,0	8,0		
9	9,o			0,0

1	V 0,2								
2	V 1,3	V 1,3							
3	V 1,3	V	V						
4	V	V	V	V					
5	V	V	V	X	X				
6	V	V	V	X	X	V 5,7			
7	V	V	V	X	X	V 6,8	V 6,8		
8	X 1,9	X	X 3,9	X	X	V 6,8	V	V	
9	X 1,9	X	X 3,9 2,0	X	V	V	V	V	V
	0	1	2	3	4	5	6	7	8

Записываем максимальные группы совместимости:

5,6,7,8,9 – A

4,9 - B

0,1,2,3,4 - C

0,1,2,5,6,7 - D

Строим таблицу покрытия и определяем минимальный класс совместимости.

	0	1	2	3	4	5	6	7	8	9
A						X	X	X	X	X
B					X					X
C	X	X	X	X	X					
D	X	X	X			X	X	X		

Записываем формулу покрытия.

Ф = (A+B)*A*(A+D)* (A+D)* (A+D)*(B+C)*C*(C+D) *(C+D) *(C+D) = A*C

Обозначаем новые состояния и строим минимизированную автоматную таблицу.

5,6,7,8,9 – 1

0,1,2,3,4 - 0

	b a			
0	0,~			0,0
0	0,1			0,~
0	0,~			0,0
0	0,1	0,1		
0		0,1	1,~	
1		1,0	1,0	
1		1,0	1,0	
1		1,0	1,0	
1	1,0	1,0		
1	1,0			0,0

	b a			
0	0,1	0,1	1,~	0,0
1	1,0	1,0	1,0	0,0

Пример. Выполнить описание и минимизацию устройства, заданного линейной диаграммой работы следующего вида.

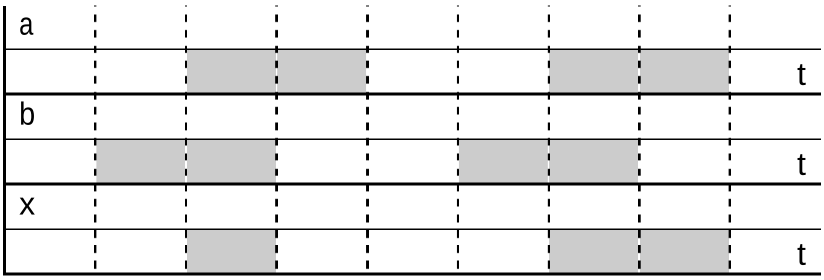


Рисунок 3.11 – Линейная диаграмма работы устройства управления

Составляем автоматную таблицу и заполняем треугольную таблицу.

Автоматная таблица

	b a				x
0			1	0	0
1		2	1		0
2	3	2			1
3	3			4	0
4			5	4	0
5		6	5		0
6	7	6			1
7	7			0	1

Треугольная таблица

1	V						
2	X	X					
3	X	V	X				
4	X	X	X	V			
5	X	X	X	V	V		
6	X	X	X	X	X	X	
7	X	X	X	X	X	X	V
	0	1	2	3	4	5	6

Группы совместимости

A – 6,7

B – 3,4,5

C – 1,3

D – 0,1

Строим таблицу покрытия и определяем минимальный класс совместимости.

	0	1	2	3	4	5	6	7
A							X	X
B				X	X	X		
C		X		X				
D	X	X						

Формула покрытия:

Ф = A*A*B*B*(B+C)*(C+D)*D = A*B*D

Из таблицы покрытия видно, что состояние 2 не вошло ни в одну из групп совместимости. Следовательно, данное состояние будет образовывать отдельную группу совместимости, состоящую из одного состояния. Обозначаем новые состояния и строим минимизированную автоматную таблицу.

6,7 - 3

3,4,5 - 2

0,1 - 0

2 - 1

Минимизированная таблица.

		b	b		x
	a	a			
0		1	0	0	0
1	2	1			1
2	2	3	2	2	0
3	3	3		0	1

3.6 **Кодирование внутренних состояний дискретного автомата**

При кодировании каждому внутреннему состоянию автомата приписывается свой, неповторяющийся набор внутренних сигналов элементов памяти (определяется код состояния).

В зависимости от значения данной переменной кодирования все внутренние состояния автомата разделяются на два блока. В первый блок входят состояния, в которых данная переменная кодирования равна единице, во второй блок – состояния в которых переменная кодирования равна нулю.

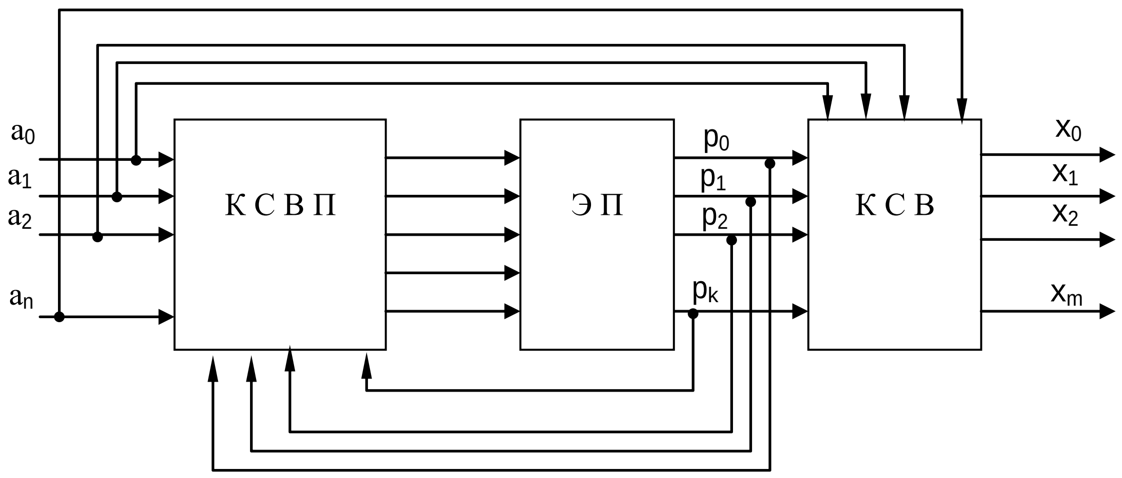
Разбиением множества внутренних состояний P называют совокупность подмножеств состояний Pi таких, что их объединение есть Р. Подмножества называются блоками разбиения и отмечаются чертой.

Произведением двух разбиений $Пi$ и $Пj$ называют такое третье разбиение $Пij$, каждый блок которого образован пересечением блоков разбиений $Пi$ и $Пj$.

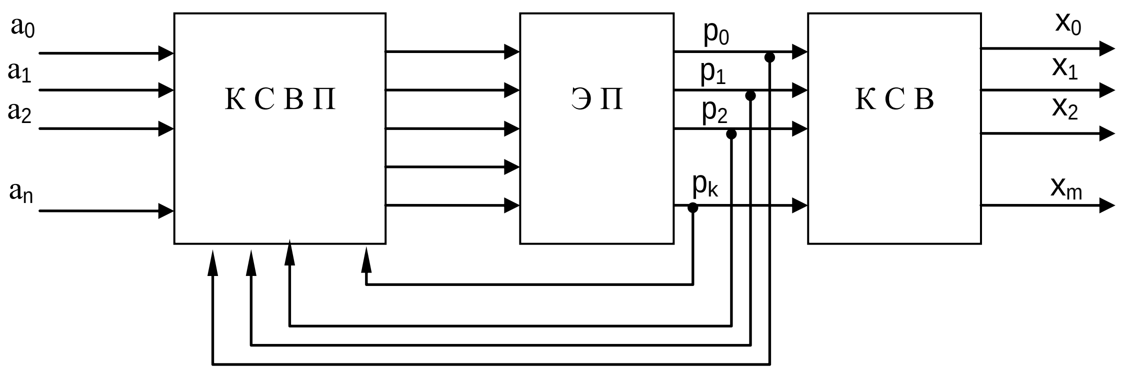
Для однозначного кодирования необходимо, чтобы пересечение разбиений, в соответствии с которыми определяется распределение переменных кодирования, было бы равно нулевому разбиению.

В нулевом разбиении каждое состояние образует один блок, и, следовательно, все состояния между собой различимы.

При кодировании дискретный автомат с памятью представляется состоящим из элементов памяти ЭП, комбинационной схемы возбуждения памяти (КСВП) и комбинационной схемы выходов (КСВ) (рисунок 3.12). Комбинационная схема возбуждения памяти КСВП формирует сигналы, переключающие элементы памяти и реализует функцию переходов. Комбинационная схема выходов КСВ формирует выходные сигналы и реализует функцию выходов.



а)



б)

Рисунок 3.12 – Функциональные схемы автомата с памятью при кодировании внутренних состояний: а – при описании по модели Мили; б – при описании по модели Мура

Возможно получение множества вариантов однозначного кодирования состояний автомата. Эти варианты приведут к различным схемам автомата по сложности комбинационных схем памяти и выходов, а так же для асинхронных автоматов могут приводить к состязанию элементов памяти.

Поэтому при кодировании внутренних состояний автомата необходимо решать следующие задачи:

1) Упрощение комбинационной схемы выходов.
2) Упрощение комбинационной схемы возбуждения памяти.

3) Исключение критических состязаний элементов памяти.
4) Обеспечение заданного быстродействия.

Задача исключения критических состояний элементов памяти стоит для асинхронных автоматов.

Данные задачи решаются следующим образом.

Для получения простой комбинационной схемы выходов необходимо выполнить кодирование по внешнему разбиению. При кодировании состояний по внешнему разбиению значения переменных кодирования в соответствующих состояниях приравниваются к значениям выходных переменных в этих же состояниях.

$$p_0 = x_0;$$
$$p_1 = x_1;$$
$$p_2 = x_2;$$
$$.....$$

При этом, чем больше совпадают значения переменных кодирования с выходными сигналами, тем проще комбинационная схема выходов. В случае если $k = m$ и все внутренние состояния имеют различные коды, получаем полное совпадение переменных кодирования с выходными переменными и комбинационная схема выходов становится не нужна.

Для упрощения комбинационной схемы возбуждения памяти используют кодирование по внутреннему разбиению. При кодировании по внутреннему разбиению в один блок включаются наиболее близкие по переходам состояния. Чем ближе по переходам состояния, включаемые в один блок, тем проще схема возбуждения памяти.

Порядок кодирования внутренних состояний автомата.

1) Определение значений переменных кодирования по внешнему разбиению.
2) Если по внешнему разбиению остались неразличимые состояния, построение внутреннего разбиения.

3) Присвоение кодов состояниям и проверка однозначности кодирования.
4) Построение автоматного графа с целью определения и устранения критических состязаний элементов памяти.
5) Построение исправленной минимизированной закодированной автоматной таблицы.

Рассмотрим примеры кодирования внутренних состояний автомата.

Пример. Выполнить кодирование внутренних состояний устройства «Генератор запрета».

Минимизированная автоматная таблица данного устройства имеет следующий вид.

	a	b			x
0	1			0	0
1	1	1	2	0	1
2	2	2	2	0	0

Автомат имеет три внутренних состояния. Для кодирования всех состояний необходимо иметь три различных кода, следовательно, необходимо две переменных кодирования.

Значения первой переменной кодирования определим по внешнему разбиению.

$$p_1 = x$$

$$\pi_{1x} = \overline{0,2} \qquad \overline{1}$$

Теперь состояние 1 по коду различимо от состояний 0 и 2, а состояния 0 и 2 между собой неразличимы. Второе разбиение должно быть таким, чтобы различались состояния 0 и 2. Для этого состояния 0 и 2 должны находиться в разных блоках разбиения. Можем записать следующие варианты:

$$\pi_{21} = \overline{0,\ 1} \qquad \overline{2}$$

$$\pi_{22} = \overline{1,\ 2} \qquad \overline{0}$$

Как видим, даже в таком простом примере возникают два варианта второго разбиения. Нам необходимо выбрать такое второе разбиение, которое упрощало бы схему. Если первое разбиение упрощает комбинационную схему выходов, то второе разбиение можно выбрать таким образом, чтобы упрощалась комбинационная схема возбуждения памяти.

Для этого определим значения второй переменной кодирования по внутреннему разбиению. Воспользуемся методом на основе множеств порядка "1".

Множеством порядка единица для состояния *pi* называется множество тех состояний, из которых возможен непосредственный переход в состояние *pi*.

Порядок определения внутреннего разбиения.

1) Для каждого состояния выписываем множество порядка единица.

Выписываем номера состояний, из которых переходим в данное состояние.

2) Выписываем объединённые наборы таких состояний, которые имеют максимальное пересечение множеств порядка единица.
3) Из объединённых наборов составляем двухблочное разбиение таким образом, чтобы это разбиение содержало в разных блоках состояния, не различимые по внешнему разбиению.
4) Проверяем, равно ли пересечение полученных разбиений нулевому разбиению

Определим внутреннее разбиение для рассматриваемого устройства.

Множества порядка «1».

0 - 0, 1, 2

1 – 0, 1

2 – 1, 2.

Объединённые наборы. Множества порядка «1» максимально пересекаются по двум состояниям.

$$\overline{0,\ 1} \quad \overline{0,\ 2}$$

Строим второе разбиение. Так как объединённый набор $\overline{0,\ 2}$ используется в первом разбиении, то второе разбиение строим с использованием набора $\overline{0,\ 1}$. Получим разбиение следующего вида

$$\pi_2 = \overline{0,\ 1} \qquad \overline{2}$$

Проверяем однозначность кодирования.

$\pi_1 \cdot \pi_2 = (\overline{0,\ 2} \quad \overline{1}) \cdot (\overline{0,\ 1} \quad \overline{2}) = \overline{0} \quad \overline{2} \quad \overline{1}$ - равно нулевому разбиению.

Распределяем переменную p_2 по состояниям в соответствии с разбиением π_2 (в 0 и 1 состояниях p_2 равна 0, во 2 состоянии - 1). Получаем минимизированную закодированную автоматную таблицу.

	b a				x	p1	p2
0	1			0	0	0	0
1	1	1	2	0	1	1	0
2	2	2	2	0	0	0	1

Полученную минимизированную закодированную автоматную таблицу необходимо проанализировать с точки зрения возможности возникновения состязаний элементов памяти при переходах.

Состязания элементов памяти возникают в том случае, если при переходе из состояния в состояние меняют своё значение два и более разряда кода. При этом, за счёт наличия задержек на срабатывание элементов памяти один разряд кода может переключиться быстрее. Это приводит к появлению промежуточных кодов, которые могут привести к нарушению алгоритма работы автомата.

Состязания элементов памяти могут быть критическими и некритическими.

Критические – это состязания, при которых нарушается алгоритм работы устройства (автомат застревает в непредусмотренном состоянии или осуществляются непредусмотренные переходы с формированием другой выходной последовательности).

Некритические – это состязания, при которых автомат переходит в другое используемое состояние или в неиспользуемое состояние, но при этом алгоритм работы не нарушается, формируется заданная выходная последовательность.

В асинхронных автоматах обычно стараются исключить состязания и критические и некритические.

Для устранения критических состязаний элементов памяти выполняют следующие действия.

1) Выполняют вместо заданного перехода переход через другое неустойчивое состояние с последовательной сменой кода.
2) Если такой переход выполнить невозможно, вводят дополнительные состояния с неиспользуемыми кодами и выполняют переход через эти состояния.
3) Если неиспользуемые коды отсутствуют, вводят дополнительную переменную кодирования для получения дополнительных кодов и выполняют переход через дополнительные состояния.

Для определения возможности возникновения критических состязаний элементов памяти по закодированной минимизированной автоматной таблице строится автоматный граф. У вершин графа отмечаются коды состояний. Просматриваются все переходы и выписываются те из них, на которых меняют своё значения два и более разряда кода, т. е. возникает состязание элементов памяти. Данные состязания устраняются одним из вышеперечисленных способов. После этого строится исправленный автоматный граф, в котором переходы, где возникали состязания элементов памяти, исправляются на новые переходы. По исправленному автоматному графу строится исправленная минимизированная закодированная автоматная таблица.

Продолжим рассматривать пример «Генератор запрета». Строим граф переходов и анализируем возможность возникновения состязания элементов памяти (рисунок 3.13).

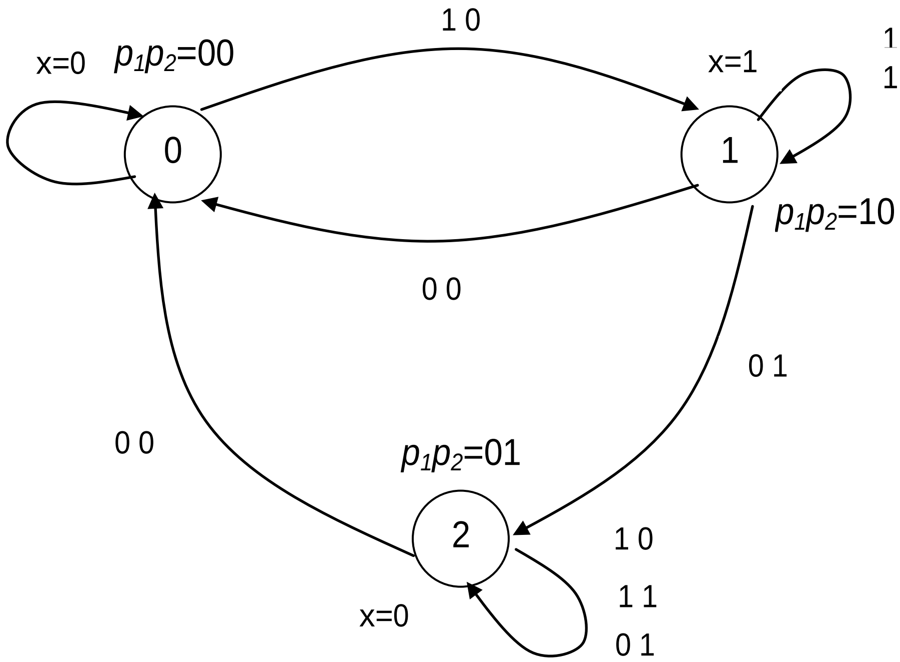


Рисунок 3.13 – Автоматный граф устройства «Генератор запрета»

Из автоматного графа видно, что при переходе из первого состояния во второе меняют свои значения два элемента памяти одновременно: p_1 с 1 на 0 ; p_2 с 0 на 1.

При этом, если быстрее переключается p_1, то код равен 00 и это соответствует нулевому состоянию. Это означает, что при входном наборе a = 0, b = 1 мы оказываемся в нулевом состоянии. В зависимости от того, является ли при данном входе данное состояние устойчивым или неустойчивым автомат может остаться в этом состоянии или перейти в другое состояние, т. е. алгоритм работы управляющего устройства может быть нарушен. Если первым переключается второй элемент памяти, в этом случае попадаем в промежуточное состояние с кодом 11. Данный код в нашем случае не используется. Поэтому можно предусмотреть использование этого дополнительного состояния с этим кодом для того, чтобы обеспечить однозначный переход. Исправленный автоматный граф представлен на рисунке 3.14.

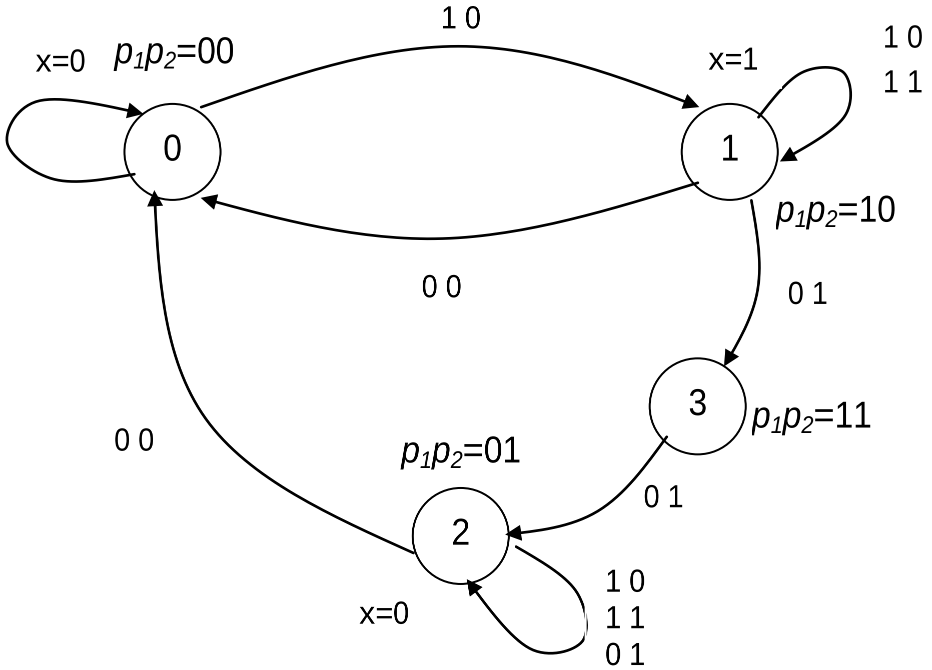


Рисунок 3.14 – Исправленный автоматный граф устройства «Генератор запрета»

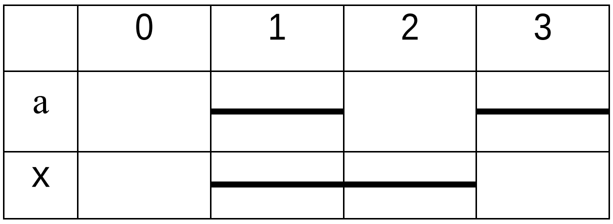
Введение дополнительного состояния не требует дополнительного элемента памяти, так как два элемента памяти позволяют закодировать четыре состояния. Таким образом, мы исключаем состязание вообще.

На основании исправленного автоматного графа строим исправленную минимизированную автоматную таблицу.

	a, b				x	p1	p2
0	1	~	~	0	0	0	0
1	1	1	3	0	1	1	0
2	2	2	2	0	0	0	1
3	~	~	2	~	1	1	1

Пример. Построить минимизированную закодированную автоматную таблицу для устройства Т-триггер.

Линейная диаграмма работы Т-триггера имеет следующий вид.



Составим автоматную таблицу по модели Мура (выходы будем отмечать в отдельном столбце)

	a		x
0	1	0	0
1	1	2	1
2	3	2	1
3	3	0	0

Данная таблица является минимизированной, так как в ней отсутствуют состояния, которые можно совместить. Выполним кодирование состояний.

Внешнее разбиение.

$$\pi_{1x} = \overline{0,\ 3} \qquad \overline{1,\ 2}$$

Варианты второго разбиения

$$\pi_{21} = \overline{0,\ 1} \qquad \overline{2,\ 3}$$

$$\pi_{22} = \overline{0,\ 2} \qquad \overline{1,\ 3}$$

Множества порядка единица:

0 – 0, 3;

1 – 0, 1;

2 – 1, 2;

3 – 2, 3.

Объединённые наборы (максимальное пересечение по одному состоянию):

$\overline{0,\ 1}$ $\overline{1,\ 2}$ $\overline{2,\ 3}$ $\overline{0,\ 3}$.

Наборы $\overline{0,\ 3}$ и $\overline{1,\ 2}$ использовать нельзя, так как они используются во внешнем разбиении. Используем первый и третий наборы. Получим внутреннее разбиение:

$$\pi_2 = \overline{0,\ 1} \qquad \overline{2,\ 3}$$

Закодированная таблица будет иметь вид.

	a		x	p1	p2
0	1	0	0	0	0
1	1	2	1	1	0
2	3	2	1	1	1
3	3	0	0	0	1

Для определения возможности возникновения состязаний элементов памяти построим автоматный граф (рисунок 3.15).

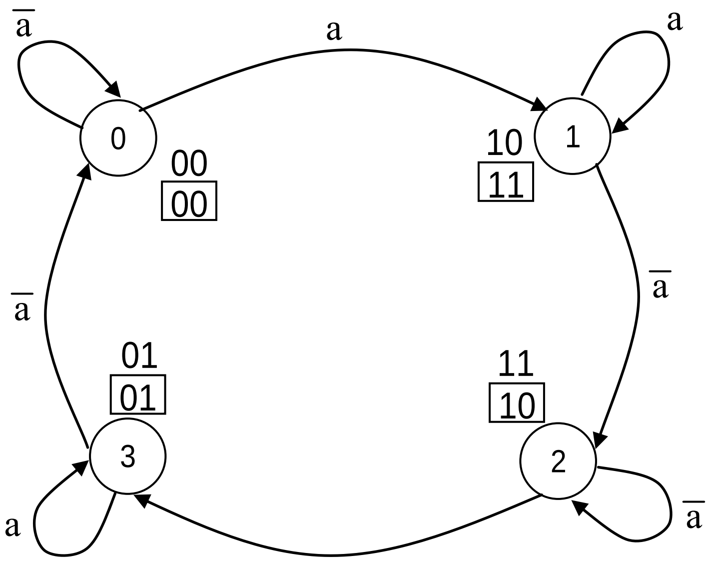


Рисунок 3.15 – Автоматный граф устройства Т-триггер

Из автоматного графа видно, что при распределении второй переменной по первому варианту состязания элементов памяти не возникают. При кодировании по второму варианту (коды в рамочках) возникают два состязания ($0 \to 1$, $2 \to 3$), которые можно устранить только через введение дополнительной переменной кодирования. Поэтому первый вариант предпочтительнее.

Пример. Выполнить кодирование внутренних состояний автомата, заданного минимизированной автоматной таблицей (продолжение примера из минимизации).

	a	b			x
0		1	0	0	0
1	2	1			1
2	2	3	2	2	0
3	3	3		0	1

Внешнее разбиение:

$$\pi_{1x} = \overline{0,\ 2} \qquad \overline{1,\ 3}$$

Множества порядка единица:

0 – 0, 3;

1 – 0, 1;

2 – 1, 2;

3 – 2, 3.

Объединённые наборы (максимальное пересечение по одному состоянию):

$\overline{0,\ 1} \quad \overline{1,\ 2} \quad \overline{2,\ 3} \quad \overline{0,\ 3}$.

Для однозначного кодирования необходимо разнести в разные блоки состояния 0, 2 и 1, 3. Для этого могут использоваться все четыре объединённых набора. Поэтому возможны два варианта распределения значений второй переменной кодирования

$$\begin{aligned} \pi_{21} &= \overline{0,\ 1} \qquad \overline{2,\ 3} \\ \pi_{22} &= \overline{1,\ 2} \qquad \overline{0,\ 3}. \end{aligned}$$

Для начала рассмотрим первый вариант кодирования. Закодированная по первому варианту автоматная таблица будет иметь вид.

	a	b			x p1	p2
0		1	0	0	0	0
1	2	1			1	0
2	2	3	2	2	0	1
3	3	3		0	1	1
	A0	A1	A2	A3		

Построим автоматный граф для определения состязаний элементов памяти (рисунок 3.16).

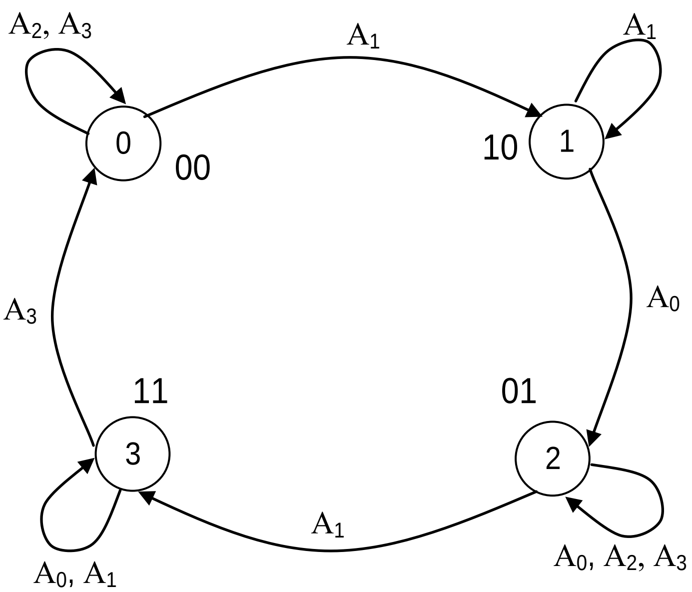


Рисунок 3.16 – Автоматный граф устройства управления

Из автоматного графа видно, что состязания элементов памяти возникают на следующих переходах

$1 \rightarrow 2$ - на наборе A_0,

$3 \rightarrow 0$ – на наборе A_3.

Для устранения состязаний выполним переходы через другие неустойчивые состояния с последовательной сменой кода. Для этого в других состояниях данные наборы должны быть свободны. Переходы выполним следующим образом

$1 \to 0 \to 2,$

$3 \to 1 \to 0.$

Исправленный автоматный граф будет иметь вид, представленный на рисунке 3.17.

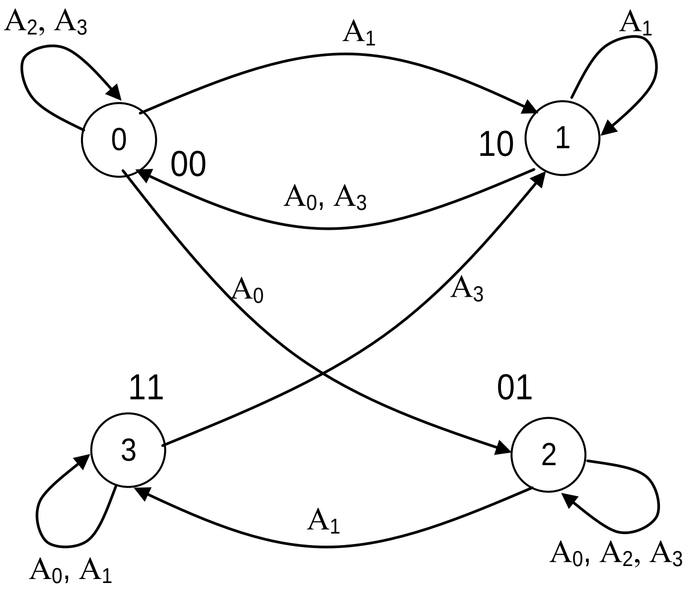


Рисунок 3.17 – Исправленный автоматный граф устройства управления

Из исправленного автоматного графа видно, что все переходы выполняются с последовательной сменой кода, следовательно, критические состязания элементов памяти не возникают. На основании исправленного автоматного графа строим исправленную закодированную автоматную таблицу.

	a, b				x p1	p2
0	2	1	0	0	0	0
1	0	1		0	1	0
2	2	3	2	2	0	1
3	3	3		1	1	1
	A0	A1	A2	A3		

Рассмотрим второй вариант распределения второй переменной кодирования.

$$\pi_{22} = \overline{1,\ 2} \qquad \overline{0,\ 3}$$

Отметим в автоматном графе коды по второму варианту кодирования (рисунок 3.18).

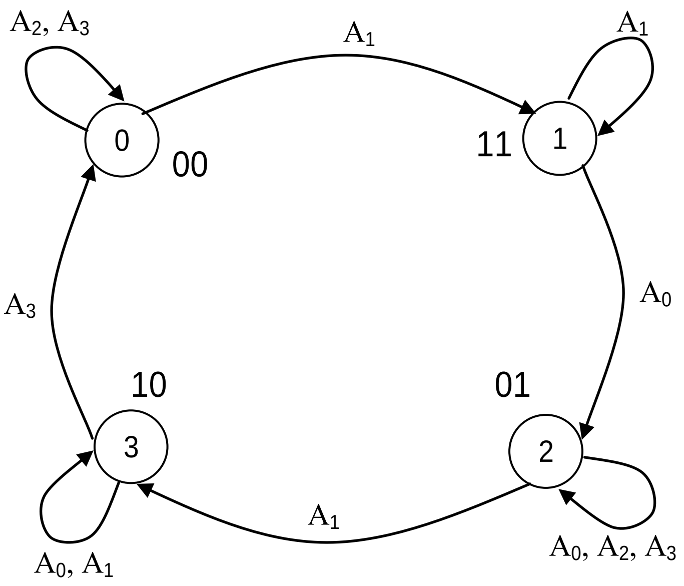


Рисунок 3.18 – Автоматный граф устройства управления для второго варианта кодирования

Из автоматного графа видно, что состязания элементов памяти возникают на следующих переходах

$0 \rightarrow 1$ - на наборе A_1,

$2 \rightarrow 3$ – на наборе A_1.

Так как состязания элементов памяти возникают на одном наборе, а в других состояниях набор A_1 занят (см. минимизированную автоматную таблицу), следовательно исправить данные состязания с помощью переходов через существующие состояния невозможно. Для устранения данных состязаний необходимо вводить дополнительную переменную кодирования, дополнительные состояния и выполнять переходы через них. Это приведёт к существенному усложнению принципиальной схемы. Следовательно, второй вариант кодирования менее предпочтителен по сравнению с первым вариантом распределения кодов.

3.7 Реализация памяти автомата

Основой памяти являются петлеобразные связи. Для того чтобы организовать память, необходимо охватить петлёй обратной связи как минимум один логический элемент. В асинхронных автоматах для реализации памяти используют непосредственно петли обратной связи и асинхронные триггеры, основным из которых является RS. В синхронных автоматах память реализуется только с помощью триггеров.

Для реализации памяти необходимо записать аналитические выражения функций возбуждения памяти (функций переходов). Рассмотрим реализацию памяти с помощью петель обратных связей и триггеров.

IV. <u>Реализация памяти автомата петлями обратных связей.</u>

При реализации памяти петлями обратных связей необходимо выполнить следующие действия.

1) Строится карта Карно для каждой переменной кодирования. При этом, для реализации петель обратных связей, в карту Карно включаются как входные переменные, так и внутренние переменные.
2) По кодам распределяются состояния автоматной таблицы по строкам карты Карно.
3) Заполняются клетки карты Карно. Для этого необходимо номер состояния в соответствующей клетке заменить на значение данной переменной в данном состоянии.
4) В соответствии со стандартными правилами строятся контуры и записываются минимизированные выражения для функций возбуждения памяти.
5) Если необходимо, строятся карты Карно для выходных переменных (если применялось только внутреннее разбиение). При этом, для модели Мура указывается зависимость выходов только от внутренних переменных, для модели Мили – зависимость выходов от вхо-

дов и внутренних переменных. Поданным картам Карно записываются минимизированные выражения для функций выхода.

6) Полученные выражения преобразуются в заданный базис.
7) По полученным выражениям строится комбинационная схема возбуждения памяти (КСВП) и комбинационная схема выходов (КСВ).

Рассмотрим примеры реализации различных устройств.

Пример 1. Построить структурную схему на контактных и бесконтактных элементах устройства «Генератор запрета».

Закодированная минимизированная автоматная таблица данного устройства имеет вид.

	a	a, b	b		x	p1	p2
0	1	~	~	0	0	0	0
1	1	1	3	0	1	1	0
2	2	2	2	0	0	0	1
3	~	~	2	~	1	1	1

По автоматной таблице строим карты Карно для переменных p1 и p2.

a	a, b	b		
1	~	~	0	
0	0	0	0	
~	~	0	~	
1	1	1	0	p1

Записываем минимизированное выражение для переменной p_1.

$$p1 = a\cdot\overline{p2} + b\cdot\overline{p2} = \overline{p2}\cdot(a+b).$$

b

a

0	~	~	0
1	1	1	0
~	~	1	~
0	0	1	0

p1

Записываем минимизированное выражение для переменной $p_{2.}$

$$p2=\bar{a}\cdot b+a\cdot p2.$$

Так как при кодировании использовалось внешнее разбиение, то карту Карно для выходной переменной x строить не надо, т. к. $p1 = x$.

По полученным выражениям строим функциональную схему на контактных элементах.

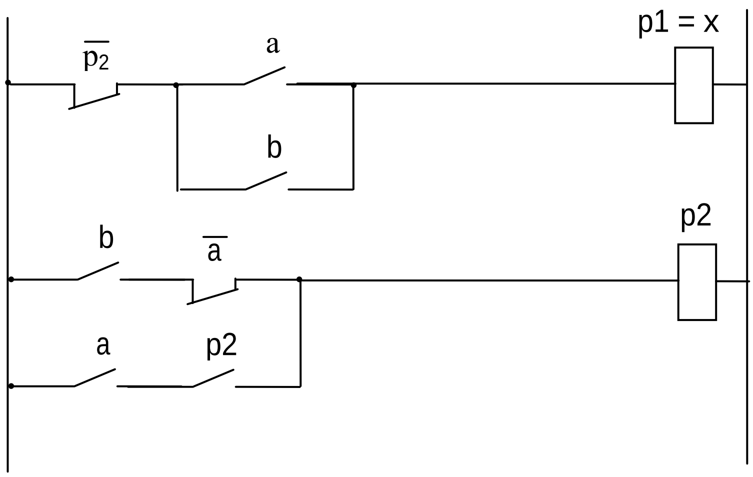


Рисунок 3.19 – Контактная схема устройства «Генератор запрета».

Для построения бесконтактной схемы запишем уравнения в базисе Шеффера.

$$p1=\overline{\overline{a\cdot\overline{p2}+b\cdot\overline{p2}}}=(a\,|\,\overline{p2})\,|\,(b\,|\,\overline{p2})$$

$$p2=\overline{\overline{\bar{a}\cdot b+a\cdot p2}}=(\bar{a}\,|\,b)\,|\,(a\,|\,p2)$$

Бесконтактная схема будет иметь вид.

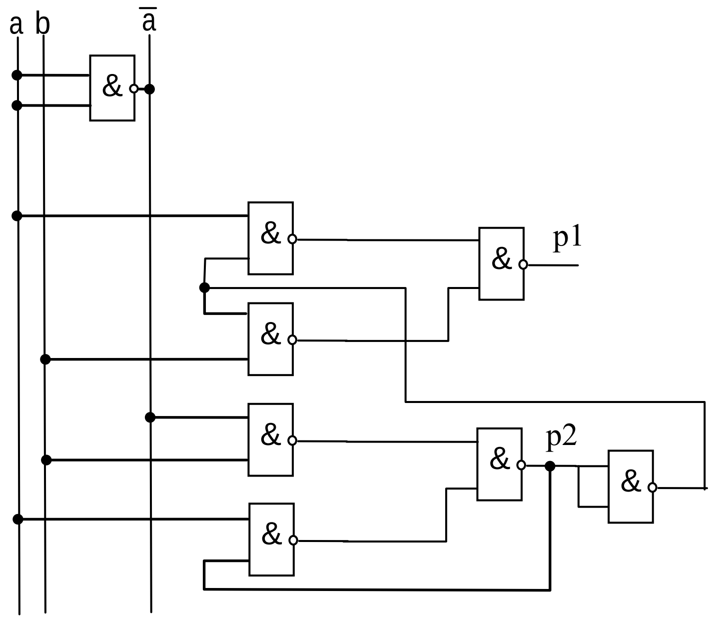


Рисунок 3.20 – Бесконтактная схема устройства «Генератор запрета»

Пример. Построить структурную схему на контактных и бесконтактных элементах устройства, заданного следующей минимизированной закодированной автоматной таблицей (продолжение примера из минимизации и кодирования).

	a, b				x	p
0	2	1	0	0	0	0
1	0	1	~	0	1	0
2	2	3	2	2	0	1
3	3	3	~	1	1	1

Строим карты Карно для переменных x и p.

x (a, b; p, x):

0	1	0	0
0	1	0	0
1	1	~	1
0	1	~	0

p (a, b; p, x):

1	0	0	0
1	1	1	1
1	1	~	0
0	0	~	0

Строим контуры и записываем минимизированные выражения.

$$x = a \cdot b + x \cdot p$$

$$p = a \cdot p + \overline{x} \cdot p + a \cdot \overline{b} \cdot \overline{x} = a \cdot p + \overline{x} \cdot (p + a \cdot \overline{b})$$

По полученным выражениям строим функциональную схему на контактных элементах (рисунок 3.21).

Бесконтактную схему реализуем на элементах И, ИЛИ, НЕ.

Функциональная схема устройства управления на бесконтактных элементах представлена на рисунке 3.22.

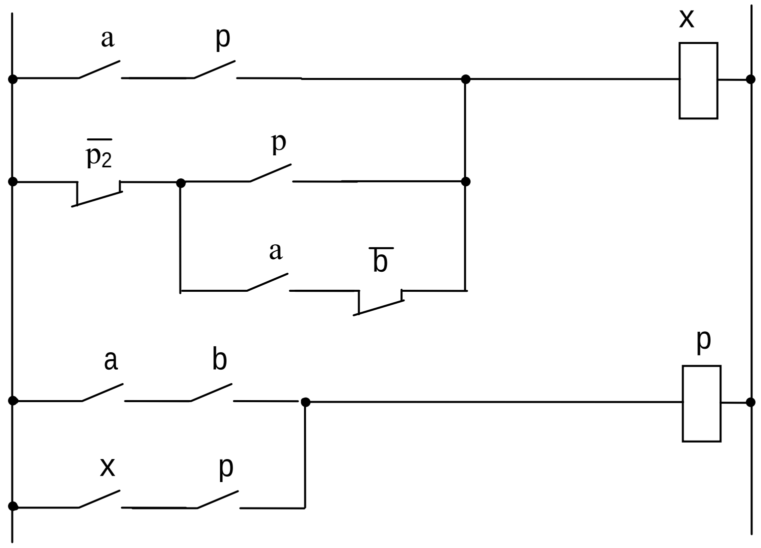


Рисунок 3.21 – Функциональная схема устройства управления на контактных элементах

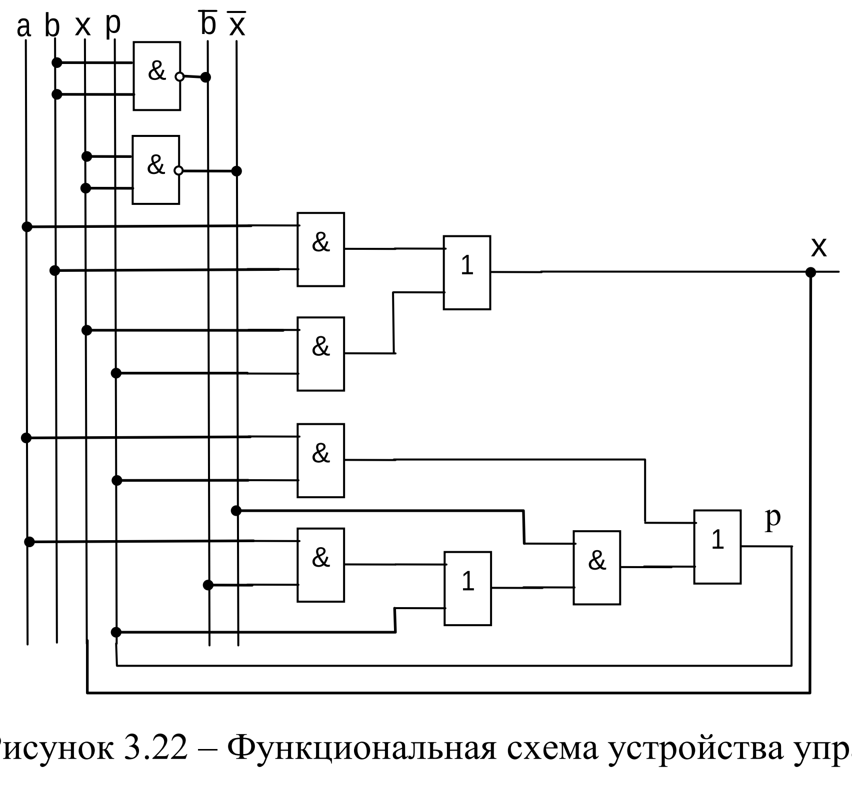


Рисунок 3.22 – Функциональная схема устройства управления

V. Реализация памяти автомата на триггерах.

В бесконтактных схемах в качестве элементов памяти наиболее часто используются триггеры.

Триггеры – элементарные автоматы, содержащие собственно элемент памяти (фиксатор) и схему управления.

Фиксатор строится на двух инверторах, связанных друг с другом "накрест", так что выход одного соединён с входом другого. Такое соединение даёт цепь с двумя устойчивыми состояниями.

Классификация триггеров проводится по признакам логического функционирования и способу записи информации:

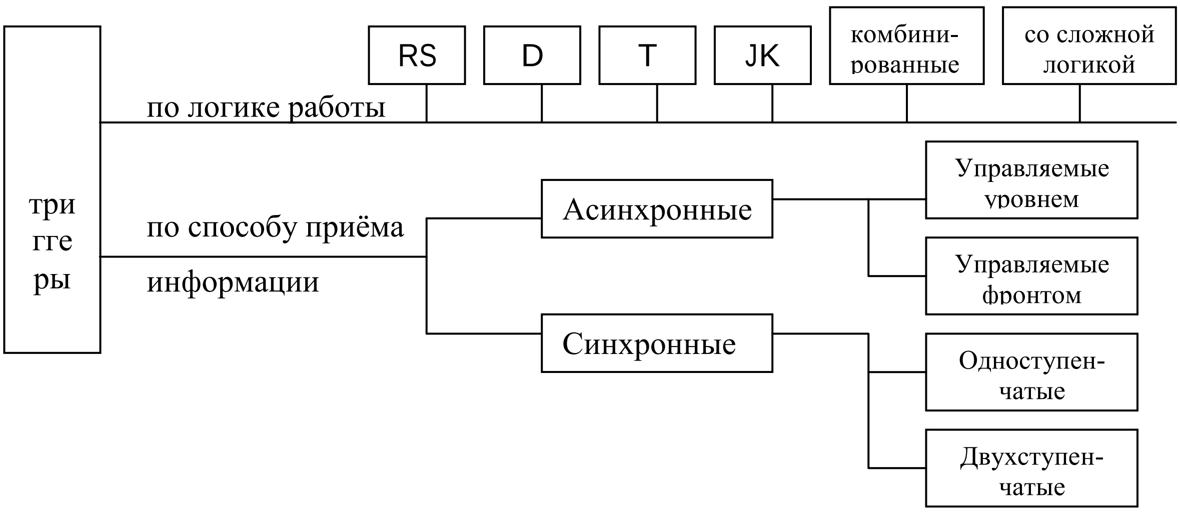


Рисунок 3.23 - Классификация триггеров, используемых в практической схемотехнике

По логическому функционированию различаю триггеры типов RS, D, T, JK и др. Кроме того, используются комбинированные триггеры, в которых совмещаются одновременно несколько типов, и триггеры со сложной входной логикой.

Триггер типа RS имеет два входа – установки в единицу (S от Set) и установки в ноль (R от Reset). Одновременная подача сигналов установки S и сброса R не допускается (запрещенная комбинация сигналов).

Триггер типа D (от слова Delay – задержка) имеет один вход. Его состояние повторяет входной сигнал, но с задержкой, определяемой тактовым сигналом.

Триггер типа T изменяет своё состояние каждый раз при поступлении входного сигнала. Имеет один вход, называется триггером со счётным входом или счётным триггером.

Триггер типа JK универсален, имеет входы установки (J) и сброса (K), подобные входам триггера RS. В отличие от последнего, допускает подачу сигналов на оба эти входа (J = K = 1). В этом режиме работает как счётный триггер относительно третьего (тактового) входа.

По способу записи информации различают асинхронные (нетактируемые) и синхронные (тактируемые) триггеры. В асинхронных переход в новое состояние вызывается непосредственно изменениями входных информационных сигналов. В синхронных, имеющих специальный вход, переход происходит только при подаче на этот вход тактовых сигналов.

По способу восприятия тактовых сигналов триггеры делятся на управляемые уровнем и управляемые фронтом. Триггеры, управляемые фронтом, называют также триггерами с динамическим управлением.

Динамический вход может быть прямым и инверсным. Прямое динамическое управление означает разрешение на переключение при изменении тактового сигнала с нулевого на единичное, инверсное – при изменении тактового сигнала с единичного значения на нулевое.

По характеру процесса переключения триггеры делятся на одноступенчатые и двухступенчатые.

В одноступенчатом триггере переключение в новое состояние происходит сразу, в двухступенчатом – по этапам. Двухступенчатые триггеры состоят из входной и выходной ступеней. Переход в новое состояние происходит в обеих ступенях поочерёдно. Один из уровней тактового сигнала разрешает приём информации во входную ступень при неизменном состоянии выходной ступени. другой уровень тактового сигнала разрешает передачу нового состояния из входной ступени в выходную.

В качестве примера рассмотрим реализацию асинхронного RS – триггера. Таблица переходов и выходов для RS-триггера будет иметь вид:

	R	S			x
0	0	---	1	0	0
1	0	---	1	1	1

Можем записать логическое выражение для выходной переменной X:

$$X = S + \bar{R}\cdot X = \overline{\overline{S + \bar{R}\cdot X}} = \overline{\bar{S}\cdot\overline{\bar{R}\cdot X}} = \bar{S}\,|\,(\bar{R}\,|\,X)$$

Реализуем схему на элементах И-НЕ:

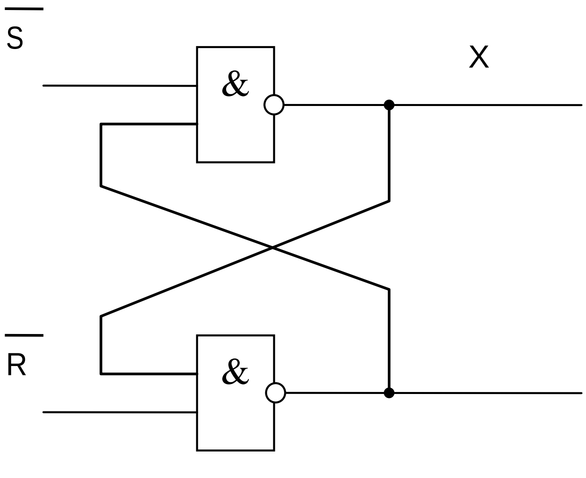


Рисунок 3.24 – Реализация RS-триггера на элементах И-НЕ

Для реализации памяти на триггерах необходимо записать функцию возбуждения памяти для каждого входа триггера. Для этого необходимо выполнить следующие действия.

1) Закодированная автоматная таблица представляется в виде карт Карно. При этом, для каждой переменной кодирования строится две карты Карно, для входа R и входа S триггера.

2) Заполняются карты Карно для каждой переменной кодирования. Вместо номера состояния в клетку вписывается значение входного сигнала триггера, которое обеспечивает его переключение из исходного состояния, определяемого строкой, в последующее, определяемое номером в клетке.
3) По каждой заполненной карте Карно записывается минимизированное логическое выражение функции возбуждения памяти для искомой переменной.
4) Полученные выражения преобразуются в заданный базис.
5) Строится комбинационная схема возбуждения памяти.
6) Строятся карты Карно для выходных переменных, записываются выражения выходных функций в заданном базисе и строится комбинационная схема выходов.

Для заполнения карт Карно можно использовать таблицу переходов RS-триггера.

$X_{исходное}$	RS	$X_{послед}$
0	~ 0	0
0	0 1	1
1	1 0	0
1	0 ~	1

Рассмотрим примеры реализации устройств на RS-триггерах.

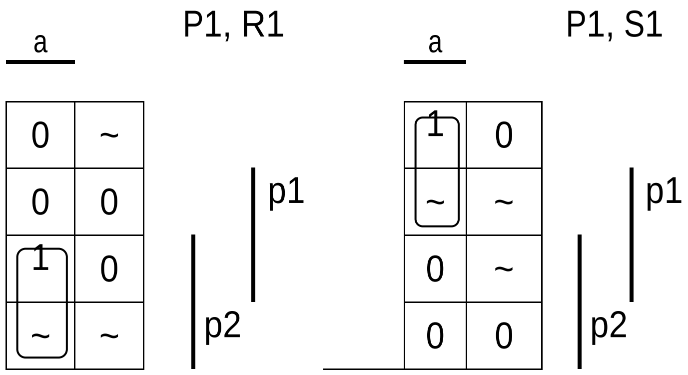
Пример. Реализовать Т-триггер с использованием RS-триггеров.

Закодированная автоматная таблица Т-триггера имеет вид.

	a		x,p1	p2
0	1	0	0	0
1	1	2	1	0
2	3	2	1	1
3	3	0	0	1

Строим по две карты Карно для каждой переменной кодирования.

Для переменной p1.

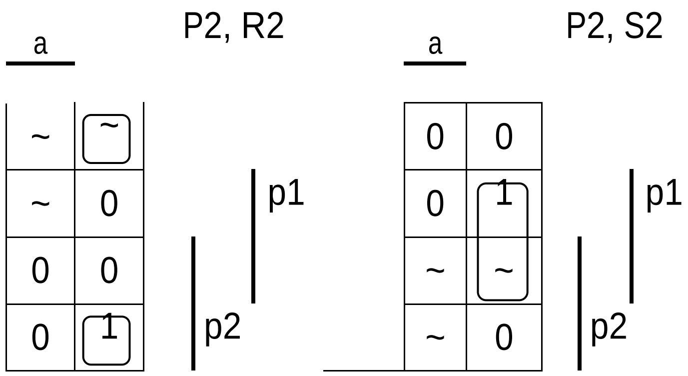


Можем записать логические выражения для переменных R1 и S1 :

$R1 = a \cdot p2$

$S1 = a \cdot \overline{p2}$

Аналогично строим карты Карно для переменной p2.



Выражения для R2 и S2.

$R2 = \overline{a} \cdot \overline{p1}$

$S1 = \overline{a} \cdot p1$

По полученным выражениям строим функциональную схему.

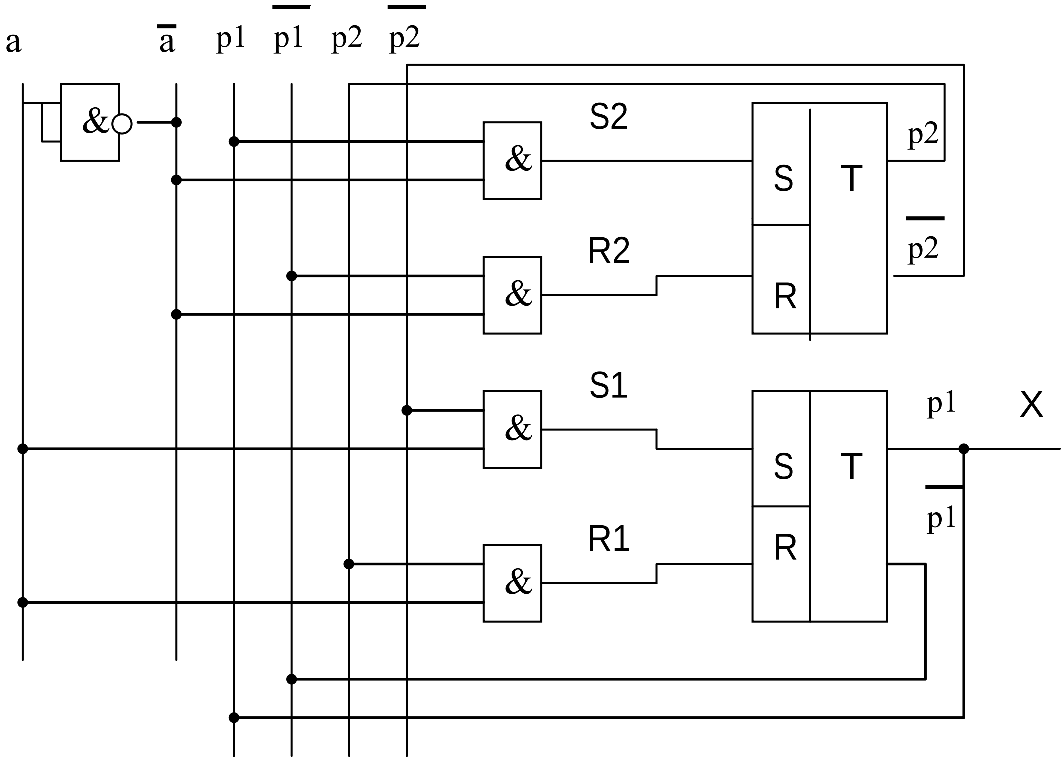


Рисунок 3.25 – Функциональная схема Т-триггера при реализации на RS-триггерах

Пример. Построить структурную схему устройства «Генератор запрета» на бесконтактных элементах с использованием RS-триггеров.

Минимизированная таблица закодированного автомата имеет вид.

	a		b		x	p1	p2
0	1	~	~	0	0	0	0
1	1	1	3	0	1	1	0
2	2	2	2	0	0	0	1
3	~	~	2	~	1	1	1

Для реализации памяти необходимо два элемента памяти. Построим карты Карно для входных переменных каждого триггера.

a		b	R1
0	---	---	---
---	---	---	---
--	--	1	--
0	0	0	1

p1 p2

a		b	S1
1	---	---	0
0	0	0	0
---	---	0	---
---	---	---	0

p1 p2

a		b	R2
---	---	---	---
0	0	0	1
---	---	0	---
---	---	0	---

p1 p2

a		b	S2
0	---	---	0
---	---	---	0
---	---	---	---
0	0	1	0

p1 p2

По картам Карно можем записать логические выражения для искомых переменных.

$$R1 = p2 + \overline{a} \cdot \overline{b}$$
$$S1 = a \cdot \overline{p2}$$
$$R2 = \overline{a} \cdot \overline{b}$$
$$S2 = \overline{a} \cdot b$$

По полученным выражениям строим структурную схему.

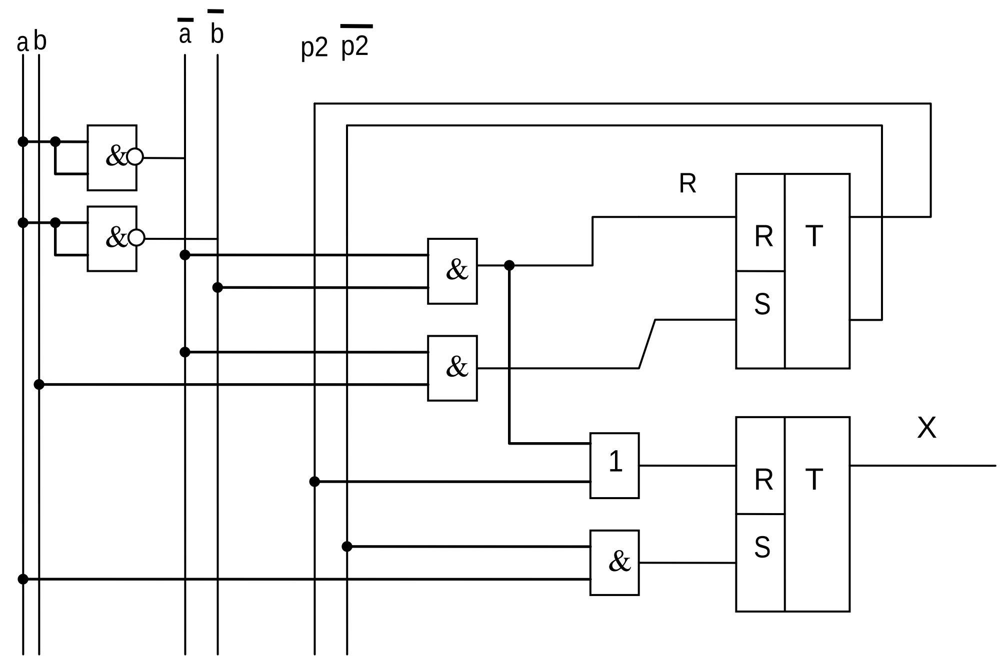


Рисунок 3.26 – Структурная схема «Генератора запрета» на бесконтактных элементах и RS-триггерах

Пример. Построить структурную схему на бесконтактных элементах и триггерах устройства, заданного следующей минимизированной закодированной автоматной таблицей (продолжение примера из минимизации и кодирования).

	a, b				x	p
0	2	1	0	0	0	0
1	0	1	~	0	1	0
2	2	3	2	2	0	1
3	3	3	~	1	1	1

Строим карты Карно для входных переменных триггера.

R1 (a, b; p, x)

0	~	0	~	~
2	~	0	~	~
3	0	0	~	0
1	1	0	~	1

S1 (a, b; p, x)

0	1	0	0
0	1	0	0
~	~	~	~
0	~	~	0

$$R1 = \overline{b} \cdot \overline{p}$$
$$S1 = a \cdot b \,.$$

R2

0	0	~	~	~
2	0	0	0	0
3	0	0	~	1
1	~	~	~	~

S2

1	0	0	0
~	~	~	~
~	~	~	0
0	0	~	0

$$R2 = \overline{a} \cdot x$$
$$S2 = a \cdot \overline{b} \cdot \overline{x} \,.$$

По полученным выражениям строим структурную схему.

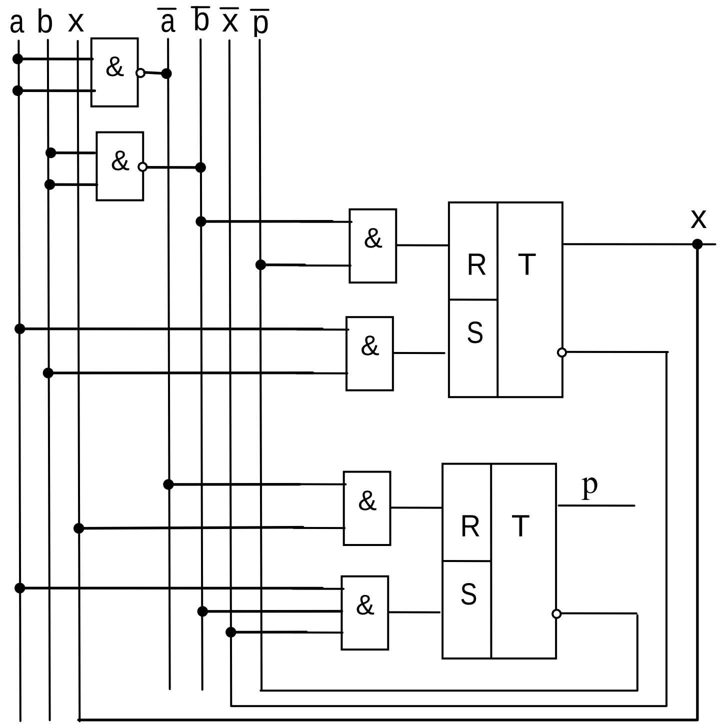
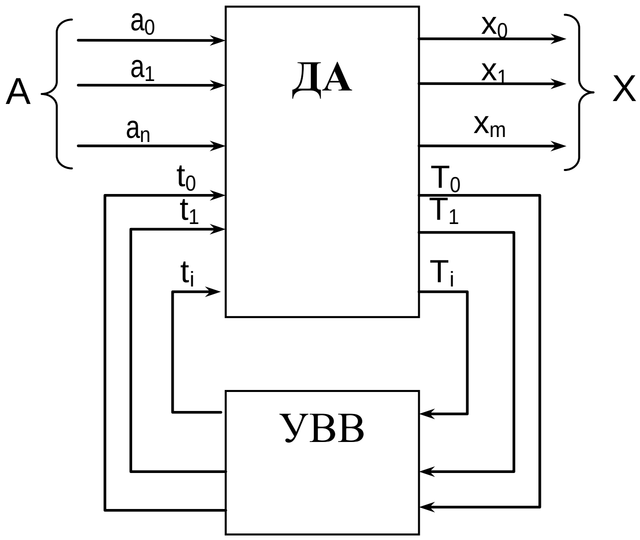


Рисунок 3. 27 – Структурная схема устройства управления

3.8 Синтез управляющих устройств с временными зависимостями

Наличие временных зависимостей в работе дискретного автомата с памятью обычно связано с необходимостью осуществлять в процессе функционирования выдержек времени. При наличии временных зависимостей дискретный автомат представляется в виде следующей функциональной схемы.



где a_0, a_1, …a_n – множество входных сигналов;

x_0, x_1, …x_m – множество выходных сигналов;

T_0, T_1, …T_i – сигналы на включение выдержек времени;

t_0, t_1, …t_i – сигналы, сигнализирующие об окончании выдержек времени;

Рисунок 3.28 – Функциональная схема дискретного автомата при наличии временных зависимостей

При описании таких устройств в автоматную таблицу добавляются дополнительные входные и выходные сигналы, количество которых определяется количеством выдержек времени. Если выдержка времени одна или временные интервалы нескольких выдержек времени одинаковы, то в автоматную таблицу добавляются два сигнала – один входной и один выходной.

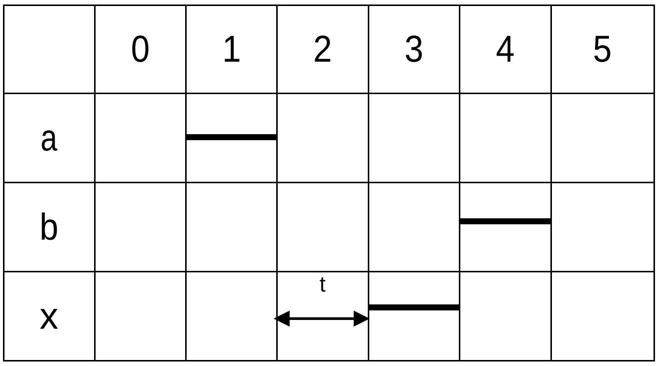
Выходной сигнал обозначается Т и поступает на вход устройства выдержки времени (УВВ). Этот сигнал является сигналом на начало отсчёта выдержки времени.

Входной сигнал обозначается t и поступает из устройства выдержки времени УВВ на вход дискретного автомата. Данный сигнал сигнализирует об окончании выдержки времени.

Таким образом, выдержка времени происходит следующим образом. В определённый момент времени, когда в соответствии с алгоритмом работы устройства необходимо выполнить выдержку времени, на выходе дискретного автомата формируется сигнал Т = 1. По этому сигналу включается устройство выдержки времени и начинается отсчёт выдержки времени. После окончания выдержки времени на выходе УВВ появляется сигнал t = 1 об окончании выдержки времени. Этот сигнал поступает на вход дискретного автомата.

Рассмотрим несколько примеров синтеза дискретных автоматов с временными зависимостями.

Пример. Синтезировать управляющее устройство, заданное следующей линейной диаграммой включения.



В этой таблице сигнал x возникает на выходе через определённый промежуток времени *t*. Поэтому необходим датчик задержки времени (реле времени). Составим автоматную таблицу по модели Мура для данного устройства.

		a	t	b		t			x	T
0	1							0	0	0
1	1							2	0	0
2							3	2	0	1
3							4		1	1
4							4	5	1	0
5					6			5	1	0
6					6			0	0	0

Построим треугольную таблицу совместимости и минимизируем автомат.

1	0,2 X					
2	X	X				
3	X	X	X			
4	X	X	X	X		
5	X	X	X	X	V	
6	V	0,2 X	X	X	X	X
	0	1	2	3	4	5

Запишем группы совместимости:

(4, 5) (0, 6) (1) (2) (3).

Построим минимизированную таблицу.

		a	t	b		t			x	T	p3
0	1				0			0	0	0	0
1	1							2	0	0	1
2							3	2	0	1	1
3							4		1	1	1
4					0		4	4	1	0	0

Выполним кодирование состояний автомата. Первые две переменные кодирования распределим по внешнему разбиению.

$\Pi_1(x) = \overline{0, 1, 2} \quad \overline{3, 4}$

$\Pi_2(T) = \overline{0, 1, 4} \quad \overline{2, 3}$

Пересечение разбиений Π_1 и Π_2 будет равно:

$\Pi_1 * \Pi_2 = \overline{0, 1} \quad \overline{4} \; \overline{2} \; \overline{3}$

Неразличимыми остались состояния 0 и 1. Запишем множества порядка единица.

0 – 0, 4

1 – 0, 1

2 – 1, 2

3 - 2

4 – 3, 4

Выписываем объединённые наборы:

(0, 1) (1, 2) (2, 3) (0, 4).

Примем следующее разбиение третьей переменной кодирования:

$\Pi_3 = \overline{0, 4} \quad \overline{1, 2, 3}.$

Проверяем, равно ли пересечение полученных разбиений нулевому разбиению.

$\Pi_1 * \Pi_2 * \Pi_3 = \overline{0} \quad \overline{1} \; \overline{2} \; \overline{3} \; \overline{4}$

Проанализируем полученные коды внутренних состояний на возникновение состязаний с помощью автоматного графа.

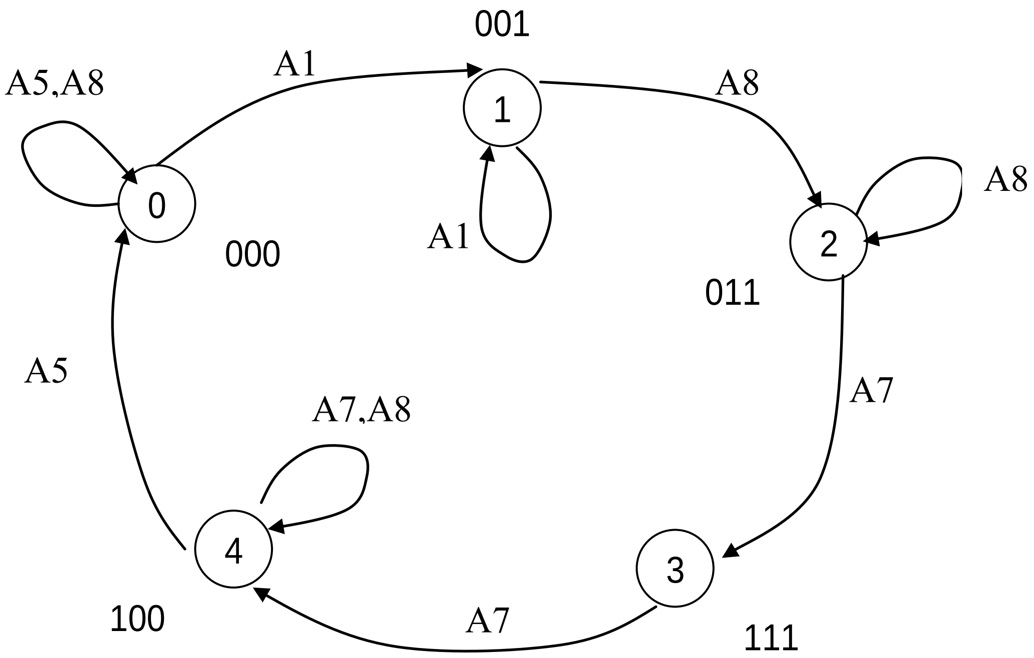


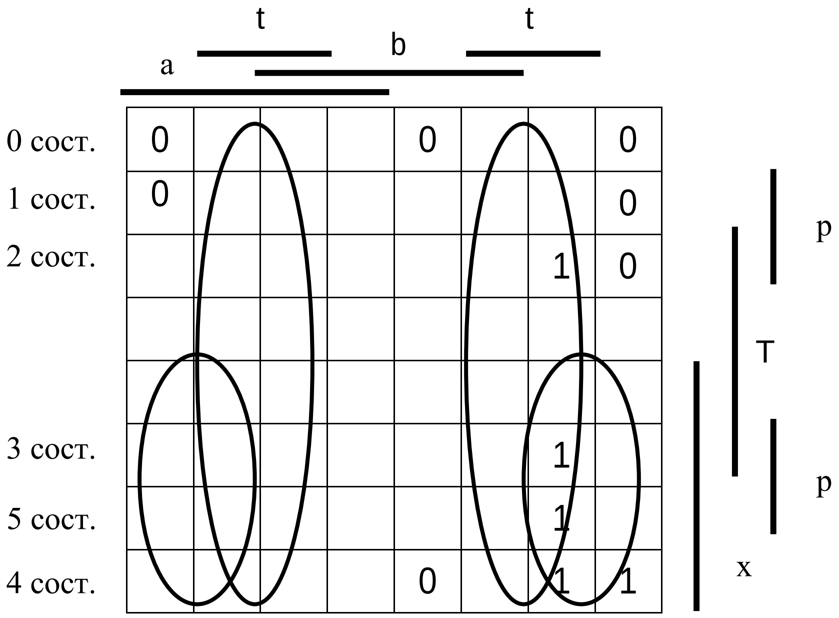
Рисунок 3.29 – Автоматный граф

При переходе из третьего в четвёртое состояние возникает состязание элементов памяти. Для устранения состязания введём дополнительное 5 состояние с кодом 101 и выполним переход через это состояние. Исправленная минимизированная автоматная таблица будет иметь вид.

		a	t	b		t			x	T	p3
0	1				0			0	0	0	0
1	1							2	0	0	1
2							3	2	0	1	1
3							5		1	1	1
4					0		4	4	1	0	0
5							4		1	0	1

Составим карты Карно для переменных X, T, P.

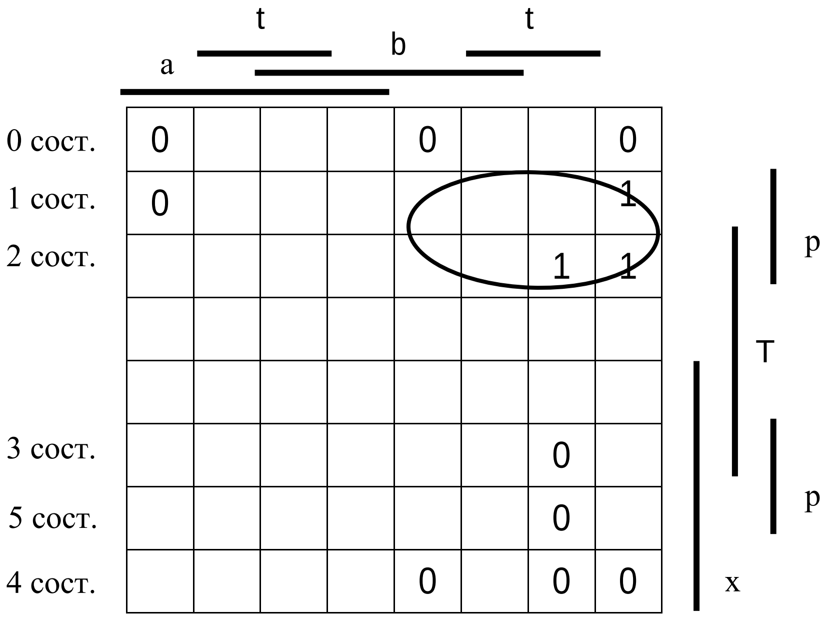
Карта Карно для переменной X.



Запишем логическое выражение для X:

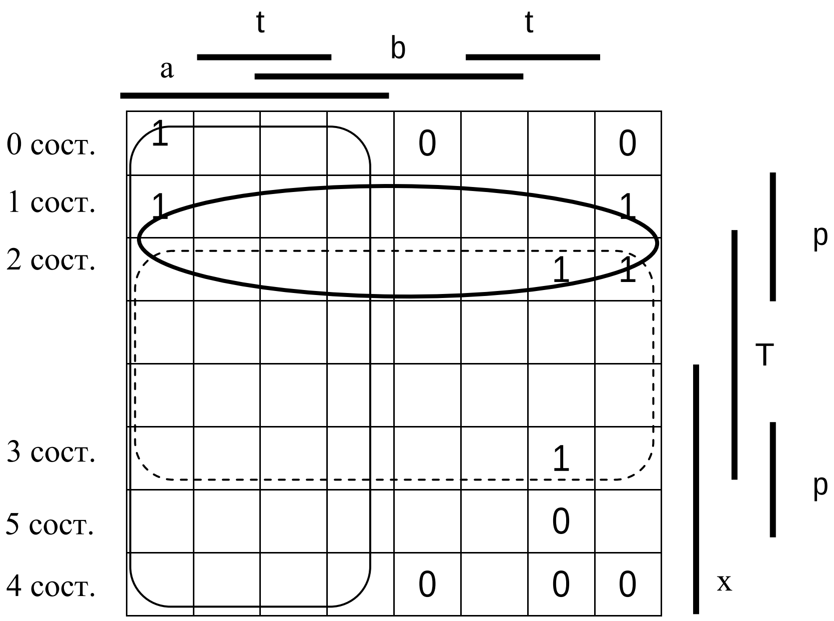
$$x = \bar{b} \cdot x + t$$

Карта Карно для переменной T.



$$T = \bar{a} \cdot \bar{x} \cdot p$$

Карта Карно для переменной p.



$$p = T + a + p \cdot \overline{x}$$

По полученным выражениям можем реализовать структурную схему на контактных элементах.

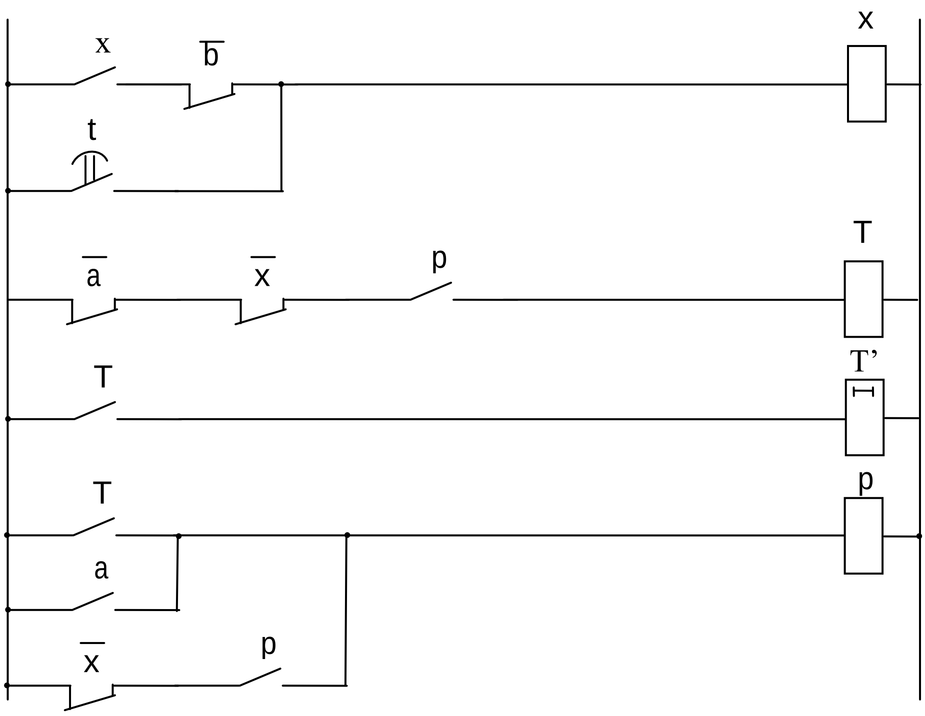


Рисунок 3.30 – Структурная схема устройства управления на контактных элементах

Рассмотрим синтез устройства при описании выдержки времени по модели Мили.

Автоматная таблица:

		a	t	b		t			x
0	1,0							0,0	0
1	1,0							2,~	0
2							3,1	2,1	0
3						4,~	3,1		1
4					5,0	4,0			0
5					5,0			0,0	0

Составим треугольную таблицу совместимости и минимизируем автомат.

1	0,2 X				
2	X	V			
3	X	X	X		
4	V	V	V	X	
5	V	2,0 X	X	X	V
	0	1	2	3	4

Выписываем группы совместимости и обозначаем новые состояния минимизированного автомата.

1, 2, 4 – 1, 2 – 1

0, 4, 5 – 0, 4, 5 – 0

3 – 2

Строим минимизированную автоматную таблицу.

		a	t	b		t			x	p
0	1,0				0,0	0,0		0,0	0	0
1	1,0						2,1	1,1	0	1
2						0,~	2,1		1	1

Распределение переменной кодирования по внешнему разбиению:

$\Pi_1(x) = \overline{0, 1} \quad \overline{2}$.

Определим внутреннее разбиение. Запишем множества порядка единица.

0 – 0, 2

1 – 0, 1

2 – 1,2

Выписываем объединённые наборы:

(0, 1) (0, 2) (1, 2).

Примем внутреннее разбиение следующего вида:

$\Pi_2(p) = \overline{0} \quad \overline{1, 2}$.

Проанализируем полученный эквивалентный автомат с помощью автоматного графа.

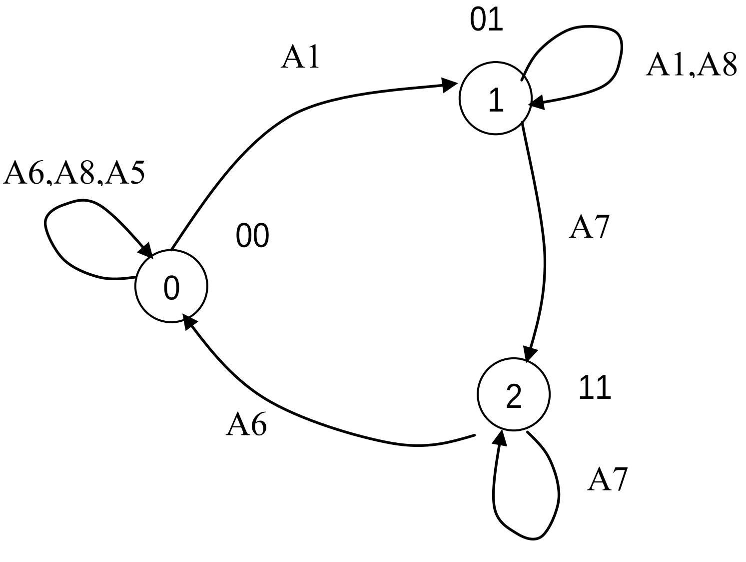


Рисунок 3.31 – Автоматный граф

Как видно из автоматного графа в схеме возникают состязания элементов памяти. При переходе из второго в нулевое состояние переключаются оба элемента памяти. Если первым переключается первый элемент, то мы переходим вместо нулевого состояния в первое состояние, что недопустимо. Для исключения критического состязания осуществим переход из второго состояния в нулевое через первое неустойчивое состояние.

Исправленная минимизированная автоматная таблица будет иметь вид.

		a	t	b		t			x	p
0	1,0				0,0	0,0		0,0	0	0
1	1,0					0,~	2,1	1,1	0	1
2						1,~	2,1		1	1

Построим карты Карно для искомых переменных и определим логические выражения.

Карта Карно для переменной x.

	a	t		b		t		
0 сост.	0				0	0		0
1 сост.	0					0	1	0
2 сост.						0	1	

p

X

$$X = \bar{b} \cdot t$$

Карта Карно для переменной p.

0 сост.	1				0	0		0
1 сост.	1					0	1	1
2 сост.						1	1	

$$p = a + x + \bar{b} \cdot p$$

Карта Карно для переменной T.

0 сост.	0				0	0		0
1 сост.	0					~	1	1
2 сост.						~	1	

$$T = \bar{a} \cdot p$$

На основании полученных выражений строим новый вариант схемы:

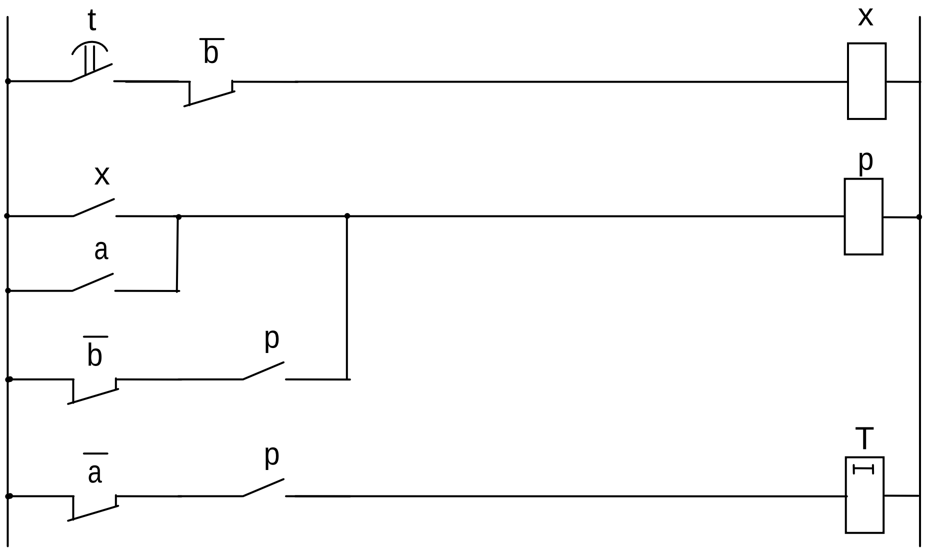


Рисунок 3.32 – Структурная схема устройства управления на контактных элементах при описании по модели Мили

Построим структурную схему на бесконтактных элементах и RS-триггерах.

Строим две карты Карно для каждой переменной кодирования.

		t	t			t	t	R1
			b	b	b	b		
	a	a	a	a				
0	~				~	~		~	
1	~					~	0	~	p
2						1	0		x p
									x

$R1 = b$

		t	t			t	t	S1
			b	b	b	b		
	a	a	a	a				
0	0				0	0		0	
1	0					0	1	0	p
2						0	~		x p
									x

$S1 = t \cdot \overline{b}$

		t	t			t	t	R2
			b	b	b	b		
	a	a	a	a				
0	0				~	~		~	
1	0					1	0	0	p
2						0	0		x p
									x

$R2 = b \cdot \overline{x}$

		t	t			t	t	S2
			b	b	b	b		
	a	a	a	a				
0	1				0	0		0	
1	~					0	~	~	p
2						~	~		x p
									x

$S2 = a$

По полученным выражениям строим схему на бесконтактных элементах.

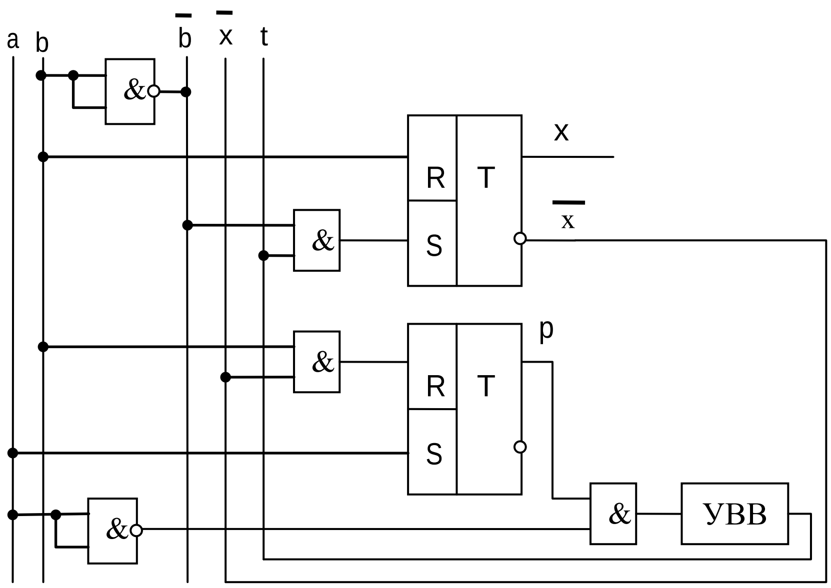
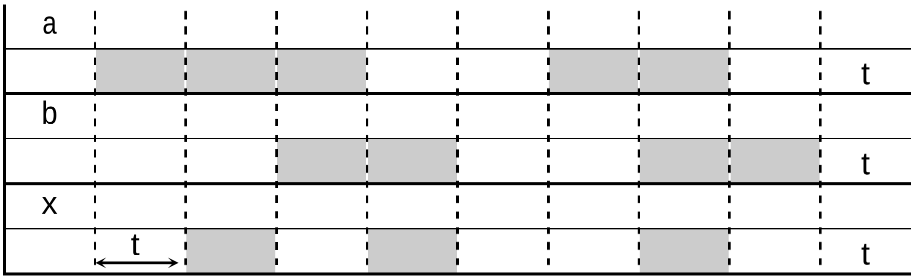
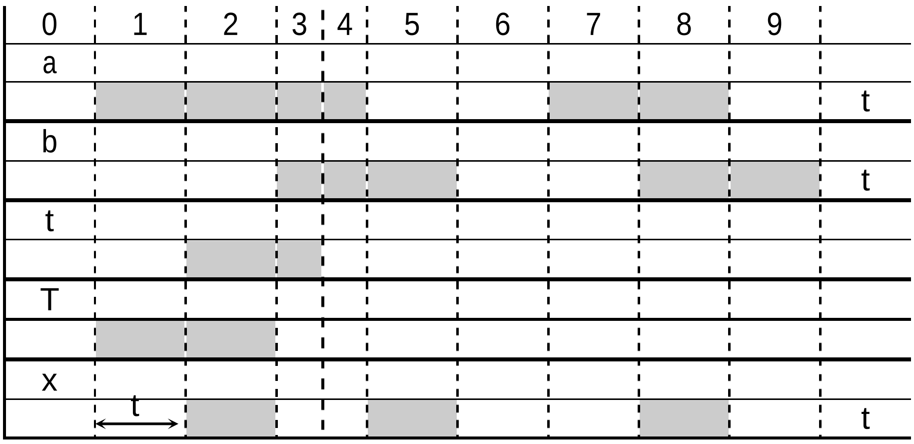


Рисунок 3.33 – Структурная схема устройства управления на бесконтактных элементах при описании по модели Мили

Пример. Синтезировать управляющее устройство, заданное следующей линейной диаграммой включения. При синтезе учесть возможность изменения момента времени отключения реле времени.



Составим автоматную таблицу. Реле времени отключим сразу после отсчёта выдержки времени. Для полного описания выдержки времени и определения количества внутренних состояний автомата линейную диаграмму можно переделать, добавив два сигнала T и t.



	a		t	b		t			x
0	1,~								0
1	1,1	2,1							0
2		2,1	3,~						1
3			3,0	4,0					0
4				4,0	5,0				0
5					5,0			6,0	1
6	7,0							6,0	0
7	7,0			8,0					0
8				8,0	9,0				1
9					9,0			0,0	0

Выполним минимизацию автоматной таблицы.

1	V								
2	X	X							
3	V	V	X						
4	V	V	X	V					
5	X	X	V	X	X				
6	X	X	X	V	V	X			
7	X	X	X	4,8 X	4,8 X	X	V		
8	X	X	V	X	X	5,9 X	X	X	
9	V	V	X	V	5,9 X	X	6,0 V	V	X
	0	1	2	3	4	5	6	7	8

Выписываем группы совместимости.

7, 9 - A

6, 7 - B

3, 4, 6 - C

0, 1, 3, 9 - D

2, 5 - E

2, 8 - F

0, 1, 4 - G

Для определения минимального класса совместимости строим таблицу покрытия.

	0	1	2	3	4	5	6	7	8	9
A								X		X
B							X	X		
C				X	X		X			
D	X	X		X						X
E			X			X				
F			X						X	
G	X	X			X					

Записываем формулу покрытия.

Ф = (A+D)*F*(A+B)*(B+C)*E*(C+G)*(C+D)*(E+F)*(D+G)*(D+G) = E*F*(A+D)*(A+B)*(B+C)*(C+G)*(C+D)*(D+G)

Раскрыв все скобки, выбираем минимальный класс совместимости A*C*D*E*F

Получаем следующие новые состояния.

A -	7, 9	- 7, 9	- 3
C -	3, 4, 6	- 4, 6	- 2
D -	0, 1, 3, 9	- 0, 1, 3	- 0
E -	2, 5	- 2, 5	- 1
F -	2, 8	- 8	- 4

Строим минимизированную автоматную таблицу.

	a		t	b		t			x
0	0,1	1,1	0,0	2,0				0,0	0
1		1,1	0,~		1,0			2,0	1
2	3,0			2,0	1,0			2,0	0
3	3,0			4,0	3,0			0,0	0
4				4,0	3,0				1

Выполним кодирование внутренних состояний автомата.

Распределение переменной кодирования по внешнему разбиению:

$П_1$(x) = $\overline{0, 2, 3}$ $\overline{1, 4}$.

Для однозначного кодирования необходимо ввести ещё две переменных кодирования. Распределим значения данных переменных по внутреннему разбиению. Запишем множества порядка единица.

0 – 0, 1, 3

1 – 0, 1, 2

2 – 0, 1, 2

3 – 2, 3, 4

4 – 3, 4

Выписываем объединённые наборы:

(1, 2) – по трём состояниям;

(0, 1) (0, 2) (3, 4).

Примем внутреннее разбиение следующего вида:

$\Pi_2(\mathrm{p}1) = \overline{0, 1, 2}\quad \overline{3, 4}$

$\Pi_3(\mathrm{p}2) = \overline{1, 2}\quad \overline{0, 3, 4}$

Получаем закодированную минимизированную автоматную таблицу следующего вида.

	a		t	b		t			x	p1	p2
0	0,1	1,1	0,0	2,0				0,0	0	0	1
1		1,1	0,~		1,0			2,0	1	0	0
2	3,0			2,0	1,0			2,0	0	0	0
3	3,0			4,0	3,0			0,0	0	1	1
4				4,0	3,0				1	1	1
	A1	A2	A3	A4	A5			A6			

Проанализируем полученный эквивалентный автомат с помощью автоматного графа.

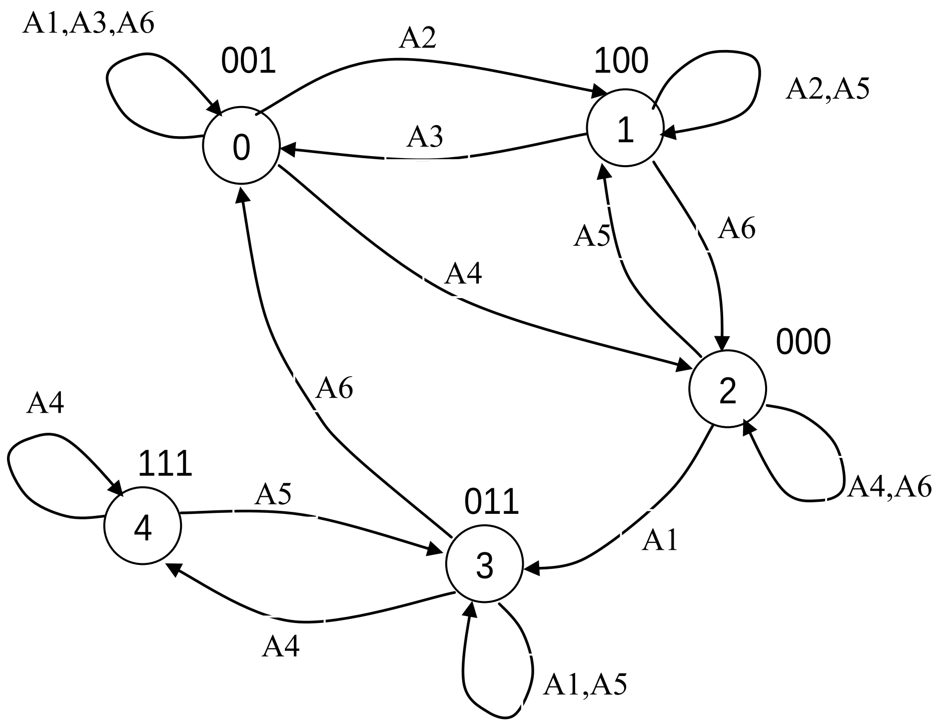


Рисунок 3.34 – Автоматный граф устройства управления

Из автоматного графа видно, что состязания элементов памяти возникают на следующих переходах:

$0 \to 1$ исправим $0 \to 2 \to 1$;

$1 \to 0$ исправим $1 \to 2 \to 0$;

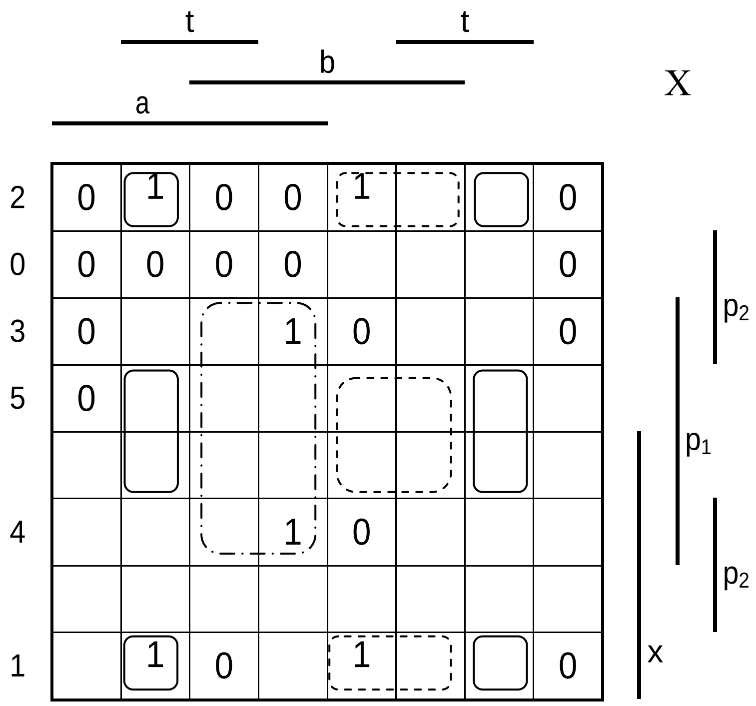
$2 \to 3$ исправим, добавив состояние 5 с кодом 010 $2 \to 5 \to 3$.

Исправленная минимизированная автоматная таблица будет иметь вид.

	a		t	b		t			x	p1	p2
0	0,1	2,1	0,0	2,0				0,0	0	0	1
1		1,1	2,~		1,0			2,0	1	0	0
2	5,0	1,1	0,~	2,0	1,0			2,0	0	0	0
3	3,0			4,0	3,0			0,0	0	1	1
4				4,0	3,0				1	1	1
5	3,0								0	1	0

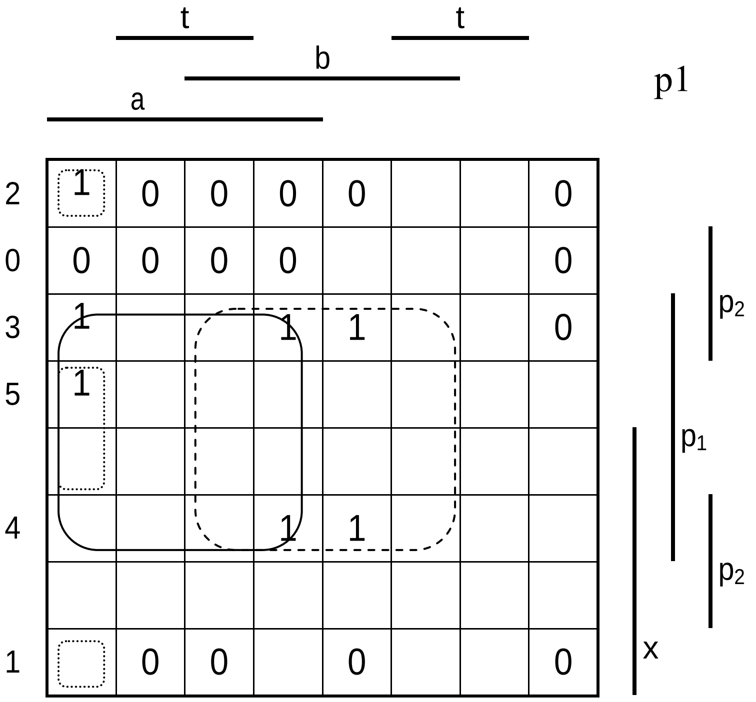
Строим карты Карно и записываем аналитические выражения для функций возбуждения памяти и выходных функций.

Карта Карно для переменной х.



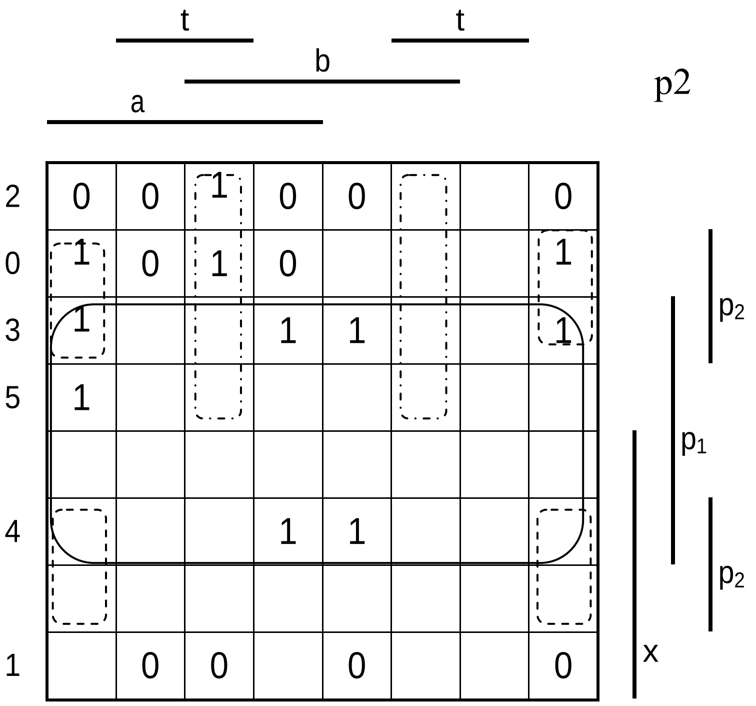
$$x = a \cdot b \cdot p1 + \bar{b} \cdot t \cdot \overline{p2} + \bar{a} \cdot b \cdot \overline{p2} = a \cdot b \cdot p1 + \overline{p2} \cdot (\bar{b} \cdot t + \bar{a} \cdot b)$$

Карта Карно для переменной p1.



$$p1 = a \cdot p1 + b \cdot p1 + a \cdot \bar{b} \cdot \bar{t} \cdot \overline{p2} = p1 \cdot (a + b) + a \cdot \bar{b} \cdot \bar{t} \cdot \overline{p2}$$

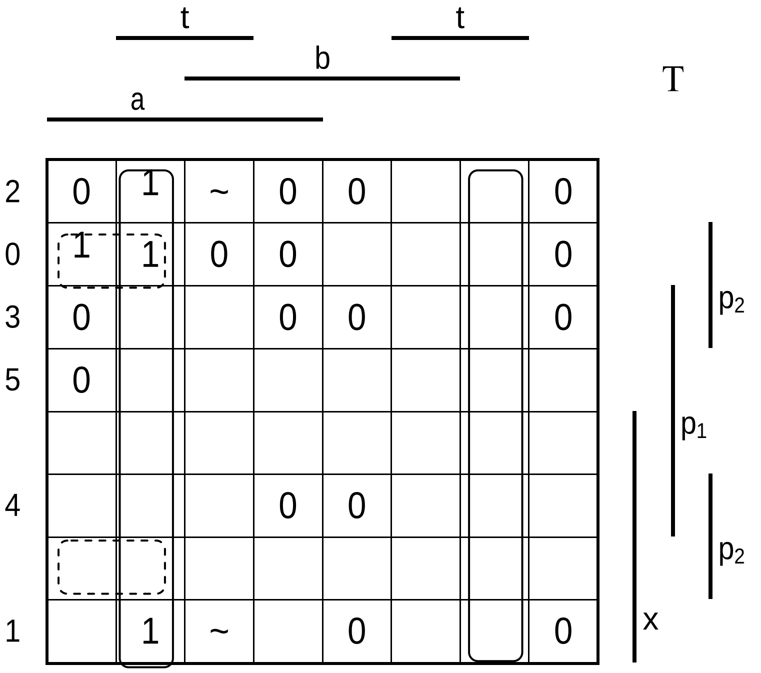
Карта Карно для переменной p2.



$$p2 = p1 + \bar{b} \cdot \bar{t} \cdot p2 + b \cdot t \cdot \bar{x}$$

На основании полученных выражений строим структурную схему на контактных элементах.

Карта Карно для переменной Т.



$$T = \bar{b} \cdot t + a \cdot \bar{b} \cdot \overline{p1} \cdot p2$$

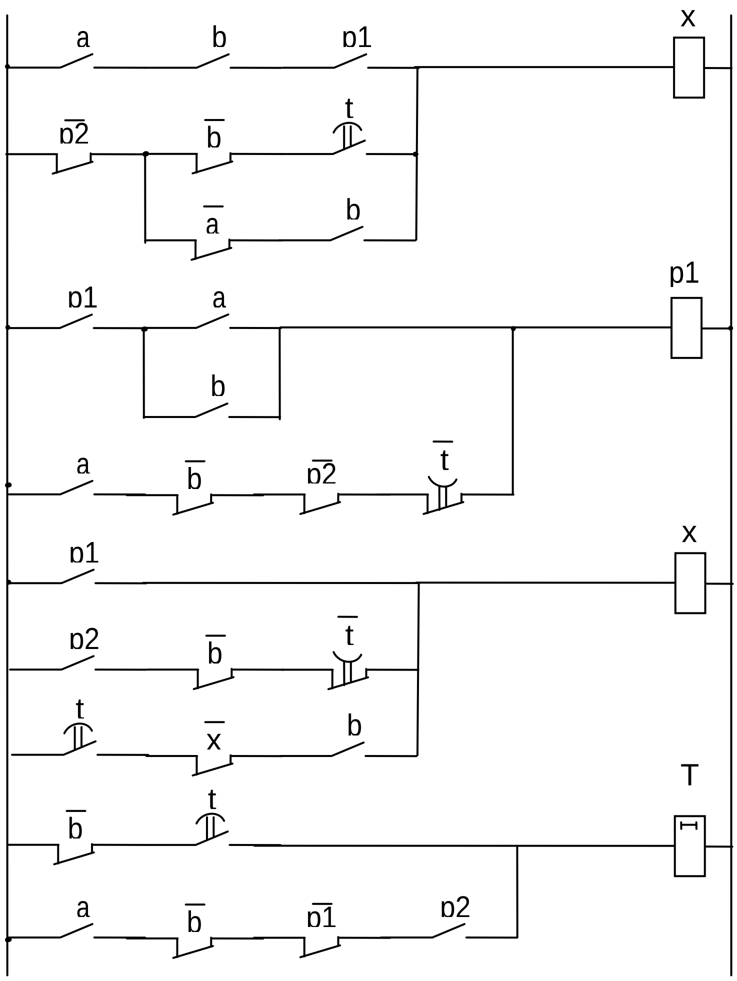
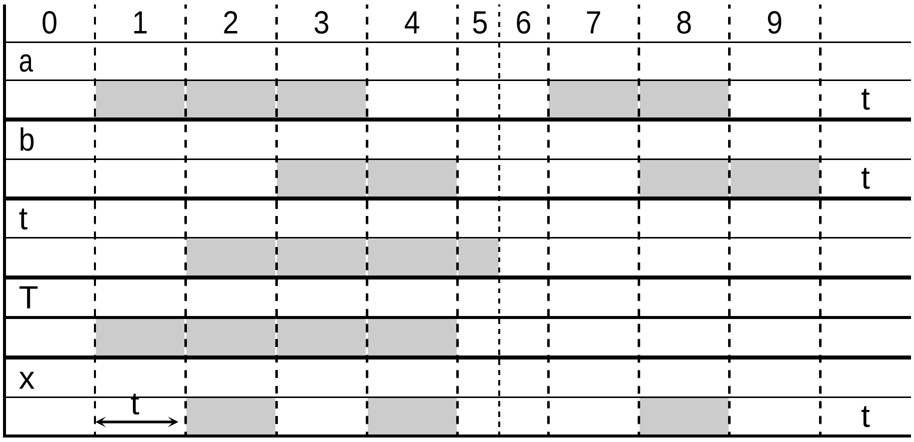


Рисунок 3.35 – Структурная схема устройства управления

Выполним описание данного устройства, изменив момент отключения реле времени.



	a		t	b		t			x
0	1,~								0
1	1,1	2,1							0
2		2,1	3,1						1
3			3,1			4,1			0
4						4,1	5,~		1
5							5,0	6,0	0
6	7,0							6,0	0
7	7,0			8,0					0
8				8,0	9,0				1
9					9,0			0,0	0

Выполним минимизацию автоматной таблицы.

1	V								
2	X	X							
3	V	V	X						
4	X	X	V	X					
5	0,6 X	V	X	V	X				
6	X	X	X	V	X	V			
7	X	X	X	V	X	V	V		
8	X	X	V	X	V	X	X	X	
9	V	V	X	V	X	X	X	V	X
	0	1	2	3	4	5	6	7	8

Выписываем группы совместимости.

3, 7, 9 - A

3, 5, 6, 7 - B

2, 4, 8 - C

1, 3, 5 - D

0, 1, 3, 9 - E

Для определения минимального класса совместимости строим таблицу покрытия.

	0	1	2	3	4	5	6	7	8	9
A				X				X		X
B				X		X	X	X		
C			X		X				X	
D		X		X		X				
E	X	X		X						X

Записываем формулу покрытия.

Ф = (A+E)*C*(A+B)*B*(B+D)*C*(A+B+D+E)*C*(D+E)*E = B*C*E

Получаем следующие новые состояния.

B - 3, 5, 6, 7 - 5, 6, 7 - 2

C - 2, 4, 8 - 2, 4, 8 - 1

E - 0, 1, 3, 9 - 0, 1, 3, 9 - 0

Строим минимизированную автоматную таблицу.

	a		t	b		t			x
0	0,1	1,1	0,1		0,0	1,1		0,0	0
1		1,1	0,1	1,0	0,0	1,1	2,~		1
2	2,0			1,0			2,0	2,0	0

Полученная автоматная таблица имеет всего три состояния. Для кодирования этих состояний достаточно двух переменных кодирования. Следовательно, схема устройства будет значительно проще, чем схема, полученная при первом способе описания. Таким образом, изменяя момент отключения реле времени, можно существенно упростить схему.

2. ПРАКТИЧЕСКИЙ РАЗДЕЛ

Практический раздел УМК представлен планом практических занятий, а также методическим пособием по выполнению курсовой работы.

План практических занятий

1. Задание функций алгебры логики.
2. Равносильные преобразования функций алгебры логики.
3. Запись логических выражений в виде нормальных и совершенных нормальных форм.
4. Преобразование логических выражений в заданный базис.
5. Минимизация функций алгебры логики.
6. Синтез комбинационных автоматов на контактных и бесконтактных логических элементах.
7. Задание дискретных автоматов с памятью.
8. Минимизация памяти автоматов.
9. Кодирование состояний автоматов с памятью.
10. Устранение критических состязаний элементов памяти.
11. Реализация памяти петлями обратных связей.
12. Реализация памяти автомата на триггерах.
13. Синтез комбинационной схемы выходов автомата.
14. Синтез дискретных автоматов с временными зависимостями.

Министерство образования Республики Беларусь
БЕЛОРУСССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра «Электропривод и автоматизация промышленных установок и технологических комплексов»

А. А. Мигдалёнок

МЕТОДИЧЕСКОЕ ПОСОБИЕ

к курсовой работе по дисциплине «Проектирование дискретных систем управления» для студентов специальности
1 - 53 01 05 – «Автоматизированные электроприводы»

СОДЕРЖАНИЕ

1 ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ. ПОРЯДОК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

При выполнении курсовой работы необходимо синтезировать схему управления перемещением механизма каретки между тремя положениями по заданному циклу.

Функциональная схема установки приведена на рисунке 1.

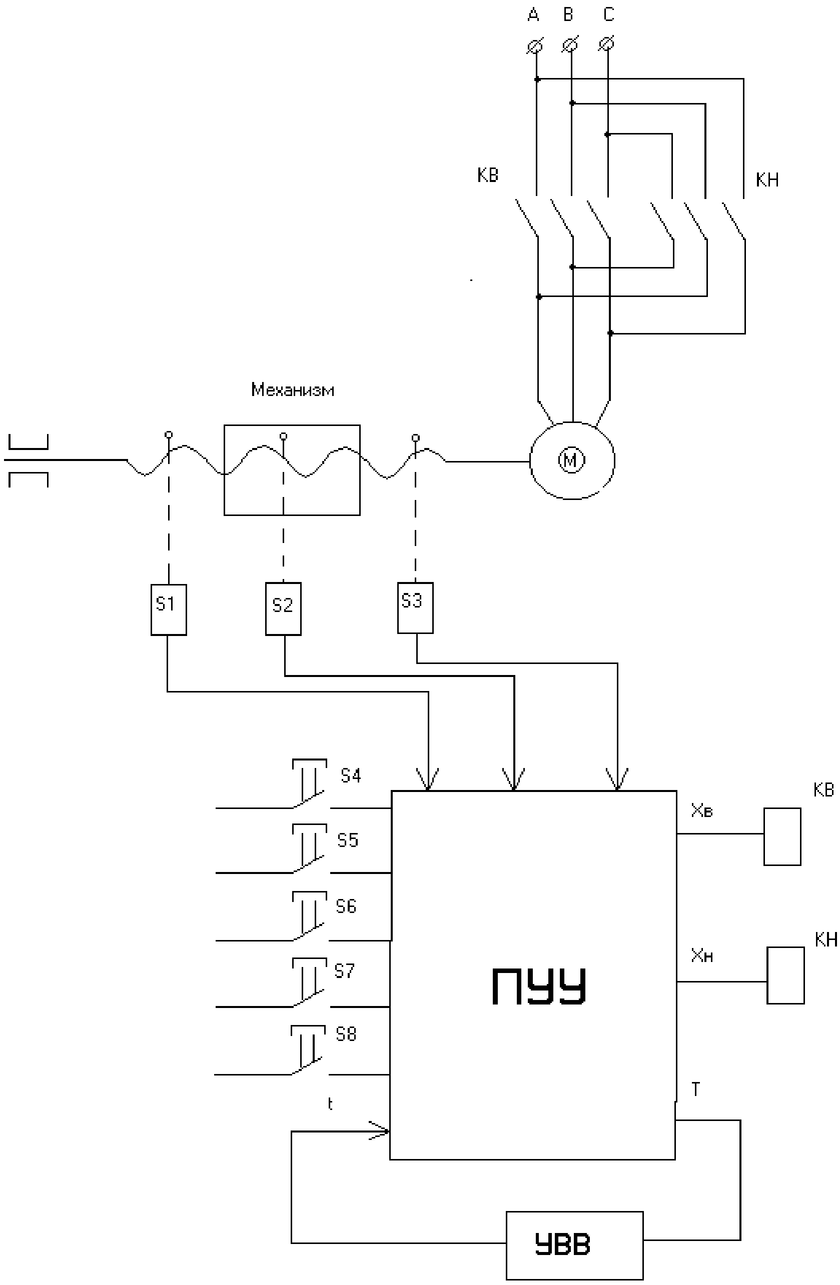


Рисунок 1 – Функциональная схема установки

На функциональной схеме введены следующие обозначения:

КВ, КН – контакторы движения "вперёд", "назад";

Хв, Хн – сигналы включения контакторов "вперёд", "назад";

S1, S2, S3 – конечные выключатели в положениях 1, 2, 3;

S4 – кнопка "Пуск" в режиме "Автомат" (S4=1 соответствует включённому состоянию кнопки "Пуск");

S5 – кнопка "Стоп" в режиме "Автомат";

S6 – кнопка переключения режимов. Если *S6* = 1 - режим "Автомат", если *S6* = 0 – режим "Наладка";

S7 – кнопка толчкового движения вперёд в режиме "Наладка";

S8 - кнопка толчкового движения назад в режиме "Наладка";

ПУУ – проектируемое устройство управления;

УВВ – устройство выдержки времени;

Т – сигнал включения отсчёта выдержки времени;

t –сигнал об окончании выдержки времени (*t* = 1, отсчёт выдержки времени закончился).

Схему управления необходимо реализовать на бесконтактных элементах И-НЕ.

Установка работает следующим образом.

Привод механизма каретки осуществляется асинхронным короткозамкнутым двигателем, реверсивное управление которым производится при помощи пускателей КВ (движение вперёд) КН (движение назад). Схема может работать в режимах "Автомат" и "Наладка". Выбор режима работы производится переключателем *S6*.

В режиме "Автомат" выполнение автоматического цикла перемещений между положениями 1, 2, 3 начинается при подаче кратковременной команды "Пуск" кнопкой *S4*. Перемещение каретки заканчивается остановкой в исходном положении после отработки всего цикла. Аварийное отключение в режиме "Автомат" осуществляется кнопкой *S5* "Стоп". Управление производится в функции положения. Контроль положения механизма выполняется с помощью конечных выключателей *S1, S2, S3*, расположенных в положениях 1, 2, 3. При отработке цикла в заданном положении выполняется остановка механизма с выдержкой времени.

В режиме "Наладка" осуществляется перемещение каретки вперёд или назад при нажатии и удержании кнопок *S7* (движение вперёд) или *S8* (движение назад).

При выполнении курсовой работы необходимо выполнить следующие пункты.

1. Запись условий работы устройства управления.
2. Блочный синтез.
3. Синтез отдельных блоков.
   - 3.1. Построение автоматной таблицы выходов и переходов.
   - 3.2. Минимизация памяти автомата (блока).
     - 3.2.1. Построение треугольной таблицы.
     - 3.2.2. Нахождение максимальных групп совместимости.
     - 3.2.3. Построение таблицы покрытия и определение минимального класса совместимости.
     - 3.2.4. Построение минимизированной автоматной таблицы.
   - 3.3. Кодирование внутренних состояний автомата.
     - 3.3.1. Составление внешнего π-разбиения.
     - 3.3.2. Составление внутреннего π-разбиения на основе множеств порядка 1.
     - 3.3.3. Проверка однозначности кодирования.
     - 3.3.4. Построение автоматного графа для проверки возможности возникновения и устранения критических состояний элементов памяти.
     - 3.3.5. Построение минимизированной закодированной исправленной автоматной таблицы.
   - 3.4. Построение функциональной схемы устройства управления (блока).

3.4.1. Определение функций возбуждения памяти.

3.4.2. Определение выходных функций.

3.4.3. Приведение функций в заданный базис.

3.4.4. Построение функциональной схемы.

4. Выбор элементов. Построение принципиальной схемы устройства управления.
5. Проверка правильности работы устройства управления.

Исходными данными к курсовой работе является цикл перемещения каретки механизма между положения 1, 2, 3. Выдержка времени в заданном положении обозначается буквой t.

Расчётно-пояснительная записка оформляется на стандартных листах печатной бумаги, заполняемых с одной стороны. На последнем листе приводится список используемой литературы, оформляемой в соответствии с ГОСТ 7.1 – 84, а на первом – содержание записки по разделам и основным параграфам разделов. Титульный лист оформляется по установленному в БНТУ образцу.

Записка обязательно должна иллюстрироваться схемами, таблицами, рисунками, которые отражают результаты отдельных этапов проектирования. Таблицы и рисунки в записке должны иметь наименования и нумерацию.

Принципиальные схемы должны строиться в соответствии с ГОСТ 2.755-87 «Обозначения условные графические в электрических схемах», ГОСТ 2.710-81 «Обозначения буквенно-цифровые в электрических схемах».

Для планомерной работы студента над курсовой работой в задании указываются ориентировочные сроки выполнения основных разделов.

Готовую курсовую работу студент сдаёт к установленному сроку руководителю для проверки. Если имеются замечания о правильности и полноте выполненной работы, курсовая работа возвращается студенту для доработки.

Студент защищает курсовую работу перед комиссией из преподавателей кафедры.

2 ЗАПИСЬ УСЛОВИЙ РАБОТЫ УСТРОЙСТВА УПРАВЛЕНИЯ

При записи условий работы устройства управления даётся конкретная словесная формулировка, которая устанавливает соответствие между входными и выходными сигналами. Присваиваются обозначения входным и выходным сигналам.

В качестве примера выполнения курсовой работы рассмотрим синтез схемы управления перемещения механизма по циклу 2 – 1 – 2 – 3 – 2t – 1 – 2.

На рисунке 2 представлена графическая модель цикла работы механизма.

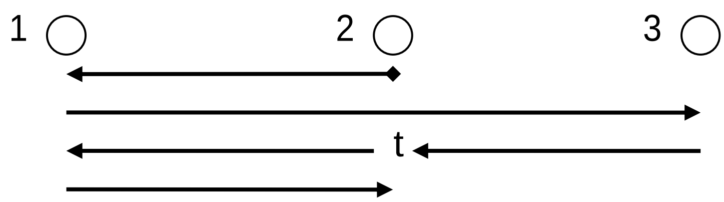


Рисунок 2 – Графическая модель цикла перемещения механизма

Перемещение каретки в направлении 1, 2, 3 принимают как перемещение вперёд, в направлении 3, 2, 1 – как перемещение назад.

В словесной формулировке необходимо описать последовательность поступления входных сигналов устройства управления (с указанием их значений) и указать, какие выходные сигналы при этом формируются.

Словесная формулировка условий работы устройства управления для рассматриваемого цикла будет иметь вид.

В режиме "Автомат" (сигнал $S6 = 1$) при нажатии на кнопку "Пуск" (сигнал $S4 = 1$) при нахождении каретки в исходном положении 2 (сигнал $S2 = 1$) включается контактор КН (назад) и включается двигатель М (сигнал $X_н = 1$). Начинается перемещение механизма из положения 2 в положение 1. При отпускании кнопки "Пуск" (сигнал $S4 = 0$) контактор КН остаётся включенным (сигнал $X_н = 1$) и продолжается движение механизма в направлении назад. Конечный выключатель $S2$ отключается (сигнал $S2 = 0$), контактор КН остаётся включенным ($Xн = 1$) и продолжается движение механизма в направлении назад. При достижении кареткой положения 1 срабатывает конечный выключатель $S1$ (сигнал $S1 = 1$), отключается контактор КН (сигнал $Xн = 0$), включается контактор КВ (сигнал $Xв = 1$) и начинается перемещение каретки вперёд в положение 2. Конечный выключатель $S1$ выключается ($S1 = 0$), контактор КВ остаётся включённым (сигнал $Xв = 1$), продолжается перемещение каретки вперёд. При достижении кареткой положения 2 срабатывает конечный выключатель $S2$ (сигнал $S2 = 1$), контактор КВ остаётся включенным (сигнал $Xв = 1$) и продолжается перемещение вперёд в положение 3. Конечный выключатель $S2$ отключается (сигнал $S2 = 0$), контактор КВ остаётся включенным (сигнал $Xв = 1$). При достижении кареткой положения 3 срабатывает конечный выключатель $S3$ (сигнал $S3 = 1$), отключается контактор КВ (сигнал $Xв = 0$), включается контактор КН (сигнал $Xн = 1$) и начинается перемещение механизма назад в положение 2. Конечный выключатель $S3$ выключается (сигнал $S3 = 0$), контактор КН остаётся включенным (сигнал $Xн = 1$), продолжается перемещение назад в положение 2. При достижении кареткой положения 2 срабатывает конечный выключатель $S2$ (сигнал $S2 = 1$), отключается контактор КН (сигнал $Xн = 0$), отключается двигатель М, останавливается каретка и по сигналу $T = 1$ включается устройство выдержки времени УВВ. Происходит отсчёт выдержки времени. По окончании отсчёта выдержки времени на выходе устройства выдержки времени УВВ появляется сигнал $t = 1$. При этом включается контактор КН (сигнал $Xн = 1$) и происходит перемещение каретки назад в положение 1. Конечный выключатель $S2$ отключается (сигнал $S2 = 0$), контактор КН остаётся включенным (сигнал $Xн = 1$) и продолжается перемещение каретки в направлении назад. Реле времени отключается (сигналы $T = 0$, $t = 0$), контактор КН остаётся включенным (сигнал $Xн = 1$) и продолжается перемещение каретки в положение 1. При достижении кареткой положения 1 срабатывает конечный выключатель $S1$ (сигнал $S1 = 1$), отключается контактор КН (сигнал $Xн = 0$), включается контактор КВ (сигнал $X_в$ = 1) и начинается перемещение каретки вперёд в положение 2. Конечный выключатель $S1$ (сигнал $S1$ = 0) отключается, контактор КВ остаётся включенным (сигнал $X_в$ = 1) и продолжается перемещение каретки в положение 2. При достижении кареткой положения 2 срабатывает конечный выключатель $S2$ (сигнал $S2$ = 1), отключается контактор КВ (сигнал $X_в$ = 0), выключается двигатель и каретка останавливается.

В режиме "Наладка" (сигнал $S6$ =0) перемещение механизма происходит при нажатии и удержании кнопок "Вперёд" (сигнал $S7$ = 1) или "Назад" (сигнал $S8$ = 1) независимо от конечных выключателей.

При нажатии на кнопку "Стоп" (сигнал $S5$ = 1) происходит остановка привода механизма в любом месте цикла.

3 БЛОЧНЫЙ СИНТЕЗ

В рассматриваемой схеме управления присутствует большое количество входных сигналов, что существенно затрудняет её формальное описание и синтез. Так как сигналы разделены по функциональному значению, целесообразно для упрощения проектирования разделить схему на отдельные функциональные блоки и наметить обмен информации между ними.

Блочная структура устройства управления представлена на рисунке 3.

Исходя из принципа действия, устройство управления можно разделить на четыре функциональных блока. Для связи между блоками введём дополнительные внутренние сигналы. Дадим описание каждого блока по отдельности.

Блок Б1 – блок, формирующий команду на отработку автоматического цикла.

Входные сигналы данного блока

$S4$ – сигнал с кнопки «Пуск»;

$S5$ – сигнал с кнопки «Стоп»;

S6 – сигнал выбора режима;

c – сигнал об окончании цикла перемещений.

Выходные сигналы

d – сигнал разрешения отработки автоматического цикла.

Блок работает следующим образом. При нажатии на кнопку «Пуск» *S4* и при отсутствии сигналов «Стоп» *S5* и *c* на выходе блока формируется сигнал *d* = 1, поступающий в блок Б3. При нажатии на кнопку «Стоп» *S5* или при поступлении сигнала об окончании цикла *c* = 1 сигнал *d* на выходе блока становится равным нулю и отработка цикла перемещений прекращается.

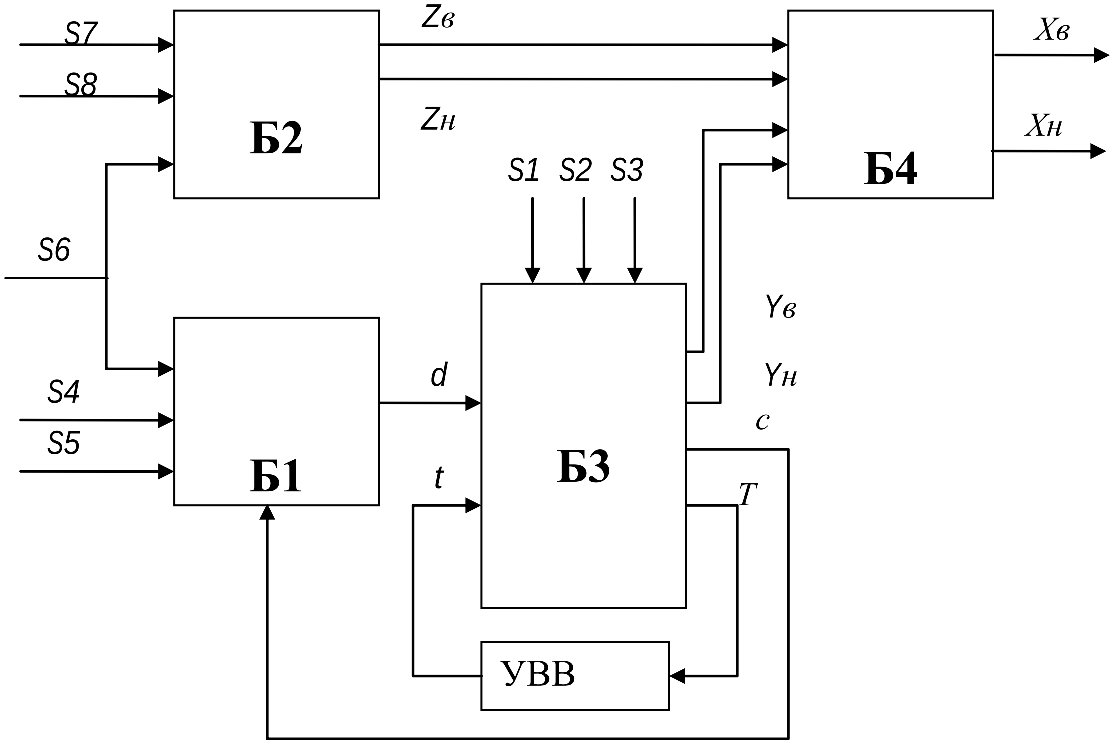


Рисунок 3 - Блочная схема проектируемого устройства

Блок Б2 – блок режима "Наладка".

Выполняет перемещение каретки в режиме наладки.

Входные сигналы

S6 – сигнал выбора режима;

S7 – сигнал с кнопки движения вперёд в режиме «Наладка»;

S8 – сигнал с кнопки движения назад в режиме «Наладка».

Выходные сигналы

$Z_в$ – сигнал перемещения вперёд;

$Z_н$ – сигнал перемещения назад.

При $S6$ = 0 и нажатой кнопке $S7$ происходит перемещение каретки вперёд ($Z_в = 1$, $Z_н = 0$). При нажатой S8 каретка движется назад ($Z_в = 0$, $Z_н = 1$). Одновременное нажатие кнопок недопустимо и должно блокироваться.

Блок Б3 – блок автоматического цикла.

Входные сигналы

$S1, S2, S3$ – сигналы с конечных выключателей в положениях 1, 2, 3;

d – сигнал разрешения отработки автоматического цикла;

t – сигнал об окончании выдержки времени.

Выходные сигналы

$Y_в, Y_н$ – сигналы перемещения вперёд, назад;

T – сигнал на включение выдержки времени;

c – сигнал об окончании цикла перемещений.

Этот блок начинает автоматическое выполнение цикла перемещений при поступлении сигнала "Начало цикла" (сигнал d = 1) из блока Б1. По включенному (сигнал S = 1) или по выключенному (сигнал S = 0) положению конечных выключателей $S1, S2, S3$ этот блок обеспечивает выполнение цикла заданных перемещений формируя команды на движение вперёд ($Y_в$ = 1) или назад ($Y_н$ = 1). Кроме этого в соответствующем положении в соответствии с циклом формирует сигнал на включение выдержки времени (сигнал $T = 1$). После отработки всего цикла перемещений на выходе блока формируются сигнал об окончания цикла (сигнал $c = 1$).

Блок Б4 – блок формирования сигналов силовых контакторов.

Производит формирование сигналов включения пускателей вперёд (сигнал $X_в$ = 1) или назад (сигнал $X_н$ = 1) из сигналов, поступающих из блоков Б2 ($Z_в, Z_н$) и Б3 ($Y_в, Y_н$). Кроме этого обеспечивает защиту от одновременного срабатывания контакторов $X_в$, $X_н$ для исключения возможности короткого замыкания.

4 СИНТЕЗ ОТДЕЛЬНЫХ БЛОКОВ

Из рассмотренных блоков наиболее сложный алгоритм работы имеет блок автоматического цикла Б3. Поэтому рассмотрим синтез данного блока первым.

4.1 Построение автоматной таблицы выходов и переходов блока Б3

Для построения автоматной таблицы на основании словесной формулировки составим первоначальную таблицу истинности. В данной таблице указывается последовательность поступления входных сигналов и отражается связь между входными и выходными сигналами. Столбцы данной таблицы соответствуют входным и выходным сигналам устройства управления, строки – входным наборам. Каждому входному набору ставится в соответствие определённое значение выходных сигналов.

Таблица 1 – Первоначальная таблица истинности

№ пер.	Входные сигналы					Выходные сигналы			
	d	*S1*	*S2*	*S3*	*t*	*Yв*	*Yн*	*T*	*c*
1	0	0	1	0	0	0	0	0	0
2	1	0	1	0	0	0	1	0	0
3	1	0	0	0	0	0	1	0	0
4	1	1	0	0	0	1	0	0	0
5	1	0	0	0	0	1	0	0	0
6	1	0	1	0	0	1	0	0	0
7	1	0	0	0	0	1	0	0	0
8	1	0	0	1	0	0	1	0	0
9	1	0	0	0	0	0	1	0	0
10	1	0	1	0	0	0	0	1	0
11	1	0	1	0	1	0	1	1	0
12	1	0	0	0	1	0	1	0	0
13	1	0	0	0	0	0	1	0	0
14	1	1	0	0	0	1	0	0	0
15	1	0	0	0	0	1	0	0	0
16	1	0	1	0	0	0	0	0	1
17	0	0	1	0	0	0	0	0	0

Из первоначальной таблицы истинности видно, что проектируемое устройство управления является дискретным автоматом с памятью. Это связано с тем, что при одинаковых входных наборах на выходе устройства формируются различные выходные сигналы. Для формального описания таких устройств используется автоматная таблица выходов и переходов [1, с 149 – 156].

Таблица переходов – это таблица, строки которой соответствуют исходным состояниям, столбцы соответствуют входным наборам, в клетках на пересечении строк и столбцов проставляется состояние, в которое переходит автомат из данного исходного состояния, определяемого строкой, под действием данной входной последовательности, определяемой столбцом.

Таблица выходов – это таблица, строки которой соответствуют исходным состояниям, столбцы соответствуют входным наборам, в клетках на пересечении строк и столбцов проставляется состояние выхода, формируемого в данном исходном состоянии под действием входной последовательности, определяемой столбцом.

Часто таблицу переходов и выходов совмещают и пользуются совмещённой автоматной таблицей.

Для полного и точного описания блока Б3 строится совмещённая избыточная автоматная таблица выходов и переходов. В данной таблице любое изменение входного или выходного сигнала будет приводить к переходу в новое состояние. Выходные сигналы *Yв*, *Yн* будем отмечать по модели Мура (в отдельных столбцах), сигналы *T, c* – по модели Мили (в клетках автоматной таблицы, *T* – первый сигнал после номера перехода, *c* – второй сигнал после номера перехода) [1, с. 144 – 149]. Переход в последующее состояние происходит по следующей входной последовательности в соответствии с циклом перемещений (изменение сигналов *S1, S2, S3*) или при исчезновении сигнала *d* = 1 (аварийная ситуация). Поэтому в каждой строке будет заполнено три клетки: приход в данное состояние и два выхода из него.

Автоматная таблица выходов и переходов для блока Б3 по заданному циклу представлена в таблице 2.

Таблица 2 - Автоматная таблица выходов и переходов блока Б3

		t				t				t				t				t					t				t			t			$У_в$	$У_н$
					S_3								S_3								S_3							S_3						
								S_2																	S_2									
																	S_1																	
	d																																	
0								1 00									0 00								0 00	0 00			0 00		0 00	0 00	0	0
1	2 00							1 00																	0 00								0	1
2	2 00															3 00																0 00	0	1
3	4 00															3 00	0 00																1	0
4	4 00							5 00																								0 00	1	0
5	6 00							5 00																	0 00								1	0
6	6 00			7 00																												0 00	1	0
7	8 00			7 00																									0 00				0	1
8	8 00							9 ~0																								0 00	0	1
9							10 10	9 10																	0 ~0								0	0
10		11 ~0					10 10																			0 ~0							0	1
11	12 00	11 00																													0 00		0	1
12	12 00															13 00																0 00	0	1
13	14 00															13 00	0 00																1	0
14	14 00							15 0~																								0 00	1	0
15								15 01																	0 0~								0	0
	A_0	A_1		A_2			A_3	A_4							A_5	A_6	A_7	A_8							A_9	A_{10}			A_{11}		A_{12}	A_{13}		

4.2 Минимизация памяти автомата

Целью минимизации автомата является получение эквивалентного автомата с наименьшим числом внутренних состояний [1, с. 165 - 173]. Чем меньше состояний в автоматной таблице, тем меньше элементов памяти необходимо для их кодирования. Минимизация заключается в нахождении совместимых состояний в автоматной таблице и их замене одним состоянием нового эквивалентного автомата.

Группа попарно совместимых состояний образует группу совместимости.

Группы совместимости, покрывающие все состояния исходного автомата, образуют класс совместимости.

Для получения минимального автомата необходимо построить минимальный класс совместимости, т. е. такой, который содержит минимум групп совместимости, группы максимальны по размеру и данное разбиение покрывает все состояния исходного автомата. Поиск совместимых состояний сводится к поиску совместимых строк автоматной таблицы и замене их одной строкой.

При поиске совместимых состояний в автоматной таблице используют два признака:

3) Из обоих состояний под действием любой допустимой входной последовательности осуществляется переходы в совместимые или непротиворечивые состояния.
4) Этим состояниям должны соответствовать непротиворечивые или совместимые выходы.

При этом, в зависимости от того, как накладываются переходы, два состояния могут быть условно совместимыми.

Два состояния называются условно совместимыми, если они совместимы при условии совместимости состояний, в которые автомат переходит из этих состояний.

Для определения совместимых состояний и построения минимального класса совместимости наиболее часто используется метод треугольной таблицы.

4.2.1 Построение треугольной таблицы

В треугольной таблице по горизонтали проставляются состояния от нулевого до предпоследнего, по вертикали – от последнего до первого (исключая нулевое состояние). Каждая клетка треугольной таблицы соответствует паре состояний. По автоматной таблице просматриваем все пары состояний в порядке, указанном в треугольной таблице справа на лево, снизу вверх и отмечаем совместимые (знак V), несовместимые (знак X) и условно совместимые состояния (записывается условие и проверяется, выполняется оно или нет).

Таблица 3 - Треугольная таблица

1	X														
2	X	V													
3	X	X	X												
4	X	X	X	V											
5	X	X	X	4,6 V	4,6 V										
6	X	X	X	4,6 V	V	V									
7	X	2,8 V	2,8 V	X	X	X	X								
8	X	X	V	X	X	X	X	V							
9	X	X	X	X	X	X	X	X	X						
10	X	V	V	X	X	X	X	V	V	X					
11	X	X	X	X	X	X	X	8,12 V	8,12 V	X	V				
12	X	X	X	X	X	X	X	8,12 V	V	X	V	V			
13	X	X	X	X	X	6,14 V	6,14 V	X	X	X	X	X	X		
14	X	X	X	X	X	X	V	X	X	X	X	X	X	V	
15	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

4.2.2 Нахождение максимальных групп совместимости

Для нахождения максимальных групп совместимости просматриваем треугольную таблицу и объединяем в группы попарно совместимые между собой состояния. Все состояния, входящие в группу совместимости должны быть между собой попарно совместимы!!! Группы совместимости должны отражать все совместимости в треугольной таблице.

Выписываем группы совместимости и обозначаем их буквами латинского алфавита.

6, 13, 14	- A
7, 8, 10, 11, 12	- B
6, 5, 13	- C
3, 4, 5, 6	- D
2, 7, 8, 10	- E
1, 2, 7, 10	- F

4.2.3 Построение таблицы покрытия и определение минимального класса совместимости

Минимальный класс совместимости представляет собой наилучший вариант объединения состояний исходной автоматной для получения таблицы с минимальным количеством внутренних состояний. Для определения минимального класса совместимости строится таблица покрытия. Строки таблицы покрытия соответствуют группам совместимости, столбцы – исходным состояниям. На пересечении строк (групп совместимости) со столбцами (исходными состояниями) отмечается вхождение данного состояния в данную группу совместимости.

На основании таблицы покрытия составляется формула покрытия. Для этого для каждого состояния записываем дизъюнкцию тех групп совместимости, в которые входит данное состояние. Дизъюнкции для каждого состояния связываются конъюнктивно. К полученной формуле применяются законы алгебры логики, раскрываются все скобки и получают в результате дизъюнктивно связанные классы совместимости. Из полученных классов необходимо выбрать тот, который содержит наименьшее количество групп совместимости. Это и будет минимальный класс совместимости.

Строим таблицу покрытия.

Таблица 4 - Таблица покрытия

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A							X							X	X	
B								X	X		X	X	X			
C						X	X							X		
D				X	X	X	X									
E			X					X	X		X					
F		X	X					X			X					

Формула покрытия будет иметь вид:

f = A*(A+C)*B*B*(B+E+F)*(B+E)*(B+E+F)*(A+C+D)*(C*D)*D*
D(E+F)*F = A*B*D*F

Как видно из таблицы покрытия состояния 0, 9, и 15 не вошли ни в одну из групп совместимости. Эти состояния будут образовывать отдельные группы совместимости, состоящие из одного состояния (эти состояния удалять нельзя!!!)

В полученном минимальном классе необходимо выполнить редукцию, т. е. убрать повторяющиеся состояния. Состояние 6 входит в группы A и D. Удалим данное состояние из группы А. Состояния 7, 10 входят в группы В и F. Удалим данные состояние из группы F.

При удалении состояний необходимо, чтобы в оставшихся группах совместимости не нарушались условия совместимости.

Запишем минимальный класс совместимости и обозначим полученные группы совместимости новыми номерами состояний эквивалентного автомата.

A	- 6, 13, 14	13, 14	- 5
B	- 7, 8, 10, 11, 12	7, 8, 10, 11, 12	- 3
D	- 3, 4, 5, 6	3, 4, 5, 6	- 2
F	- 1, 2, 7, 10	1, 2	- 1
	- 0	0	- 0
	- 9	9	- 4
	- 15	15	- 6

По полученным новым состояниям эквивалентного автомата можем построить минимизированную автоматную таблицу.

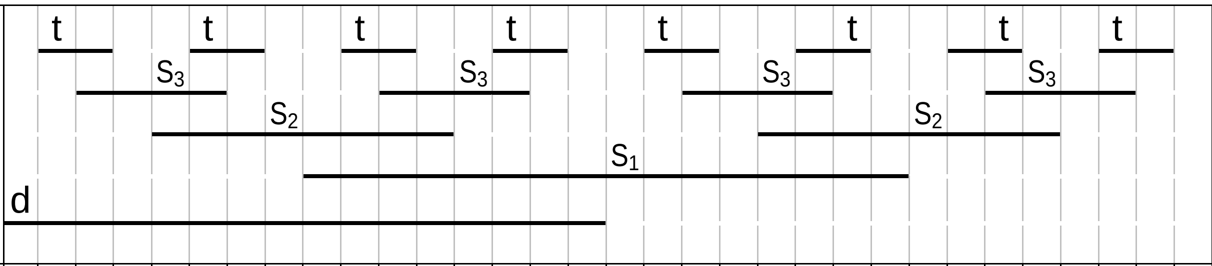
4.2.4 Построение минимизированной автоматной таблицы

Для построения минимизированной автоматной таблицы необходимо в исходной автоматной таблице старые номера состояний заменить на новые и строки с одинаковыми номерами объединить в одну строку путём наложения.

Правильность построения минимизированной автоматной таблицы можно проверить следующим образом. При объединении строк исходной автоматной таблицы в клетках на соответствующих наборах должны накладываться одинаковые номера. Если накладываются два разных номера, следовательно, в группу совместимости, соответствующую данному новому состоянию, входят несовместимые состояния. Это может произойти из-за неправильной записи группы совместимости или из-за нарушения условий совместимости.

Кроме этого, после построения минимизированной таблицы можно проверить, выполняются ли все переходы в соответствии с циклом перемещений, отмеченные в исходной автоматной таблице. Обычно в минимизированной автоматной таблице последовательность переходов отображается стрелками. Если все переходы выполняются, следовательно, минимизированная автоматная таблица построена верно.

Таблица 5 – Минимизированная автоматная таблица



	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	$Y_в$	$Y_н$
0					1 00		0 00	0 00	0 00	0 00	0 00	0 00	0	0
1	1 00				1 00	2 00		0 00				0 00	0	1
2	2 00		3 00		2 00	2 00	0 00	0 00				0 00	1	0
3	3 00	3 00	3 00	3 10	4 ~0	5 00			0 ~0	0 00	0 00	0 00	0	1
4				3 10	4 10			0 ~0					0	0
5	5 00				6 0~	5 00	0 00					0 00	1	0
6					6 01			0 0~					0	0

4.3 Кодирование внутренних состояний автомата

При кодировании каждому внутреннему состоянию автомата приписывается свой, неповторяющийся набор внутренних сигналов элементов памяти (определяется код состояния) [1, с. 177 - 183].

В зависимости от значения данной переменной кодирования все внутренние состояния автомата разделяются на два блока. В первый блок входят состояния, в которых данная переменная кодирования равна единице, во второй блок – состояния в которых переменная кодирования равна нулю.

Разбиением множества внутренних состояний *P* называют совокупность подмножеств состояний *Pi* таких, что их объединение есть Р. Подмножества называются блоками разбиения и отмечаются чертой.

Произведением двух разбиений *Пi* и *Пj* называют такое третье разбиение *Пij*, каждый блок которого образован пересечением блоков разбиений *Пi* и *Пj*.

Для однозначного кодирования необходимо, чтобы пересечение разбиений, в соответствии с которыми определяется распределение переменных кодирования, было бы равно нулевому разбиению.

В нулевом разбиении каждое состояние образует один блок, и, следовательно, все состояния между собой различимы.

При кодировании могут использоваться внешнее и внутреннее разбиения.

При кодировании состояний по внешнему разбиению значения переменных кодирования в соответствующих состояниях приравниваются к значениям выходных переменных в этих же состояниях. При этом, чем больше совпадают значения переменных кодирования с выходными сигналами, тем проще комбинационная схема выходов.

При кодировании по внутреннему разбиению в один блок включаются наиболее близкие по переходам состояния. Чем ближе по переходам состояния, включаемые в один блок, тем проще схема возбуждения памяти.

4.3.1 Составление внешнего π-разбиения

Выполним кодирование по внешнему разбиению (приравняем значения переменных кодирования *p1* и *p2* к значениям $Y_в$ $Y_н$. Получим разбиения следующего вида

$$\pi_{1Yв} = \overline{0,\ 1,\ 3,\ 4,\ 6} \qquad \overline{2,\ 5}$$
$$\pi_{2Yн} = \overline{0,\ 2,\ 4,\ 5,\ 6} \qquad \overline{1,\ 3}.$$

Определим пересечение полученных разбиений.

$$\pi_1 \cdot \pi_2 = (\overline{0,\ 1,\ 3,\ 4,\ 6} \qquad \overline{2,\ 5}) \cdot (\overline{0,\ 2,\ 4,\ 5,\ 6} \qquad \overline{1,\ 3}) = \overline{0,\ 4,\ 6} \quad \overline{1,\ 3} \quad \overline{2,\ 5}$$

Из полученного пересечения видно, что неразличимыми по внешнему разбиению остались 1 и 3 состояния, 2 и 5 состояния и 0, 4 и 6 состояния (имеют одинаковый код). При этом для того чтобы различить три неразличимых состояния, необходимо ввести ещё две переменных кодирования. Таким образом, при использовании внешнего разбиения для кодирования всех семи состояний необходимо четыре переменных кодирования. Данный вариант кодирования является неоптимальным, так как для кодирования семи состояний достаточно трёх переменных кодирования. Поэтому, в данном случае целесообразно использовать только внутреннее разбиение для трёх переменных кодирования.

В общем случае, если при кодировании внутренних состояний автомата с использованием внешнего разбиения достаточно трёх переменных кодирования, необходимо использовать внешнее разбиение. Если трёх переменных недостаточно, необходимо использовать внутреннее разбиение и три переменных кодирования. Это верно для устройств, которые реализуются на бесконтактных элементах и содержат в минимизированной автоматной таблице не больше восьми состояний. Если состояний больше или схема реализуется на контактных элементах, необходимо использовать внешнее разбиение.

4.3.2 Составление внутреннего π-разбиения на основе множеств порядка 1

Для уменьшения переменных кодирования для нашей схемы используем внутреннее разбиение.

Воспользуемся методом на основе множеств порядка "1".

Множеством порядка единица для состояния *pi* называется множество тех состояний, из которых возможен непосредственный переход в состояние *pi*.

Порядок определения внутреннего разбиения.

5) Для каждого состояния выписываем множество порядка единица.

Выписываем номера состояний, из которых переходим в данное состояние.

6) Выписываем объединённые наборы таких состояний, которые имеют максимальное пересечение множеств порядка единица.

7) Из объединённых наборов составляем двухблочное разбиение таким образом, чтобы это разбиение содержало в разных блоках состояния, не различимые по внешнему разбиению.
8) Проверяем, равно ли пересечение полученных разбиений нулевому разбиению.

Запишем множества порядка единицы:

0 – 1, 2, 3, 4, 5, 6

1- 0, 1

2- 1, 2

3- 2, 3, 4

4- 3, 4

5- 3, 5

6- 5, 6

Запишем объединённые наборы:

0, 3 – пересекаются по трём состояниям;

0, 1 0, 2 0, 4 0, 5 0, 6 3, 4 - пересекаются по двум состояниям.

Примем разбиения следующего вида:

$\pi_1 = \overline{0,\ 5,\ 6}\quad \overline{1,\ 2,\ 3,\ 4}$

$\pi_2 = \overline{0,\ 1,\ 4,\ 6}\quad \overline{2,\ 3,\ 5}$

$\pi_3 = \overline{0,\ 1,\ 2}\quad \overline{3,\ 4,\ 5,\ 6}$

4.3.3 Проверка однозначности кодирования

Для проверки однозначности кодирования необходимо определить произведение полученных разбиений.

$$\pi_1 * \pi_2 * \pi_3 = \bar{0}\quad \bar{1}\quad \bar{2}\quad \bar{3}\quad \bar{4}\quad \bar{5}\quad \bar{6}$$

Так как произведение полученных разбиений равно нулевому разбиению (каждое состояние образует отдельный блок), следовательно, каждое состояние имеет свой индивидуальный неповторяющийся код.

Распределим значения переменных кодирования по состояниям.

Минимизированная закодированная автоматная таблица будет иметь вид.

Таблица 6 – Минимизированная закодированная автоматная таблица

																																	Yв	Yн	p1	p2	p3
		t	—			t	—			t	—			t	—			t	—			t	—			t	—			t	—						
			S_3 —	—	S_3	—					—	—	S_3	—					—	—	S_3	—					—	S_3	—	—							
					—	—	—	S_2	—	—	—	—									—	—	—	—	S_2	—	—	—									
									—	—	—	—	—	—	—	—	S_1	—	—	—	—	—	—	—													
	d	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—																					
0								1 00									0 00								0 00	0 00			0 00		0 00	0 00	0	0	0	0	0
1	1 00							1 00								2 00									0 00							0 00	0	1	1	0	0
2	2 00			3 00				2 00								2 00	0 00								0 00							0 00	1	0	1	1	0
3	3 00	3 00		3 00			3 10	4 ~0								5 00										0 ~0			0 00		0 00	0 00	0	1	1	1	1
4							3 10	4 10																	0 ~0								0	0	1	0	1
5	5 00							6 0~								5 00	0 00															0 00	1	0	0	1	1
6								6 01																	0 0~								0	0	0	0	1
	A_0	A_1		A_2			A_3	A_4								A_5	A_6								A_7	A_8			A_9		A_{10}	A_{11}					

4.3.4 Построение автоматного графа для проверки возможности возникновения и устранения критических состояний элементов памяти

Полученную минимизированную закодированную автоматную таблицу необходимо проанализировать с точки зрения возможности возникновения состязаний элементов памяти при переходах [1, с. 209 – 212].

Состязания элементов памяти возникают в том случае, если при переходе из состояния в состояние меняют своё значение два и более разряда кода. При этом, за счёт наличия задержек на срабатывание элементов памяти один разряд кода может переключиться быстрее. Это приводит к появлению промежуточных кодов, которые могут привести к нарушению алгоритма работы автомата.

Состязания элементов памяти могут быть критическими и некритическими.

Критические – это состязания, при которых нарушается алгоритм работы устройства (автомат застревает в непредусмотренном состоянии или осуществляются непредусмотренные переходы с формированием другой выходной последовательности).

Некритические – это состязания, при которых автомат переходит в другое используемое состояние или в неиспользуемое состояние, но при этом алгоритм работы не нарушается, формируется заданная выходная последовательность.

Так как проектируемое устройство относится к асинхронным автоматам, то необходимо устранить как критические, так и некритические состязания элементов памяти.

Для определения возможности возникновения критических состязаний элементов памяти по закодированной минимизированной автоматной таблице строится автоматный граф. Вершины автоматного графа соответствуют состояниям автомата, дуги – переходам из состояния в состояние. Для дискретного автомата Мура у вершин графа проставляются коды состояний и соответствующие им выходные наборы, над дугами отмечаются входные наборы, под действием которых осуществляется данный переход. Просматриваются все переходы и выписываются те из них, на которых меняют своё значения два и более разряда кода, т. е. возникает состязание элементов памяти.

Граф состояний и переходов представлен на рисунке 4.

Из автоматного графа видно, что состязания элементов памяти возникают на следующих переходах:

Переход	Переключение кода	Наборы
$2 \rightarrow 0$	$110 \rightarrow 000$	A6, A7, A11
$3 \rightarrow 0$	$111 \rightarrow 000$	A8, A9, A10, A11
$4 \rightarrow 0$	$101 \rightarrow 000$	A7
$5 \rightarrow 0$	$011 \rightarrow 000$	A6, A11

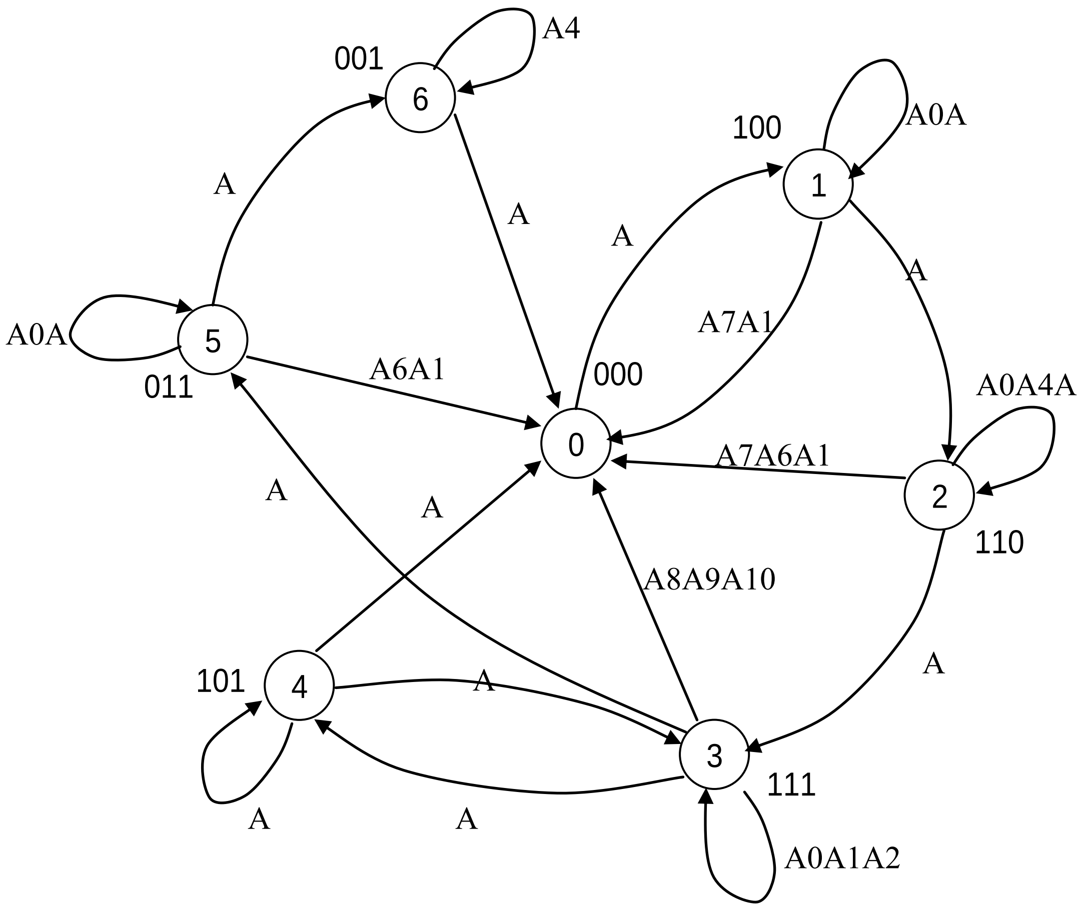


Рисунок 4 - Граф состояний и переходов

Для устранения критических состязаний элементов памяти выполняют следующие действия.

4) Выполняют вместо заданного перехода переход через другое неустойчивое состояние с последовательной сменой кода.
5) Если такой переход выполнить невозможно, вводят дополнительные состояния с неиспользуемыми кодами и выполняют переход через эти состояния.
6) Если неиспользуемые коды отсутствуют, вводят дополнительную переменную кодирования для получения дополнительных кодов и выполняют переход через дополнительные состояния.

Устраним состязания, выполнив переходы через другие неустойчивые состояния. Для этого, в тех состояниях, через которые будем осуществлять переходы, наборы, на которых исправляются переходы, должны быть не заняты. Выполним переходы следующим образом.

Новые переходы	Переключение кода
2 → 1 → 0	110 → 100 → 000
3 → 2 → 1 → 0	111 → 110 → 100 → 000
4 → 1 → 0	101 → 100 → 000
5 → 6 → 0	011 → 001 → 000

Строим исправленный автоматный граф.

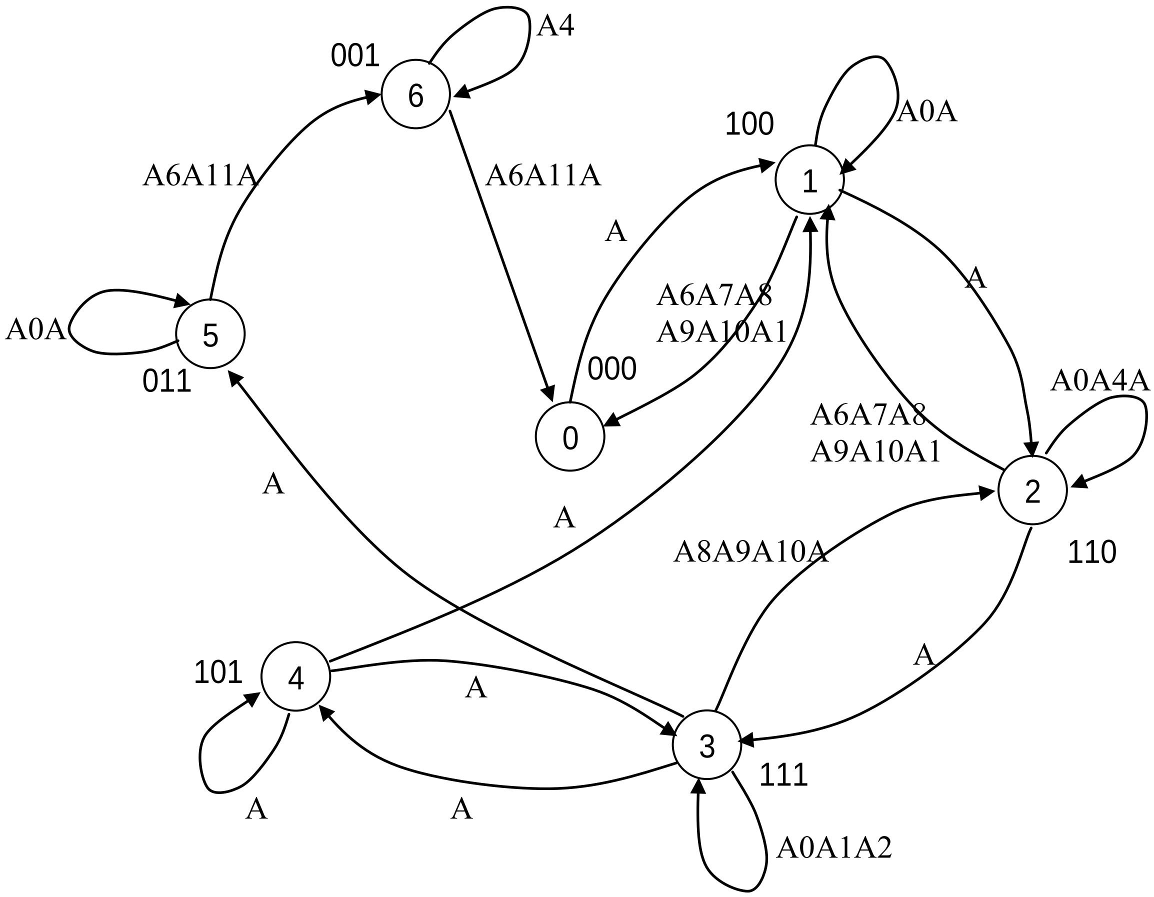


Рисунок 5 - Исправленный граф состояний и переходов

4.3.5 Построение минимизированной закодированной исправленной автоматной таблицы

На основании исправленного графа можем построить исправленную минимизированную закодированную автоматную таблицу.

Таблица 7 – Исправленная закодированная минимизированная автоматная таблица

		t				t				t				t				t					t				t			t							
					S3								S3								S3							S3									
								S2																	S2												
																	S1																Yв	Yи	p1	p2	p3
	d																																				
0								1 00									0 00								0 00	0 00			0 00		0 00	0 00	0	0	0	0	0
1	1 00							1 00								2 00	0 00								0 00	0 ~0			0 00		0 00	0 00	0	1	1	0	0
2	2 00			3 00				2 00								2 00	1 00								1 00	1 ~0			1 00		1 00	1 00	1	0	1	1	0
3	3 00	3 00		3 00			3 10	4 ~0								5 00										2 ~0			2 00		2 00	2 00	0	1	1	1	1
4							3 10	4 10																	1 ~0								0	0	1	0	1
5	5 00							6 0~								5 00	6 00															6 00	1	0	0	1	1
6								6 01									0 00								0 0~							0 00	0	0	0	0	1
	A_0	A_1		A_2			A_3	A_4								A_5	A_6								A_7	A_8			A_9		A_{10}	A_{11}					

4.4 Построение функциональной схемы устройства управления

Для построения функциональной схемы устройства управления необходимо записать аналитические выражения для функций возбуждения памяти и выходных функций. В бесконтактных схемах память может реализовываться с помощью петель обратных или на триггерах [1, с. 183 – 188].

4.4.1 Определение функций возбуждения памяти и выходных функций при реализации памяти петлями обратных связей.

Для построения функциональной схемы на бесконтактных элементах И-НЕ с реализацией памяти петлями обратных связей необходимо выполнить следующие действия.

1. Строится карта Карно для каждой переменной кодирования. При этом, для реализации петель обратных связей, в карту Карно включаются как входные переменные, так и внутренние переменные.

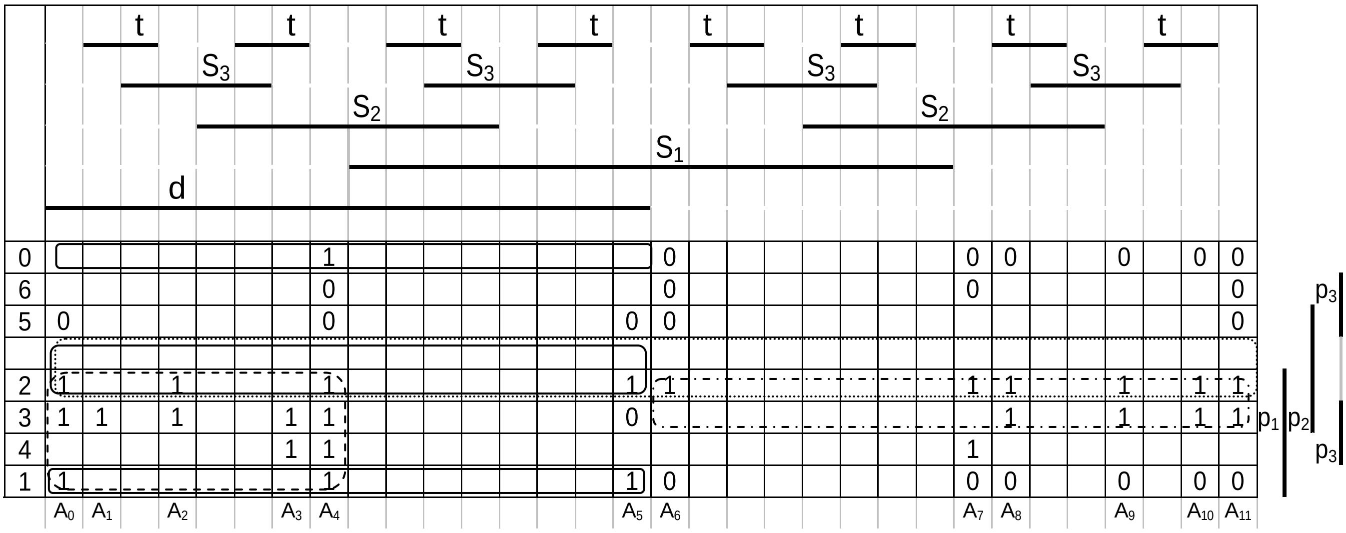
2. По кодам распределяются состояния автоматной таблицы по строкам карты Карно.
3. Заполняются клетки карты Карно. Для этого необходимо номер состояния в соответствующей клетке заменить на значение данной переменной в данном состоянии.
4. В соответствии со стандартными правилами строятся контуры и записываются минимизированные выражения для функций возбуждения памяти.
5. Если необходимо, строятся карты Карно для выходных переменных (если применялось только внутреннее разбиение). При этом, для модели Мура указывается зависимость выходов только от внутренних переменных, для модели Мили – зависимость выходов от входов и внутренних переменных. Поданным картам Карно записываются минимизированные выражения для функций выхода.
6. Полученные выражения преобразуются в заданный базис.
7. По полученным выражениям строится комбинационная схема возбуждения памяти (КСВП) и комбинационная схема выходов (КСВ).

Для того чтобы найти минимизированное выражение функции в виде ДНФ по карте Карно необходимо.

4. Все единицы карты Карно охватить минимальным числом контуров. При этом, контуры должны быть максимальны по размеру, включать $2^к$ клеток (1, 2, 4, 8, 16, 32, 64 и т. д.) и быть симметричными относительно осей симметрии всей карты Карно, половины карты Карно, четверти и т. д.
5. Каждый контур даёт в общее выражение минимизированной функции конъюнкцию тех переменных, которые в данном контуре не меняют своего значения. Причём, если в контуре переменная в наборах равна 1, то в конъюнкцию записывается сама переменная, если равна 0 – то записывается её инверсия.
6. Минимизированное выражение получается как дизъюнкция конъюнкций, соответствующих каждому контуру.

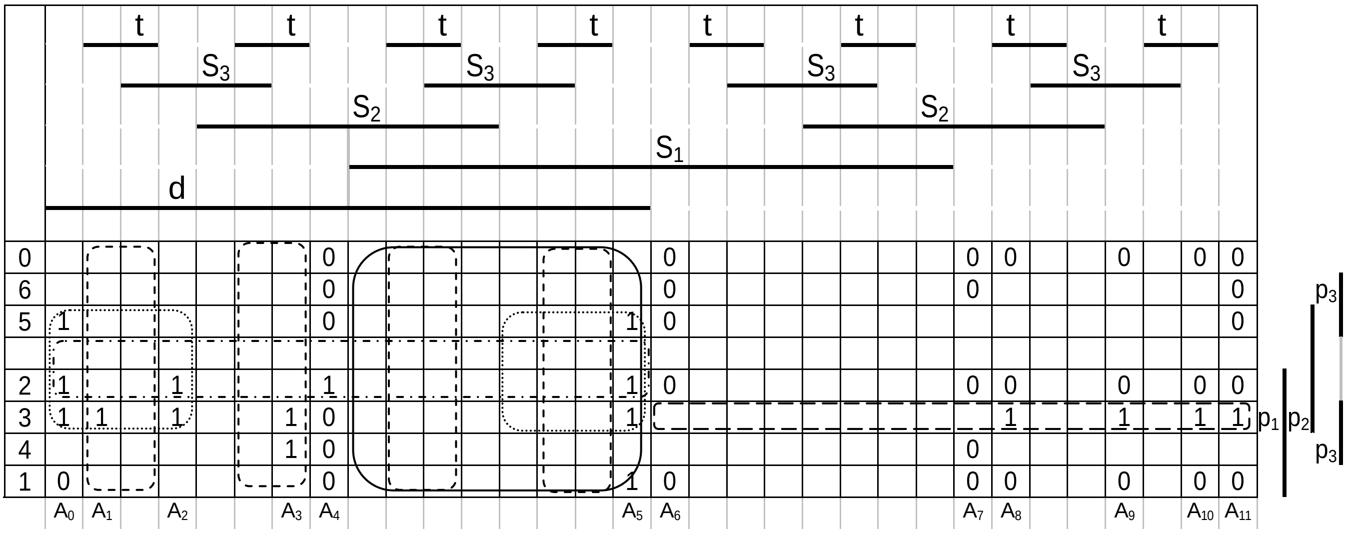
Построим карты Карно для переменных *p1, p2, p3* и определим функции возбуждения памяти.

Таблица 8 – Карта Карно для переменной *p1*



$$p1 = d \cdot \overline{p3} + d \cdot \overline{S1} \cdot p1 + \overline{d} \cdot p1 \cdot p2 + p2 \cdot \overline{p3}$$

Таблица 9 - Карта Карно для переменной *p2*



$$p2 = d \cdot S1 + d \cdot t + d \cdot \overline{S2} \cdot p2 + d \cdot p2 \cdot \overline{p3} + \overline{d} \cdot p1 \cdot p2 \cdot p3$$

Таблица 10 - Карта Карно для переменной *p3*

0								0									0								0	0			0		0	0
6								1									0								0							0
5	1							1								1	1															1
2	0			1				0								0	0								0	0			0		0	0
3	1	1		1			1	1								1										0			0		0	0
4							1	1																	0							
1	0							0								0	0								0	0			0		0	0
	A_0	A_1		A_2			A_3	A_4								A_5	A_6								A_7	A_8			A_9		A_{10}	A_{11}

$$p3 = d \cdot p3 + d \cdot S3 + \overline{p1} \cdot p2$$

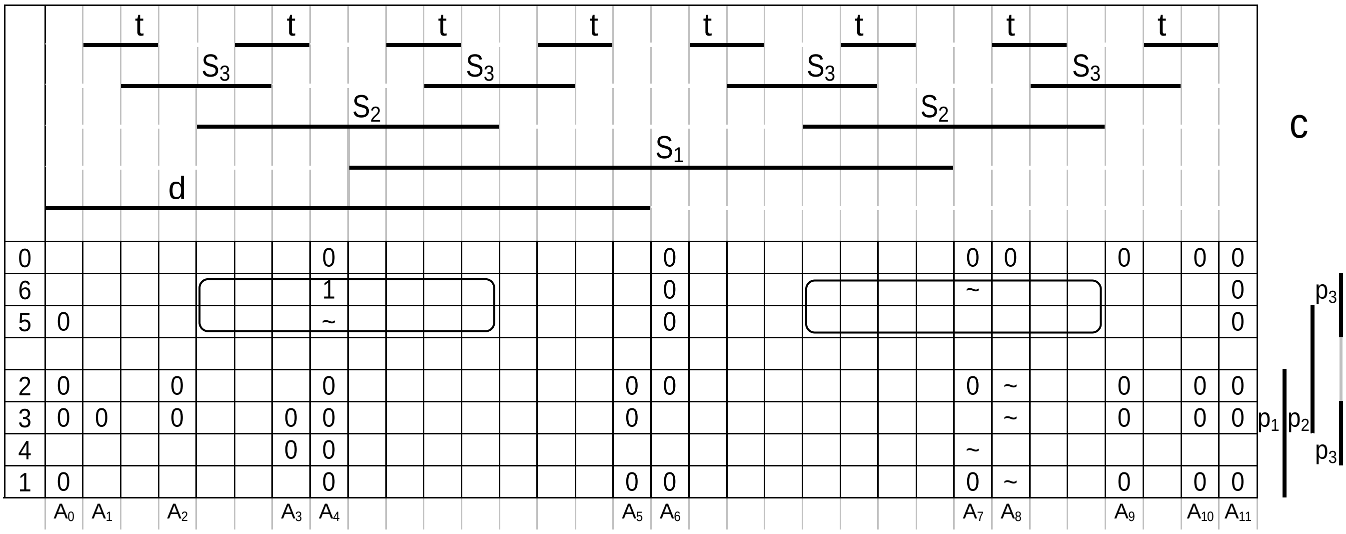
Построим карты Карно и запишем выражения для выходных переменных *T* и *c*.

Таблица 11 - Карта Карно для переменной *T*

0								0									0								0	0			0		0	0
6								0									0								0							0
5	0							0									0															0
2	0			0				0								0	0								0	~			0		0	0
3	0	0		0			1	~								0										~			0		0	0
4							1	1																	~							
1	0							0								0	0								0	~			0		0	0
	A_0	A_1		A_2			A_3	A_4								A_5	A_6								A_7	A_8			A_9		A_{10}	A_{11}

$$T = S2 \cdot p1 \cdot p3$$

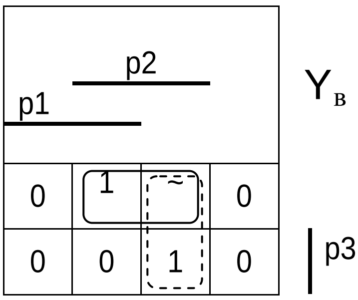
Таблица 12 - Карта Карно для переменной *c*



$$c = S2 \cdot \overline{p1} \cdot p3$$

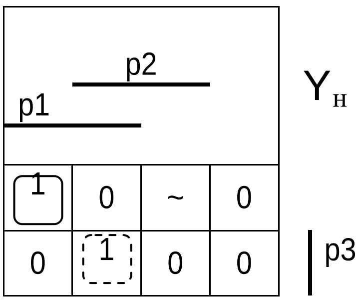
Для выходных переменных $Y_в$, $Y_н$ строятся карты Карно, в которых указывается их зависимость только от внутренних переменных *p1, p2, p3*.

Таблица 13 - Карта Карно для переменной $Y_в$



$$Y_в = p2 \cdot \overline{p3} + p2 \cdot \overline{p1}$$

Таблица 14 - Карта Карно для переменной $Y_н$



$$Y_н = p1 \cdot \overline{p2} \cdot \overline{p3} + p1 \cdot p2 \cdot p3$$

4.4.2 Определение функций возбуждения памяти при реализации памяти на RS-триггерах

Для реализации памяти на триггерах необходимо записать функцию возбуждения памяти для каждого входа триггера. Для этого необходимо выполнить следующие действия.

1. Закодированная автоматная таблица представляется в виде карт Карно. При этом, для каждой переменной кодирования строится две карты Карно, для входа R и входа S триггера.
2. Заполняются карты Карно для каждой переменной кодирования. Вместо номера состояния в клетку вписывается значение входного сигнала триггера, которое обеспечивает его переключение из исходного состояния, определяемого строкой, в последующее, определяемое номером в клетке.
3. По каждой заполненной карте Карно записывается минимизированное логическое выражение функции возбуждения памяти для искомой переменной.
4. Полученные выражения преобразуются в заданный базис.
5. Строится комбинационная схема возбуждения памяти.
6. Строятся карты Карно для выходных переменных, записываются выражения выходных функций в заданном базисе и строится комбинационная схема выходов.

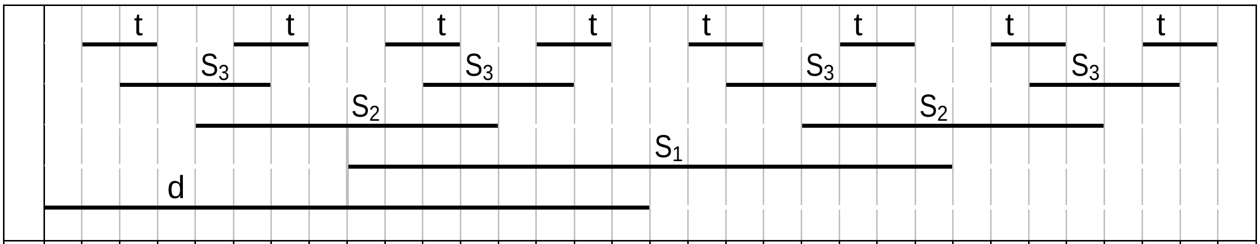
Для заполнения карт Карно можно использовать таблицу переходов RS-триггера.

$X_{исходное}$	RS	$X_{послед}$
0	~ 0	0
0	0 1	1
1	1 0	0
1	0 ~	1

Так как у нас для кодирования используются три переменных кодирования, то для каждой из них строим две карты Карно.

Строим карты Карно для входов триггеров *R1, S1, R2, S2, R3, S3.*

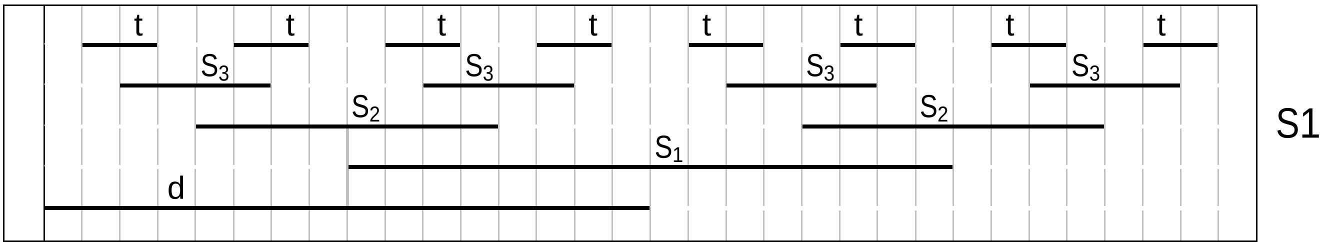
Таблица 15 – Карта Карно для переменной *R1*



0								0									~								~	~			~		~	~
6								~									~								~							~
5	~							~								~	~															~
2	0			0				0								0	0								0	0			0		0	0
3	0	0		0			0	0								1										0			0		0	0
4							0	0																	0							
1	0							0								0	1								1	1			1		1	1
	A_0	A_1		A_2			A_3	A_4								A_5	A_6								A_7	A_8			A_9		A_{10}	A_{11}

$$R1 = S1 \cdot p3 + \overline{d} \cdot \overline{p2} \cdot \overline{p3}$$

Таблица 16 - Карта Карно для переменной *S1*



0								1									0								0	0			0		0	0
6								0									0								0							0
5	0							0								0	0															0
2	~			~				~								~	~								~	~			~		~	~
3	~	~		~			~	~								0										~			~		~	~
4							~	~																	~							
1	~							~								~	0								0	0			0		0	0
	A_0	A_1		A_2			A_3	A_4								A_5	A_6								A_7	A_8			A_9		A_{10}	A_{11}

$$S1 = d \cdot \overline{p3}$$

Таблица 17 - Карта Карно для переменной *R2*

0								~									~								~	~			~		~	~
6								~									~								~							~
5	0							1								0	1															1
2	0			0				0								0	1								1	1			1		1	1
3	0	0		0			0	1								0										0			0		0	0
4							0	~																								
1	~							~								0	~								~	~			~		~	~
	A_0	A_1		A_2			A_3	A_4								A_5	A_6								A_7	A_8			A_9		A_{10}	A_{11}

$$R2=\overline{d}\cdot\overline{p1}+\overline{d}\cdot\overline{p3}+S2\cdot\overline{t}\cdot p3$$

Таблица 18 - Карта Карно для переменной *S2*

0								0									0								0	0			0		0	0
6								0									0								0							0
5	~							0								~	0															0
2	~			~				~								~	0								0	0			0		0	0
3	~	~		~			~	0								~										~			~		~	~
4							1	0																	0							
1	0							0								1	0								0	0			0		0	0
	A_0	A_1		A_2			A_3	A_4								A_5	A_6								A_7	A_8			A_9		A_{10}	A_{11}

$$S2=d\cdot S1+d\cdot t$$

Таблица 19 - Карта Карно для переменной *R3*

0								~									~								~	~			~		~	~
6								0									1								1							1
5	0							0								0	0															0
2	~			0				~								~	~								~	~			~		~	~
3	0	0		0			0	0								0										1			1		1	1
4							0	0																	1							
1	~							~								~	~								~	~			~		~	~
	A_0	A_1		A_2			A_3	A_4								A_5	A_6								A_7	A_8			A_9		A_{10}	A_{11}

$$R3=\overline{d}\cdot\overline{p1}+\overline{d}\cdot\overline{p2}$$

Таблица 20 - Карта Карно для переменной *S3*

			t				t				t				t			t				t				t				t			
					S_3							S_3									S_3							S_3					
									S_2															S_2									S3
																	S_1																
				d																													
0								0									0								0	0			0		0	0	
6								~									0								0							0	p_3
5	~							~								~	~															~	
2	0			1				0								0	0								0	0			0		0	0	
3	~	~		~			~	~								~										0			0		0	0	p_1 p_2
4							~	~																	0								p_3
1	0							0								0	0								0	0			0		0	0	
	A_0	A_1		A_2			A_3	A_4								A_5	A_6								A_7	A_8			A_9		A_{10}	A_{11}	

$$S3 = d \cdot S3$$

4.4.3 Приведение функций в заданный базис

Так как схему необходимо реализовать на элементах И-НЕ необходимо полученные выражения преобразовать в базис операции Шеффера [1, с.91 – 93]. Операция Шеффера имеет вид

$$f = a \,|\, b = \overline{a \cdot b}.$$

Для преобразования выражения в базис операции Шеффера необходимо взять двойную инверсию от выражения, после чего раскрыть нижнюю инверсию, используя закон де-Моргана. В полученном выражении ввести обозначение операции Шеффера.

Если исходная функция имела общую инверсию, то эту инверсию можно реализовать, используя выражения:

$$a \,|\, a = \bar{a} \qquad a \,|\, 1 = \bar{a}.$$

Запишем полученные выражения для внутренних и выходных переменных в базисе операции Шеффера.

$$p1 = \overline{\overline{d \cdot \overline{p3} + d \cdot \overline{S1} \cdot p1 + \overline{d} \cdot p1 \cdot p2 + p2 \cdot \overline{p3}}} = \overline{\overline{d \cdot \overline{p3}} \cdot \overline{d \cdot \overline{S1} \cdot p1} \cdot \overline{\overline{d} \cdot p1 \cdot p2} \cdot \overline{p2 \cdot \overline{p3}}} =$$
$$= (d \,|\, \overline{p3}) \,|\, (d \,|\, \overline{S1} \,|\, p1) \,|\, (\overline{d} \,|\, p1 \,|\, p2) \,|\, (p2 \,|\, \overline{p3})$$

$$p2=\overline{\overline{d\cdot S1+d\cdot t+d\cdot\overline{S2}\cdot p2+d\cdot p2\cdot\overline{p3}+\overline{d}\cdot p1\cdot p2\cdot p3}}=$$

$$=\overline{\overline{d\cdot S1}\cdot\overline{d\cdot t}\cdot\overline{d\cdot\overline{S2}\cdot p2}\cdot\overline{d\cdot p2\cdot\overline{p3}}\cdot\overline{\overline{d}\cdot p1\cdot p2\cdot p3}}=$$

$$=(d\mid S1)\mid(d\mid t)\mid(d\mid\overline{S2}\mid p2)\mid(d\mid p2\mid\overline{p3})\mid(\overline{d}\mid p1\mid p2\mid p3)$$

$$p3=\overline{\overline{d\cdot p3+d\cdot S3+\overline{p1}\cdot p2}}=\overline{\overline{d\cdot p3}\cdot\overline{d\cdot S3}\cdot\overline{\overline{p1}\cdot p2}}=(d\mid p3)\mid(d\mid S3)\mid(\overline{p1}\mid p2)$$

$$R1=\overline{\overline{S1\cdot p3+\overline{d}\cdot\overline{p2}\cdot\overline{p3}}}=\overline{\overline{S1\cdot p3}\cdot\overline{\overline{d}\cdot\overline{p2}\cdot\overline{p3}}}=(S1\mid p3)\mid(\overline{d}\mid\overline{p2}\mid\overline{p3})$$

$$S1=\overline{\overline{d\cdot\overline{p3}}}=\overline{d\mid\overline{p3}}=(d\mid\overline{p3})\mid(d\mid\overline{p3})$$

$$R2=\overline{\overline{\overline{d}\cdot\overline{p1}+\overline{d}\cdot\overline{p3}+S2\cdot\overline{t}\cdot p3}}=\overline{\overline{\overline{d}\cdot\overline{p1}}\cdot\overline{\overline{d}\cdot\overline{p3}}\cdot\overline{S2\cdot\overline{t}\cdot p3}}=$$

$$=(\overline{d}\mid\overline{p1})\mid(\overline{d}\mid\overline{p3})\mid(S2\mid\overline{t}\mid p3)$$

$$S2=\overline{\overline{d\cdot S1+d\cdot t}}=\overline{\overline{d\cdot S1}\cdot\overline{d\cdot t}}=(d\mid S1)\mid(d\mid t)$$

$$R3=\overline{\overline{\overline{d}\cdot\overline{p1}+\overline{d}\cdot\overline{p2}}}=\overline{\overline{\overline{d}\cdot\overline{p1}}\cdot\overline{\overline{d}\cdot\overline{p2}}}=(\overline{d}\mid\overline{p1})\mid(\overline{d}\mid\overline{p2})$$

$$S3=\overline{\overline{d\cdot S3}}=\overline{d\mid S3}=(d\mid S3)\mid(d\mid S3)$$

$$T=\overline{\overline{S2\cdot p1\cdot p3}}=\overline{S2\mid p1\mid p3}=(S2\mid p1\mid p3)\mid(S2\mid p1\mid p3)$$

$$c=\overline{\overline{S2\cdot\overline{p1}\cdot p3}}=\overline{S2\mid\overline{p1}\mid p3}=(S2\mid\overline{p1}\mid p3)\mid(S2\mid\overline{p1}\mid p3)$$

$$Y_{в}=\overline{\overline{p2\cdot\overline{p3}+p2\cdot\overline{p1}}}=\overline{\overline{p2\cdot\overline{p3}}\cdot\overline{p2\cdot\overline{p1}}}=(p2\mid\overline{p3})\mid(p2\mid\overline{p1})$$

$$Y_{н}=\overline{\overline{p1\cdot\overline{p2}\cdot\overline{p3}+p1\cdot p2\cdot p2}}=\overline{\overline{p1\cdot\overline{p2}\cdot\overline{p3}}\cdot\overline{p1\cdot p2\cdot p2}}=$$

$$=(p1\mid\overline{p2}\mid\overline{p3})\mid(p1\mid p2\mid p3)$$

По полученным выражениям строим функциональную схему блока Б3.

Функциональная схема блока Б3 на бесконтактных элементах И-НЕ с реализацией памяти петлями обратных связей представлена на рисунке 6.

Функциональная схема блока Б3 на бесконтактных элементах И-НЕ с реализацией памяти на RS-триггерах представлена на рисунке 7.

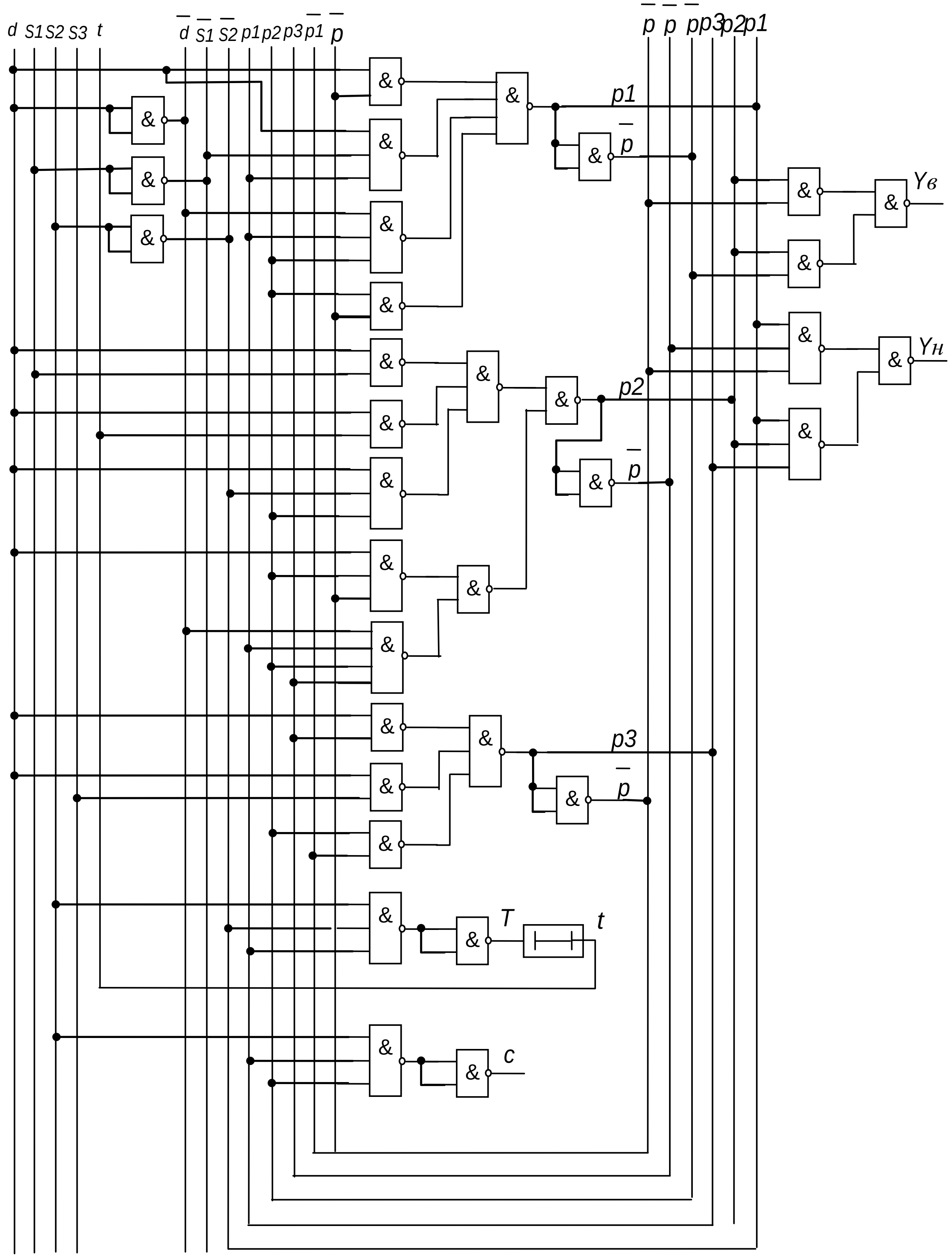


Рисунок 6 – Функциональная схема блока Б3 на элементах И-НЕ с реализацией памяти петлями обратных связей

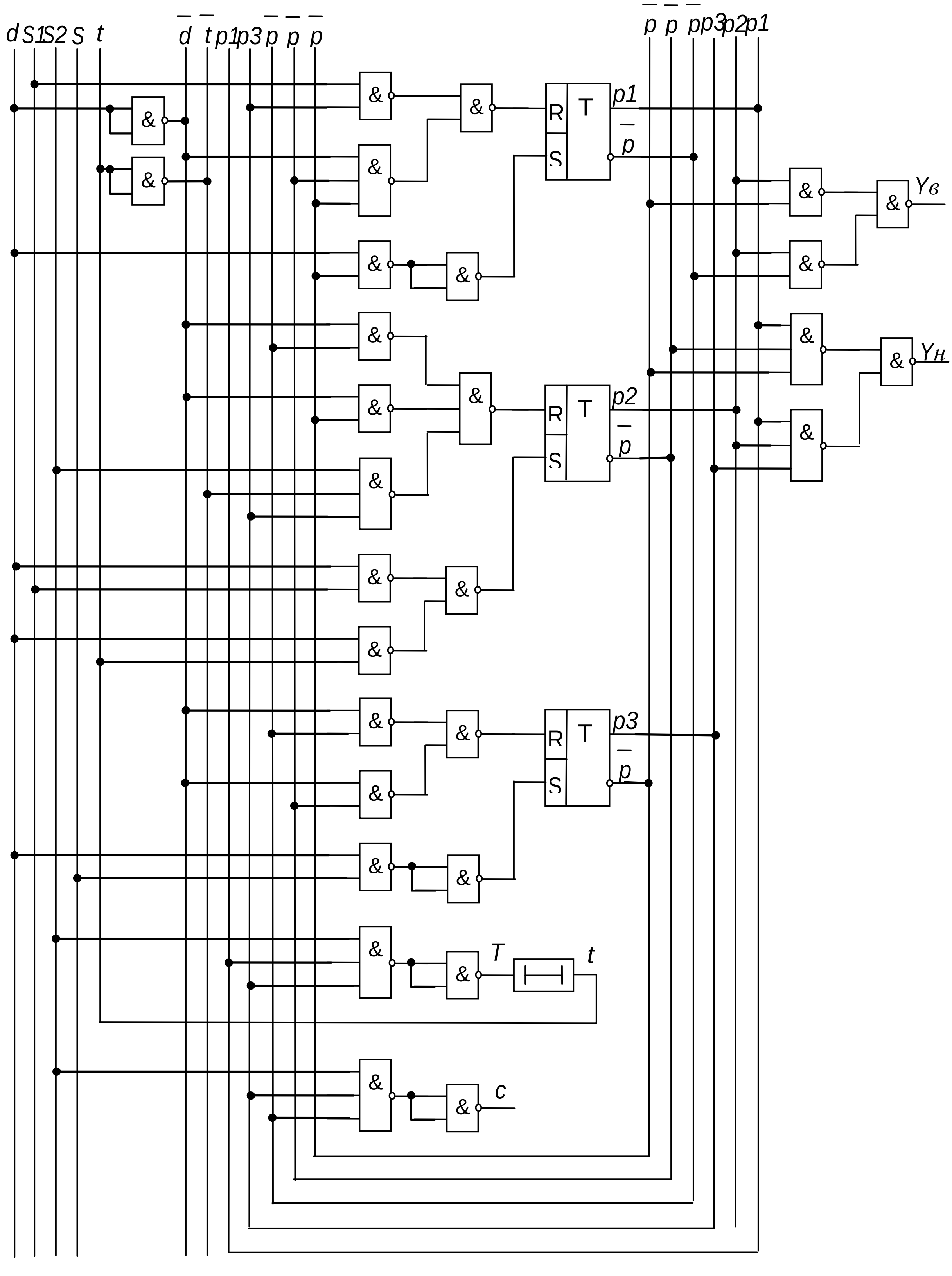


Рисунок 7 – Функциональная схема блока Б3 на элементах И-НЕ с реализацией памяти на RS-триггерах

4.5 Синтез блока, формирующего команду отработки автоматического цикла Б1

Входные и выходные сигналы блока представлены на рисунке 8.

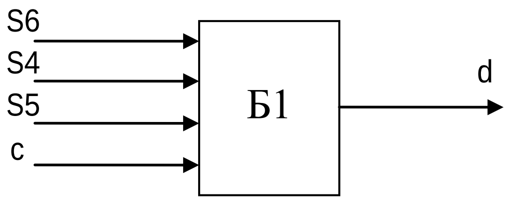


Рисунок 8 – Входные и выходные сигналы блока Б1

Для формального описания принципа работы блока составим автоматную таблицу. Так как последовательность поступления сигналов на вход блока неизвестна, следовательно, в автоматной таблице необходимо рассмотреть реакцию блока на любую допустимую входную последовательность. Количество состояний в автоматной таблице будет определяться количеством комбинаций значений выходного сигнала (2 комбинации: $d = 0$; $d = 1$). Автоматная таблица для блока Б1 будет иметь следующий вид.

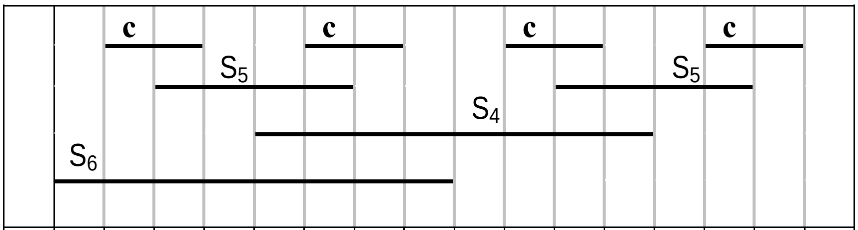
Таблица 21 – Автоматная таблица выходов и переходов блока Б1

		c				**c**				**c**				**c**			
				S_5									S_5				
									S_4								
	S_6																d
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

Данная таблица является минимизированной, так как содержит два состояния, которые несовместимы между собой по выходам. В качестве кодов состояний можно использовать значение выходной переменной d.

Для определения логического выражения для выходной переменной построим карту Карно.

Таблица 22 – Карта Карно для сигнала d



0c	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1c	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

| d

По карте Карно запишем минимизированное выражение для переменной d.

$$d = S4 \cdot S6 \cdot \overline{S5} \cdot \bar{c} + d \cdot S6 \cdot \overline{S5} \cdot \bar{c}$$

Преобразуем полученное выражение в базис операции Шеффера.

$$d = \overline{\overline{S4 \cdot S6 \cdot \overline{S5} \cdot \bar{c} + d \cdot S6 \cdot \overline{S5} \cdot \bar{c}}} = \overline{\overline{S4 \cdot S6 \cdot \overline{S5} \cdot \bar{c}} \cdot \overline{d \cdot S6 \cdot \overline{S5} \cdot \bar{c}}} =$$
$$= (S4 \,|\, S6 \,|\, \overline{S5} \,|\, \bar{c}) \,|\, (d \,|\, S6 \,|\, \overline{S5} \,|\, \bar{c})$$

Функциональная схема блока Б1 представлена на рисунке

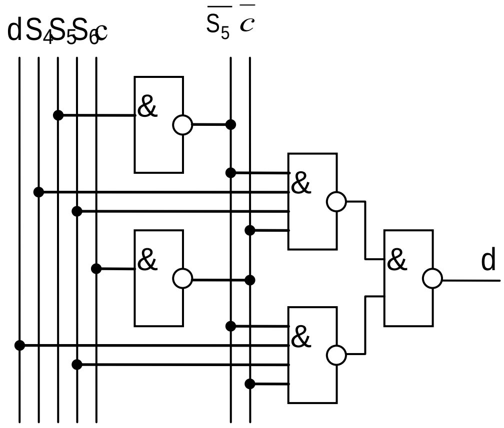


Рисунок 9 – Функциональная схема блока Б1 на элементах И-НЕ

4.6 Синтез блока режима «Наладка» Б2

Входные и выходные сигналы блока представлены на рисунке 10.

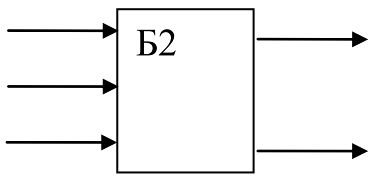


Рисунок 10 – Входные и выходные сигналы блока Б2

Автоматная таблица для данного блока составляется аналогично предыдущему блоку. Таблица будет содержать три внутренних состояния по числу допустимых комбинаций значений выходных сигналов ($Z_в$,$Z_н$ = 00; 01; 10). Комбинация $Z_в$,$Z_н$ = 11 является запрещённой. В автоматной таблице необходимо предусмотреть блокировку от одновременного нажатия кнопок *S7*, *S8*. Если *S7* = 1 и *S8* = 1, то на выходе должен формироваться сигнал $Z_в$ = 0, $Z_н$ = 0 для исключения короткого замыкания.

Автоматная таблица блока Б2 будет иметь следующий вид.

Таблица 23 – Автоматная таблица выходов и переходов блока Б2

	S_6		S_8		S_7	S_8			$Z_в$	$Z_н$
0	0	0	0	0	1	0	2	0	0	0
1	0	0	0	0	1	0	0	0	1	0
2	0	0	0	0	0	0	2	0	0	1

Данная таблица является минимизированной, так как содержит три состояния, которые несовместимы между собой по выходам. В качестве кодов состояний можно использовать значения выходных сигналов $Z_в$,$Z_н$.

Для определения выходных функций строим карты Карно для переменных $Z_в$,$Z_н$.

Таблица 24 – Карта Карно для переменной $Z_в$

	S_6	S_8		S_7		S_8			
0с	0	0	0	0	1	0	0	0	
2с	0	0	0	0	0	0	0	0	$Z_н$
	0	0	0	0	0	0	0	0	
1с	0	0	0	0	1	0	0	0	$Z_в$

$$Z_в = S7 \cdot \overline{S6} \cdot \overline{S8} \cdot \overline{Z_н}$$

Таблица 25 – Карта Карно для переменной $Z_н$

		S_8					S_8		
					S_7				
	S_6								
0с	0	0	0	0	0	0	1	0	
2с	0	0	0	0	0	0	1	0	$Z_Н$
	0	0	0	0	0	0	0	0	
1с	0	0	0	0	0	0	0	0	$Z_В$

$$Z_н = \overline{S6} \cdot \overline{S7} \cdot S8 \cdot \overline{Z_в}$$

Преобразуем полученные выражения в базис Шеффера.

$$Z_в = \overline{\overline{S7 \cdot \overline{S6} \cdot \overline{S8} \cdot \overline{Z_н}}} = \overline{S7 | \overline{S6} | \overline{S8} | \overline{Z_н}} = (S7 | \overline{S6} | \overline{S8} | \overline{Z_н}) | (S7 | \overline{S6} | \overline{S8} | \overline{Z_н})$$

$$Z_н = \overline{\overline{\overline{S6} \cdot \overline{S7} \cdot S8 \cdot \overline{Z_в}}} = \overline{\overline{S6} | \overline{S7} | S8 | \overline{Z_в}} = (\overline{S6} | \overline{S7} | S8 | \overline{Z_в}) | (\overline{S6} | \overline{S7} | S8 | \overline{Z_в})$$

На основании полученных выражений можем построить функциональную схему блока Б2.

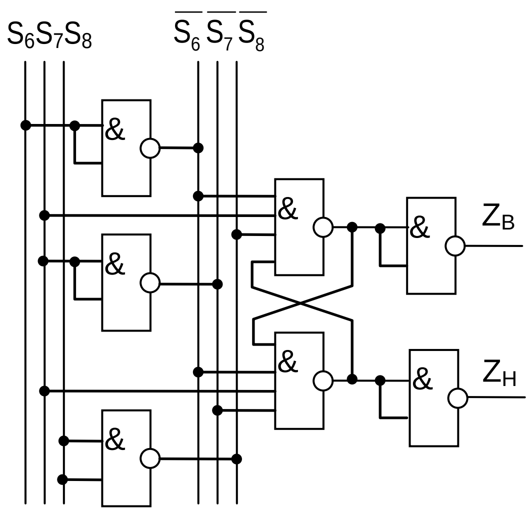


Рисунок 11 – Функциональная схема блока Б2 на элементах И-НЕ

4.7 Синтез выходного блока Б4

Входные и выходные сигналы блока представлены на рисунке 12.

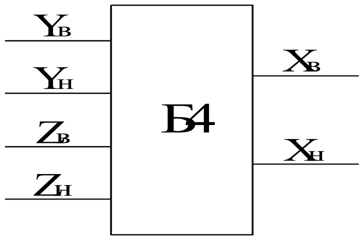


Рисунок 12 – Входные и выходные сигналы блока Б4

Автоматная таблица выходов и переходов для данного блока будет иметь следующий вид.

Таблица 26 – Автоматная таблица выходов и переходов блока Б4

																	Xв	Xн
		Yн				Yн				Yн				Yн				
			Zн								Zн							
					Yв													
	Zв																	
0	1	~	~	0	~	~	~	~	1	0	~	~	2	~	2	0	0	0
1	1	~	~	0	~	~	~	~	1	0	~	~	0	~	0	0	1	0
2	0	~	~	0	~	~	~	~	0	0	~	~	2	~	2	0	0	1

На основании автоматной таблицы составим карты Карно и запишем выражения для выходных переменных $X_в$, $X_н$.

Таблица 27 – Карта Карно для переменной $X_в$

		Yн				Yн				Yн				Yн			
			Zн								Zн						
					Yв												
	Zв																
0	1	~	~	0	~	~	~	~	1	0	~	~	0	~	0	0	
2	0	~	~	0	~	~	~	~	0	0	~	~	0	~	0	0	Xн
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	1	~	~	0	~	~	~	~	1	0	~	~	0	~	0	0	Xв

$$X_B = \overline{X_H} \cdot Z_B \cdot \overline{Z_H} + \overline{X_H} \cdot Y_B \cdot \overline{Y_H}$$

Таблица 28 – Карта Карно для переменной $X_н$

		Yн			Yн					Yн				Yн			
				Zн							Zн						
							Yв										
	Zв																
0	0	~	~	0	~	~	~	~	0	0	~	~	1	~	1	0	
2	0	~	~	0	~	~	~	~	0	0	~	~	1	~	1	0	Xн
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	~	~	0	~	~	~	~	0	0	~	~	0	~	0	0	Xв

$$X_H = \overline{X_B} \cdot Z_H \cdot \overline{Z_B} + \overline{X_B} \cdot Y_H \cdot \overline{Y_B}$$

Преобразуем полученные выражения в базис Шеффера.

$$X_B = \overline{\overline{\overline{X_H} \cdot Z_B \cdot \overline{Z_H} + \overline{X_H} \cdot Y_B \cdot \overline{Y_H}}} = (\overline{X_H} \,|\, Z_B \,|\, \overline{Z_H}) \,|\, (\overline{X_H} \,|\, Y_B \,|\, \overline{Y_H})$$

$$X_H = \overline{\overline{\overline{X_B} \cdot Z_H \cdot \overline{Z_B} + \overline{X_B} \cdot Y_H \cdot \overline{Y_B}}} = (\overline{X_B} \,|\, Z_H \,|\, \overline{Z_B}) \,|\, (\overline{X_B} \,|\, Y_H \,|\, \overline{Y_B})$$

На основании полученных выражений можем построить функциональную схему блока Б4.

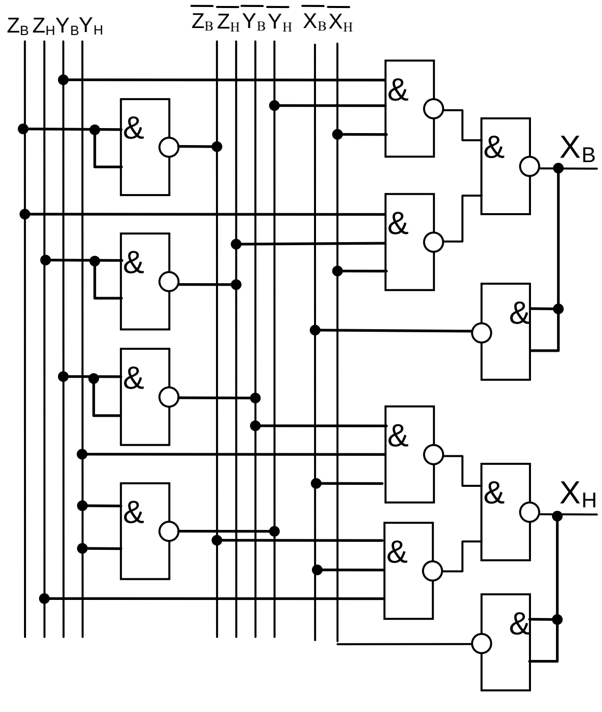


Рисунок 13 – Функциональная схема блока Б4 на элементах И-НЕ

5 ВЫБОР ЭЛЕМЕНТОВ. ПОСТРОЕНИЕ ПРИНЦИПИАЛЬНОЙ СХЕМЫ УСТРОЙСТВА УПРАВЛЕНИЯ

Принципиальная схема устройства управления может быть реализована на любых логических элементах. В нашем случае схема должна быть реализована на элементах И-НЕ.

Логическую операцию И-НЕ реализуют интегральные микросхемы типа ЛА [2]. Каждый из корпусов интегральных схем (ИС) типа ЛА содержит от двух до четырёх логических элементов, а микросхемы ЛА2 и ЛА19 содержат по одному логическому элементу И-НЕ на восемь и на двенадцать входов соответственно.

Условные обозначения и цоколёвка микросхем типа ЛА представлены на рисунке 14

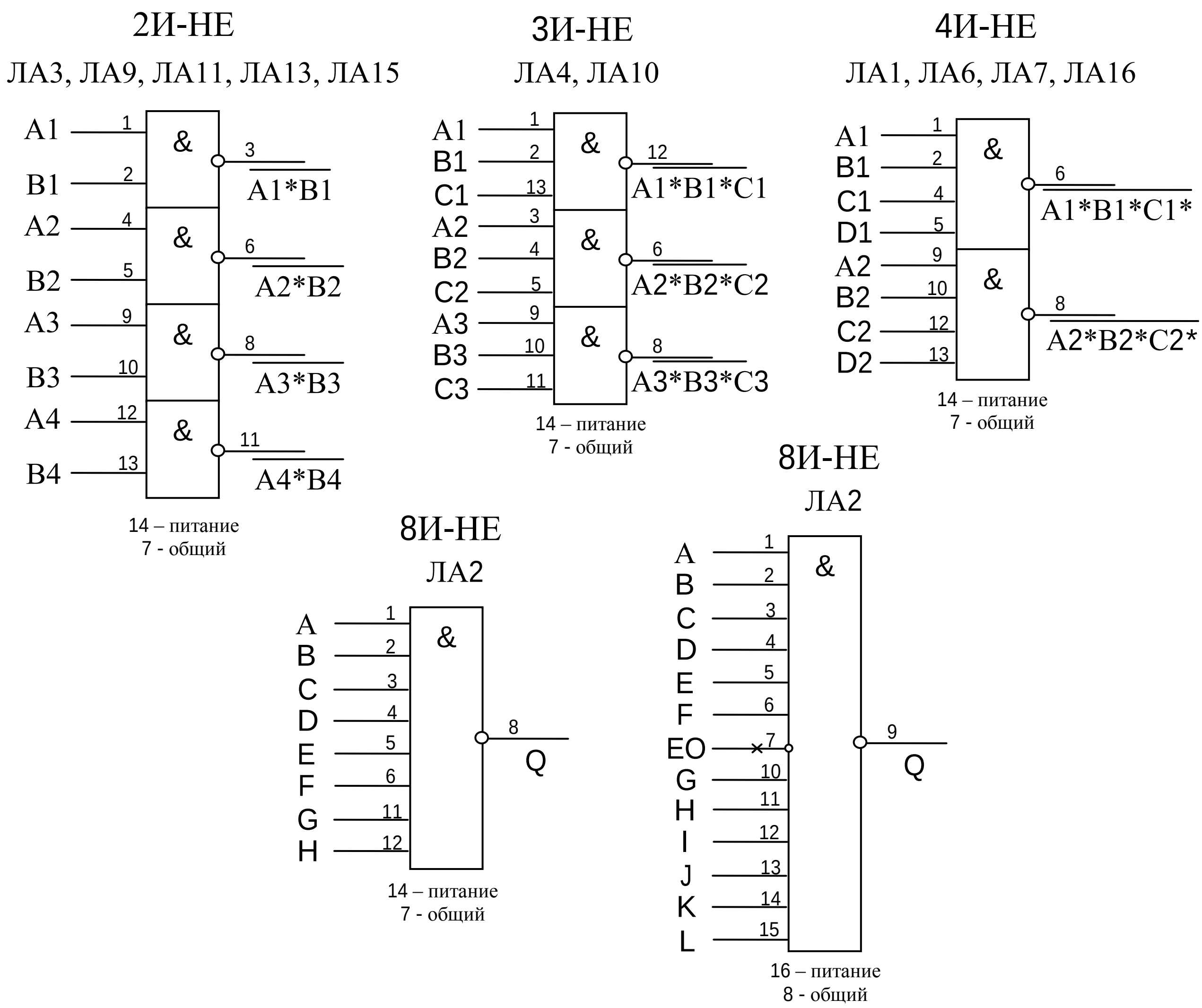


Рисунок 14 - Условные обозначения и цоколёвка микросхем типа ЛА

EO – дополнительный вход, дающий разрешение по выходу (EO = 1 обеспечивает 3 состояние Z с очень большим выходным сопротивлением).

Микросхемы ЛА7…ЛА11, ЛА13, ЛА18 имеют выходы с открытым коллектором. Для формирования выходного перепада напряжения к выходу такого элемента необходимо подключить нагрузку. В качестве нагрузки могут выступать сегменты индикаторов, лампы накаливания, светодиоды, обмотки реле.

Для построения бесконтактных схем используем микросхемы 155 серии. Так же будет использован асинхронный RS-триггер. Выходные сигналы $X_в$, $X_н$ будем формировать с помощью микросхем с открытым коллектором для подключения обмоток промежуточных реле, которые будут включать силовые контакторы.

Для первой схемы (без использования RS-триггеров) будем использовать следующие микросхемы:

К155ЛА3 – DD1 – DD9;

К155ЛА9 – DD10;

К155ЛА4 – DD11 - DD15;

К155ЛА1 – DD16 - DD18.

Для второй схемы (с использования RS-триггеров) будем использовать следующие микросхемы

К155ЛА3 – DD1 – DD9;

К155ЛА9 – DD10;

К155ЛА4 – DD11 - DD14;

К155ЛА1 – DD15, DD16

К155ТР2 - DD17

Принципиальные схемы устройства управления представлены в приложениях 1, 2.

6 ПРОВЕРКА ПРАВИЛЬНОСТИ РАБОТЫ УСТРОЙСТВА УПРАВЛЕНИЯ

Для проверки правильности работы устройства управления необходимо построить проверочную таблицу. В данной таблице отмечают реакцию управляющего устройства на входную последовательность сигналов. Для этого по значениям входных и внутренних сигналов в данный момент времени по полученным выражениям вычисляют значения выходных сигналов и внутренних сигналов в следующий момент времени. По этим сигналам проверяется, выполняется ли заданный цикл перемещений.

Проверочная таблица строится для блока Б3, так как данный блок имеет наиболее сложный алгоритм работы.

При заполнении таблицы необходимо придерживаться следующей последовательности. Задаёмся первым входным набором (нахождение каретки в исходном положении) и начальным значением внутренних сигналов (код исходного внутреннего состояния). На основании значений данных сигналов вычисляем значения выходных сигналов и последующих внутренних сигналов. Если значения последующих внутренних сигналов отличаются от значений предыдущих внутренних сигналов (переход из состояния в состояние), то на следующем шаге для вычислений берём тот же входной набор и полученные на предыдущем шаге значения внутренних сигналов. Если значения последующих внутренних сигналов не отличаются от значений предыдущих внутренних сигналов (автомат находится в устойчивом состоянии), то на следующем шаге для вычислений берём следующий входной набор и полученные на предыдущем шаге значения внутренних сигналов. Таким образом, в проверочной таблице должны отражаться все переходы, отмеченные в исправленной закодированной минимизированной автоматной таблице.

Для построения проверочной таблицы может использоваться редактор таблиц Excel.

Для вычисления выходных сигналов RS-триггера может использоваться следующее выражение

$$X(t+1) = S + \overline{R} \cdot X(t),$$

где *S*, *R* – значения входных сигналов триггера в данный момент времени;

X(t) – значение выходного сигнала в данный момент времени;

X(t +1) – значение выходного сигнала в последующий момент времени.

Проверочная таблица для первого варианта реализации будет иметь следующий вид.

Таблица 29 – Проверочная таблица устройства управления для первого варианта реализации

	Входные сигналы					Предыдущие внутренние			Последующие внутренние			Выходные сигналы			
	d	S1	S2	S3	t	p1	p2	p3	p1	p2	p3	Yв	Yн	T	c
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0	1	0	0	0	1	0	0
3	1	0	1	0	0	1	0	0	1	0	0	0	1	0	0
4	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0
5	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0
6	1	1	0	0	0	1	1	0	1	1	0	1	0	0	0
7	1	0	0	0	0	1	1	0	1	1	0	1	0	0	0
8	1	0	1	0	0	1	1	0	1	1	0	1	0	0	0
9	1	0	0	0	0	1	1	0	1	1	0	1	0	0	0
10	1	0	0	1	0	1	1	0	1	1	1	0	1	0	0
11	1	0	0	1	0	1	1	1	1	1	1	0	1	0	0
12	1	0	0	0	0	1	1	1	1	1	1	0	1	0	0
13	1	0	1	0	0	1	1	1	1	0	1	0	0	1	0
14	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0
15	1	0	1	0	1	1	0	1	1	1	1	0	1	1	0
16	1	0	1	0	1	1	1	1	1	1	1	0	1	1	0
17	1	0	0	0	1	1	1	1	1	1	1	0	1	0	0
18	1	0	0	0	0	1	1	1	1	1	1	0	1	0	0
19	1	1	0	0	0	1	1	1	0	1	1	1	0	0	0
20	1	1	0	0	0	0	1	1	0	1	1	1	0	0	0
21	1	0	0	0	0	0	1	1	0	1	1	1	0	0	0
22	1	0	1	0	0	0	1	1	0	0	1	0	0	0	1
23	1	0	1	0	0	0	0	1	0	0	1	0	0	0	1
24	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
25	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Таблица 30 – Проверочная таблица устройства управления для второго варианта реализации

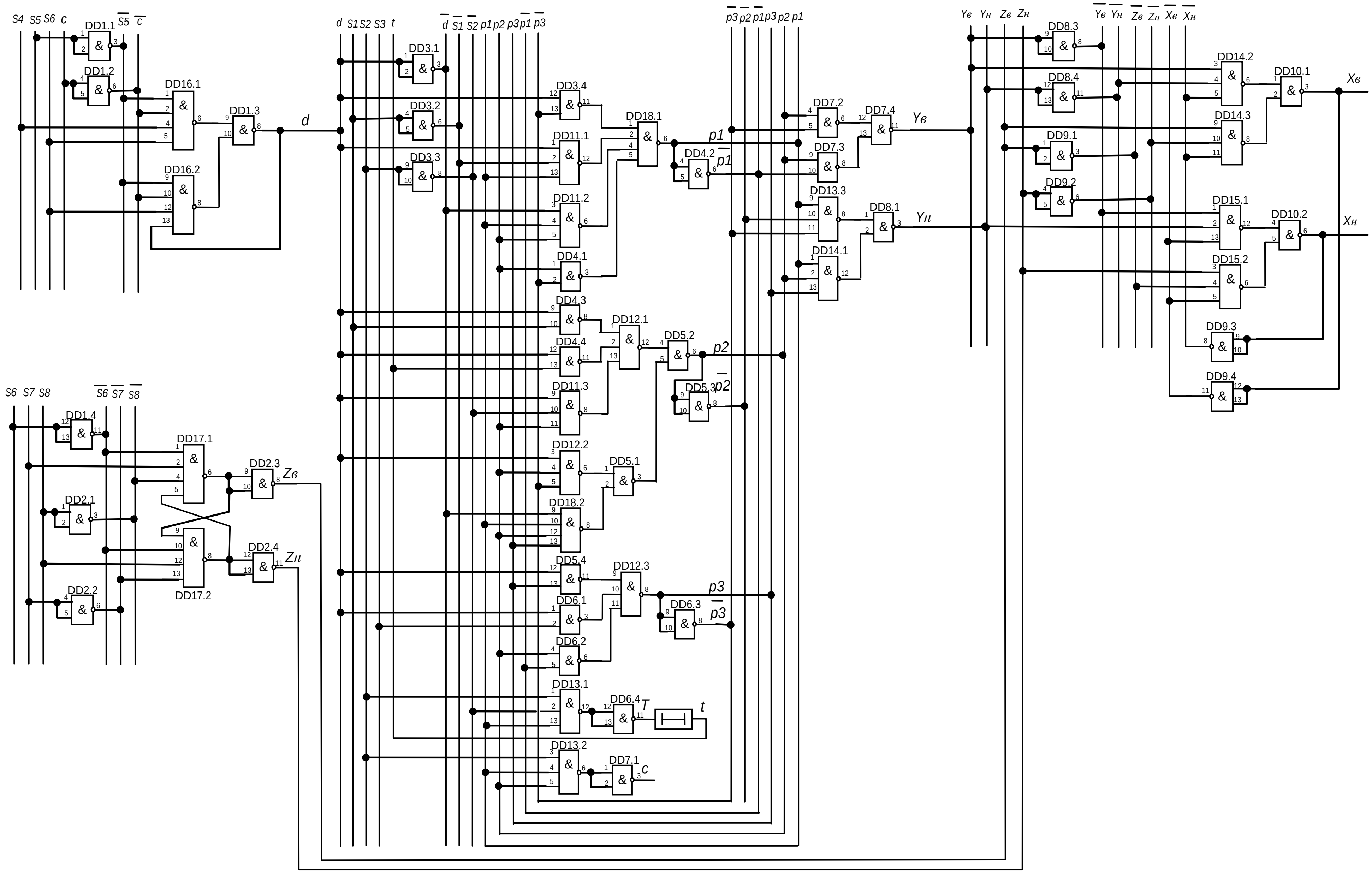
	Входные сигналы					Предыду-щие внут-ренние			Входные сигналы триггеров						Последую-щие внут-ренние			Выходные сигналы			
	d	S1	S2	S3	t	p1	p2	p3	R1	S1	R2	S2	R3	S3	p1	p2	p3	Yв	Yн	T	c
1	0	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0
3	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0
4	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0
5	1	1	0	0	0	1	0	0	0	1	0	1	0	0	1	1	0	1	0	0	0
6	1	1	0	0	0	1	1	0	0	1	0	1	0	0	1	1	0	1	0	0	0
7	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1	1	0	1	0	0	0
8	1	0	1	0	0	1	1	0	0	1	0	0	0	0	1	1	0	1	0	0	0
9	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1	1	0	1	0	0	0
10	1	0	0	1	0	1	1	0	0	1	0	0	0	1	1	1	1	0	1	0	0
11	1	0	0	1	0	1	1	1	0	0	0	0	0	1	1	1	1	0	1	0	0
12	1	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1	1	0	1	0	0
13	1	0	1	0	0	1	1	1	0	0	1	0	0	0	1	0	1	0	0	1	0
14	1	0	1	0	0	1	0	1	0	0	1	0	0	0	1	0	1	0	0	1	0
15	1	0	1	0	1	1	0	1	0	0	0	1	0	0	1	1	1	0	1	1	0
16	1	0	1	0	1	1	1	1	0	0	0	1	0	0	1	1	1	0	1	1	0
17	1	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	0	1	0	0
18	1	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1	1	0	1	0	0
19	1	1	0	0	0	1	1	1	1	0	0	1	0	0	0	1	1	1	0	0	0
20	1	1	0	0	0	0	1	1	1	0	0	1	0	0	0	1	1	1	0	0	0
21	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0
22	1	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0	0	1
23	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	1
24	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0
25	0	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Из проверочных таблиц видно, что при поступлении входных последовательностей в соответствии с циклом перемещений на выходе устройства управления формируются заданные значения выходных сигналов. Следовательно, проектирование схемы управления выполнено верно, и устройство управления работает по заданному алгоритму.

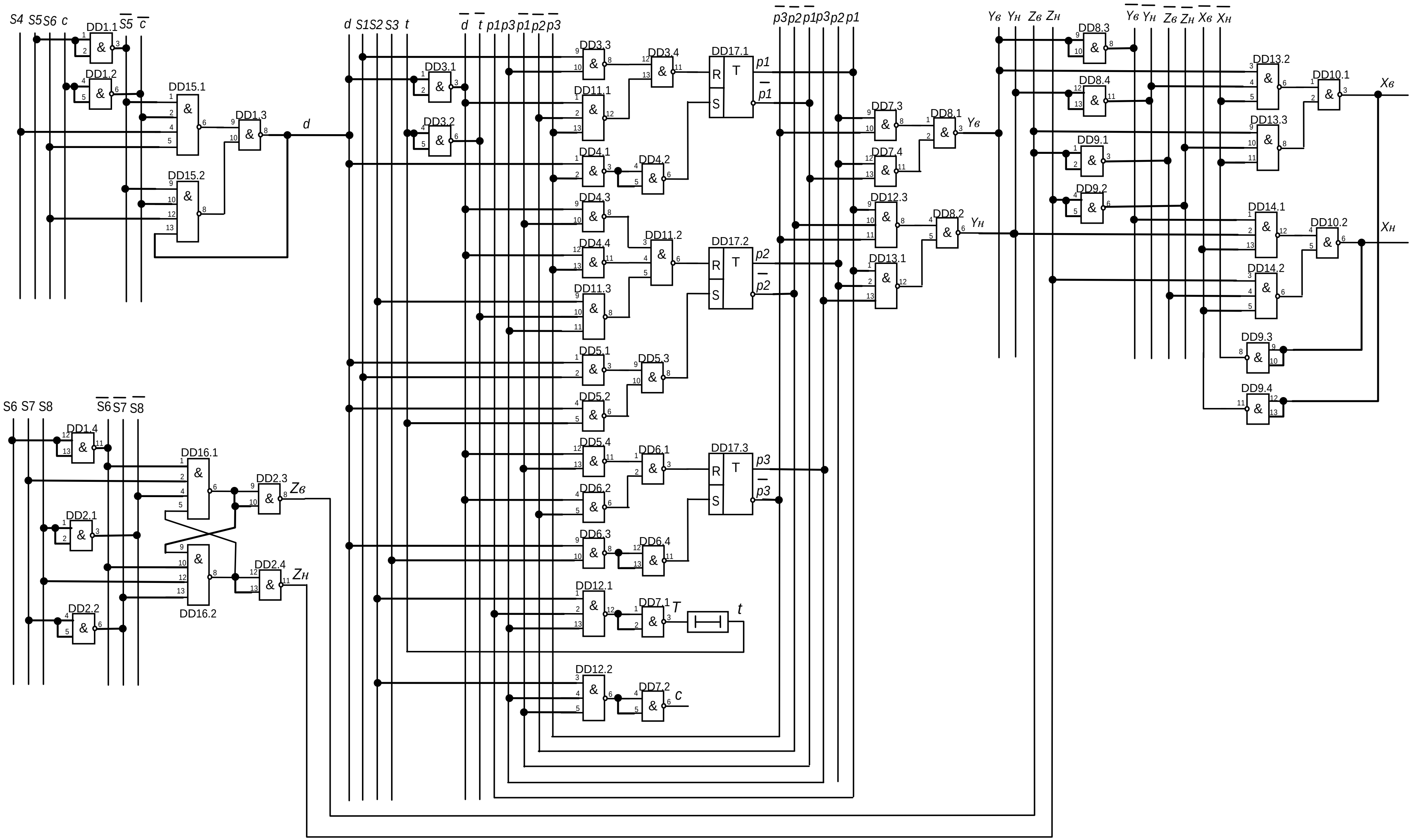
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ершова Э. Б., Рогинский В. Н., Маркин Н. П. Основы дискретной автоматики в электросвязи. – М.: Связь, 1980.
2. Цифровые интегральные микросхемы: Справочник/ М. И. Богданович, И. Н. Грель, С. А. Дубина и др. – 2-е изд., перераб. и доп. – Мн.: Беларусь, Полымя. 1996.

ПРИЛОЖЕНИЯ



Принципиальная схема устройства управления при реализации только на бесконтактных элементах И-НЕ.



Принципиальная схема устройства управления при реализации на бесконтактных элементах И-НЕ и RS-триггерах.

3. РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ

Раздел контроля знаний УМК представлен типовыми заданиями к зачёту.

Типовые задания к зачёту

Зачёт проводится в письменной форме. При сдаче зачёта необходимо выполнить синтез управляющего устройства, алгоритм работы которого задан линейной диаграммой работы. Устройство имеет два входных сигнала (*a*, *b*) и один выходной сигнал (*x*). Сигналы на выходах могут появляться сразу или через некоторый промежуток времени (дискретное устройство с временной зависимостью). Примеры линейных диаграмм для устройств обоих типов представлены на рисунках 1 – 2.

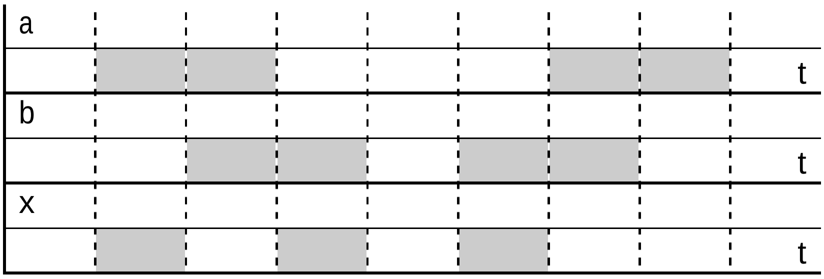


Рисунок 1 – Линейная диаграмма работы устройства без временной зависимости

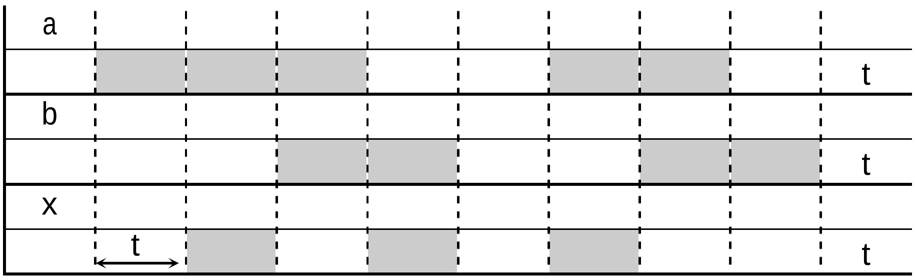
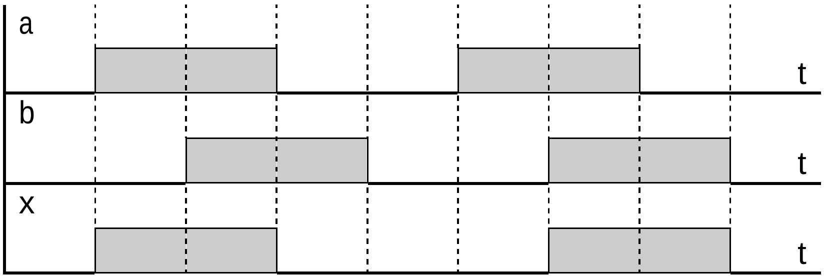


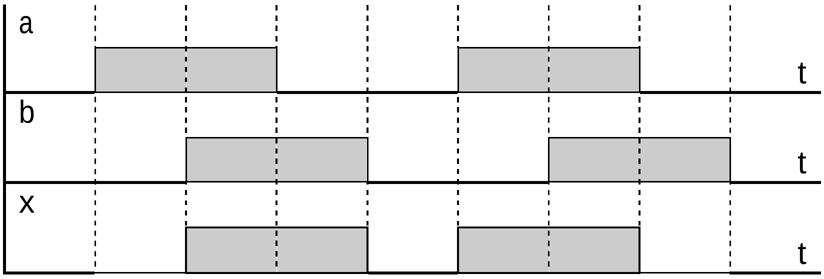
Рисунок 2 – Линейная диаграмма работы устройства с временной зависимостью (выдержкой времени)

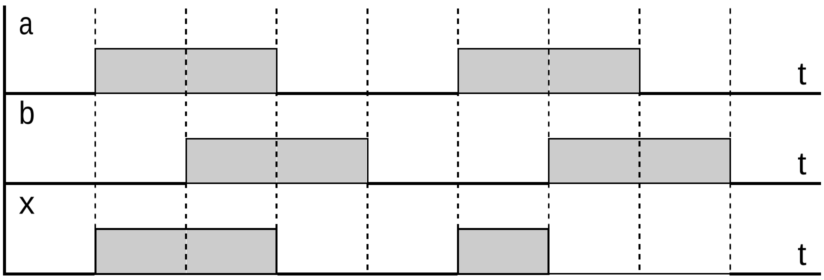
Сигналы на входах и выходах могут формироваться в разной последовательности, что приводит к большому количеству вариантов заданий.

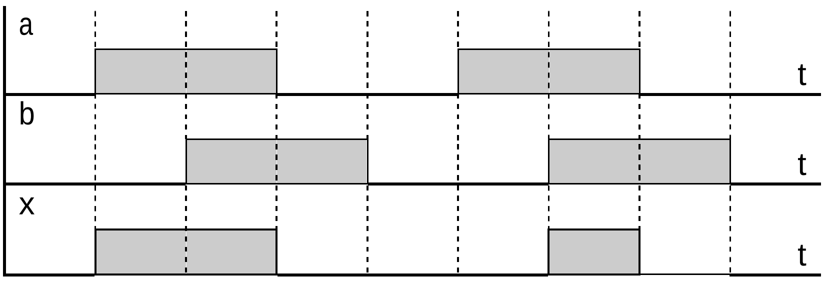
Примеры заданий с временными зависимостями и без представлены ниже.

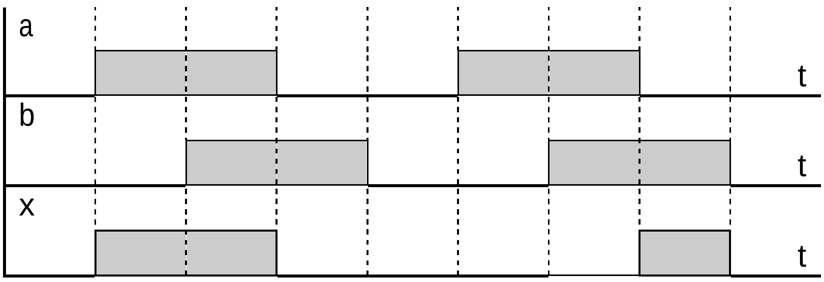
Типовые задания без временных зависимостей.

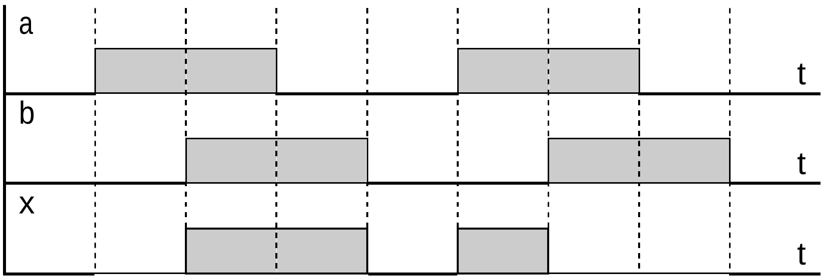


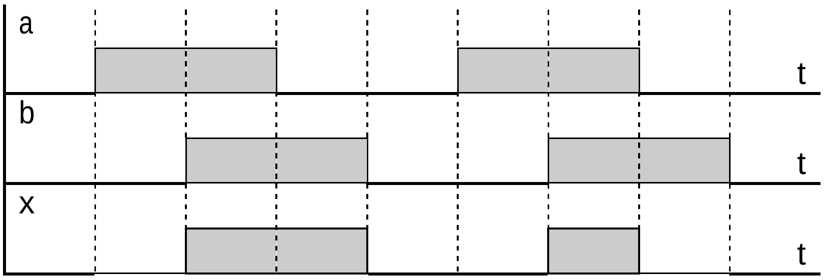


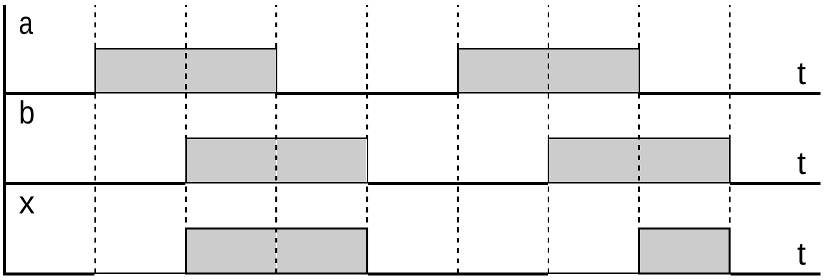


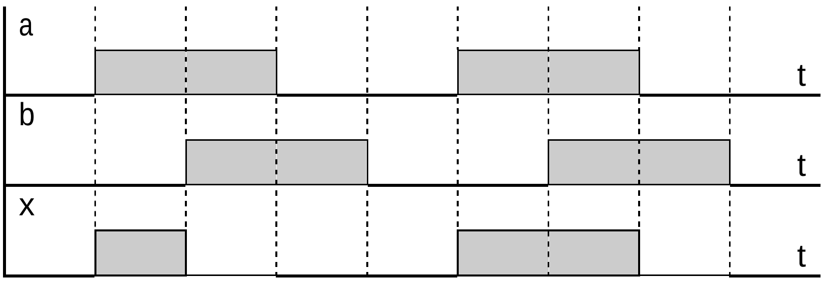


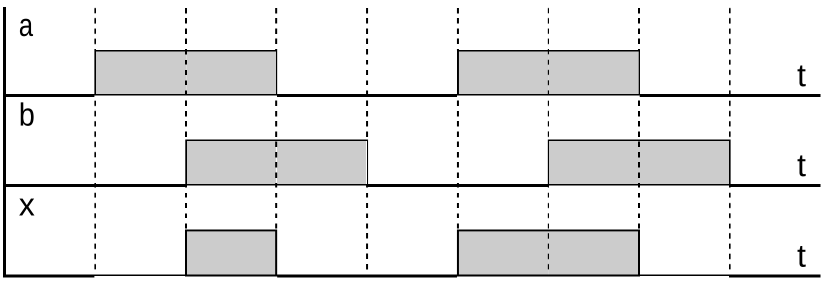


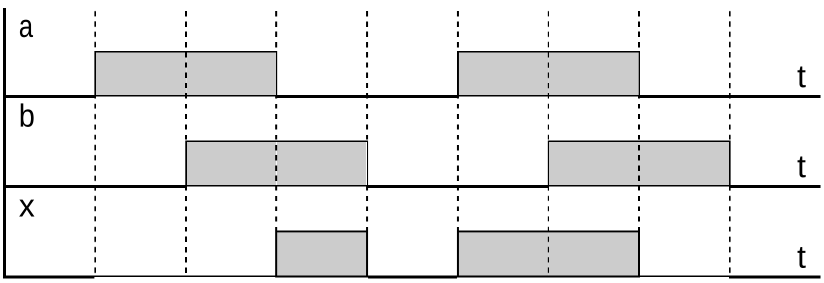

a
t
b
t
x
t

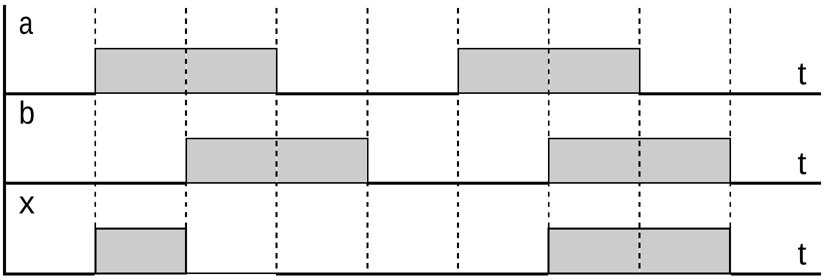

a
t
b
t
x
t

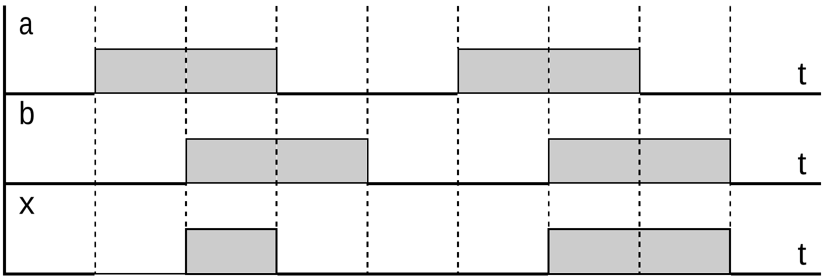

a
t
b
t
x
t

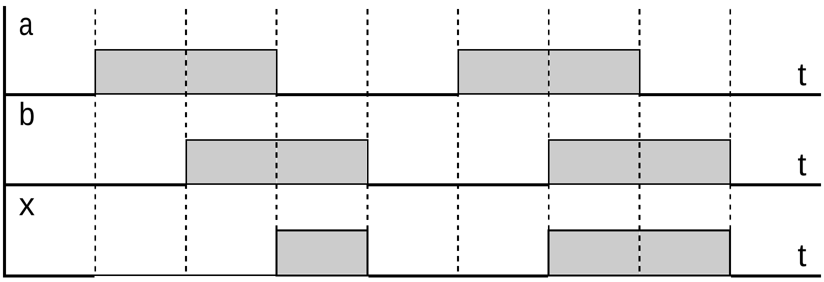

a
t
b
t
x
t

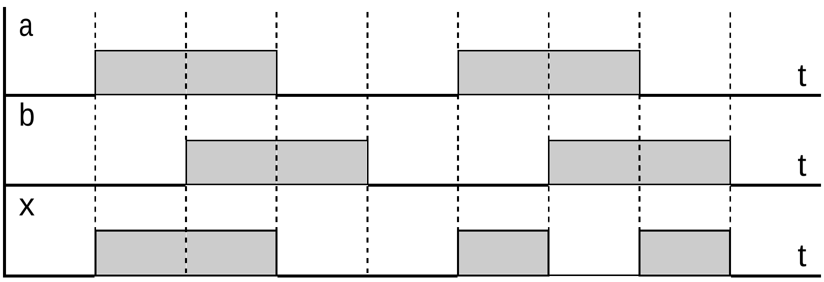

a
t
b
t
x
t

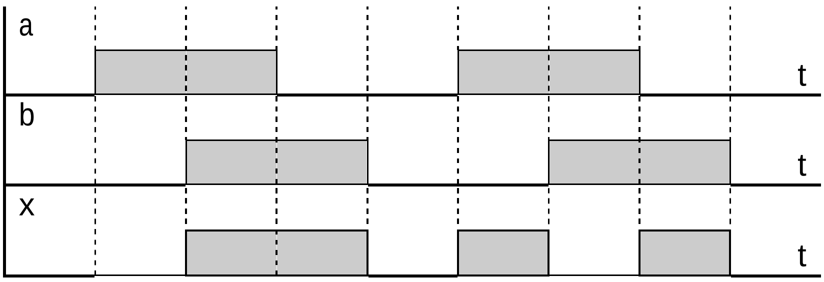

a
t
b
t
x
t

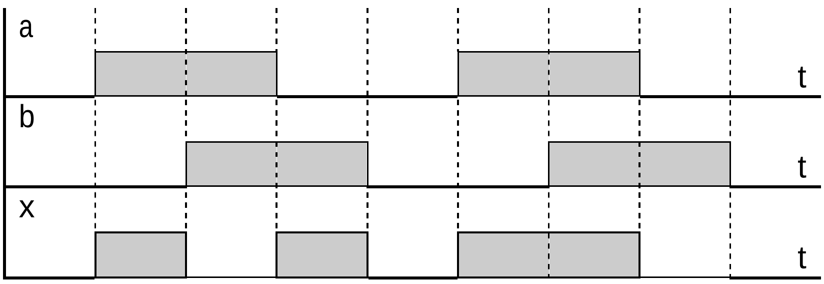

a
t
b
t
x
t

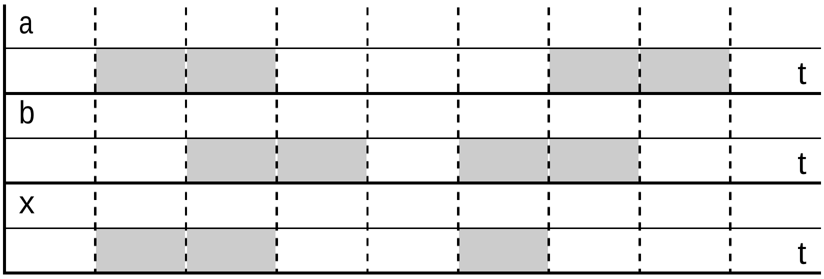

a
t
b
t
x
t

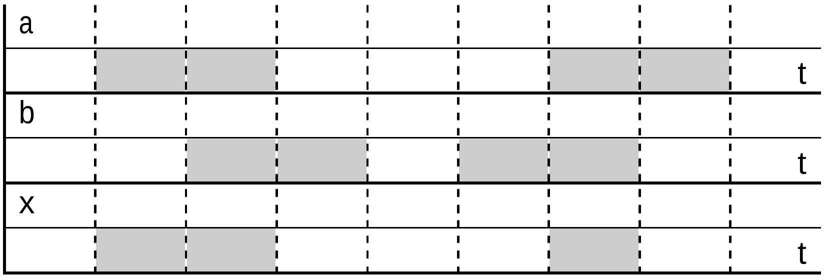

a
t
b
t
x
t

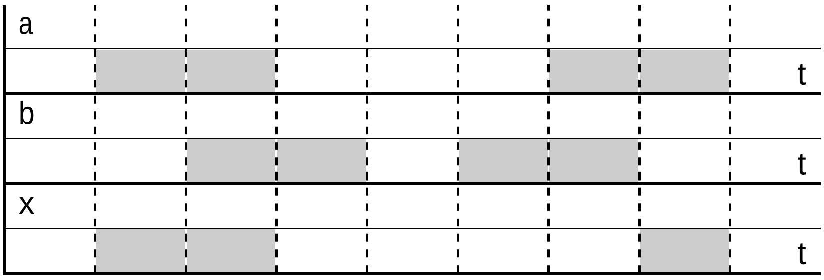

a
t
b
t
x
t

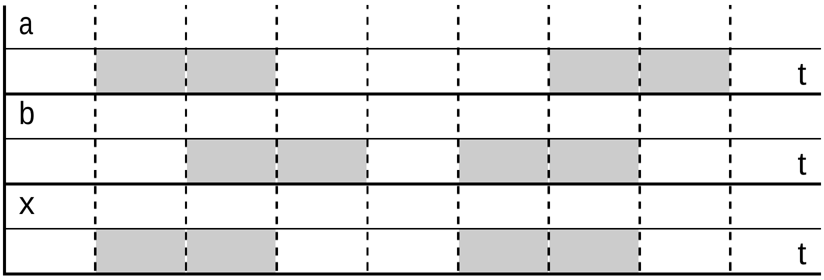

a
t
b
t
x
t

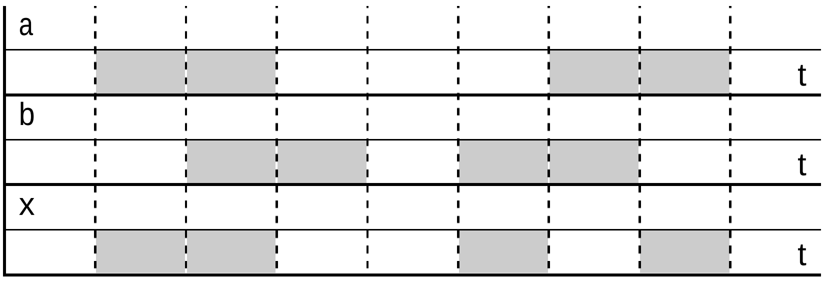

a
t
b
t
x
t

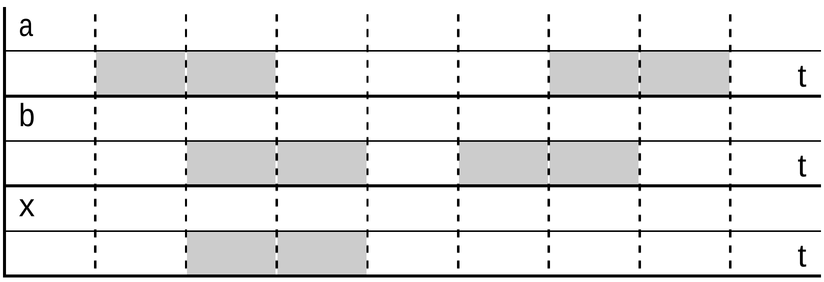

a
t
b
t
x
t

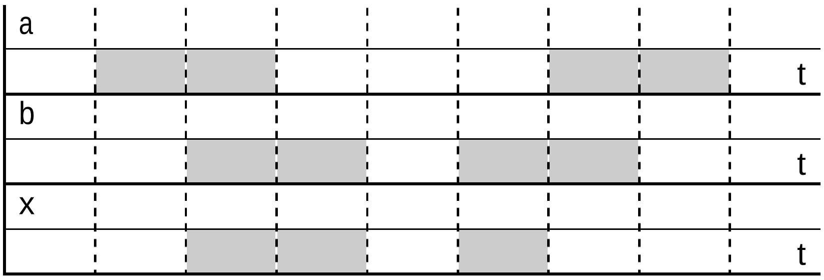

a
t
b
t
x
t

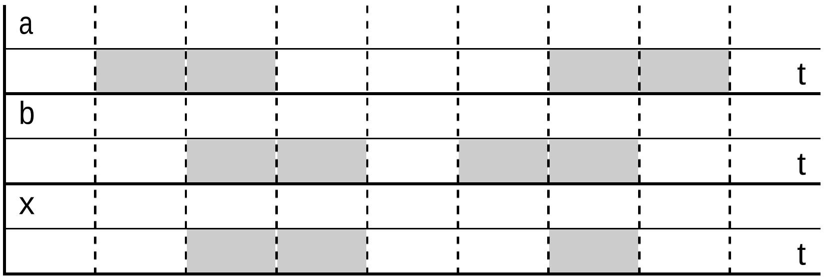

a
t
b
t
x
t

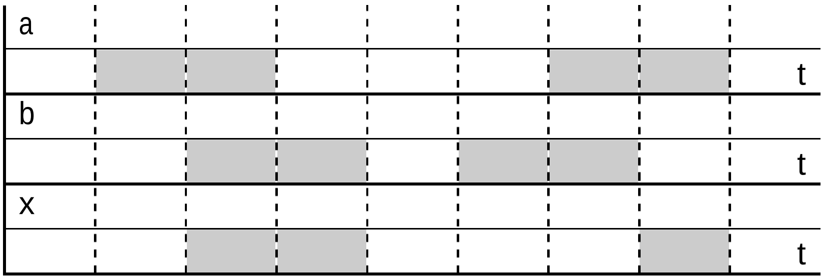

a
t
b
t
x
t

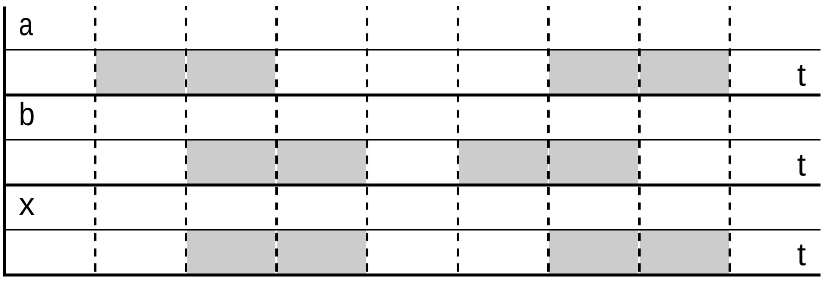

a
t
b
t
x
t

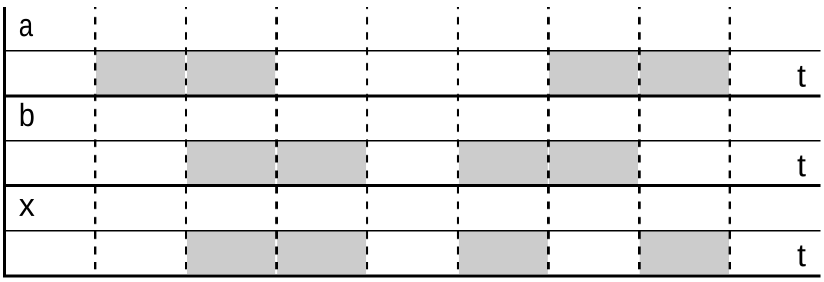

a
t
b
t
x
t

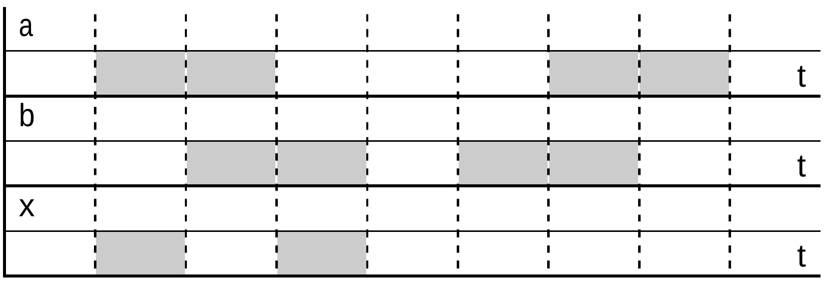

a
t
b
t
x
t


a
t
b
t
x
t

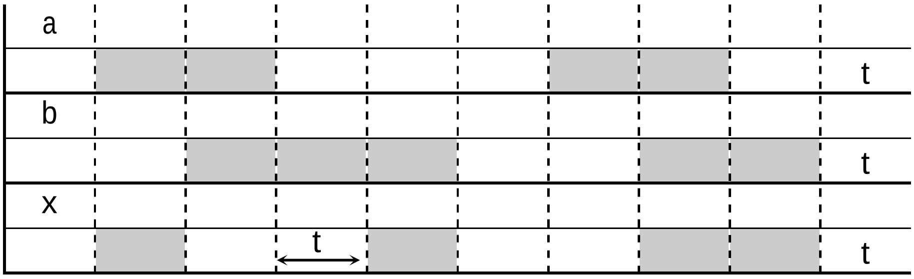

a
t
b
t
x
t

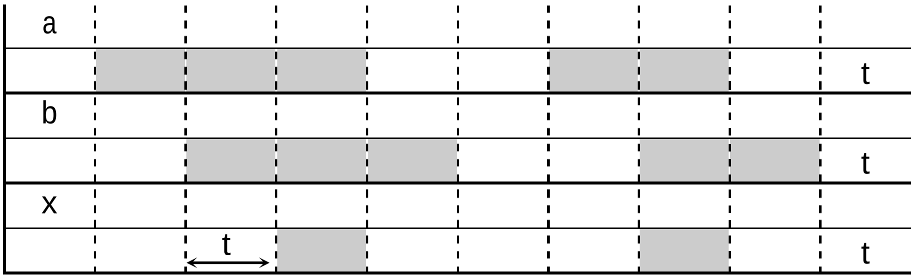

a
t
b
t
x
t

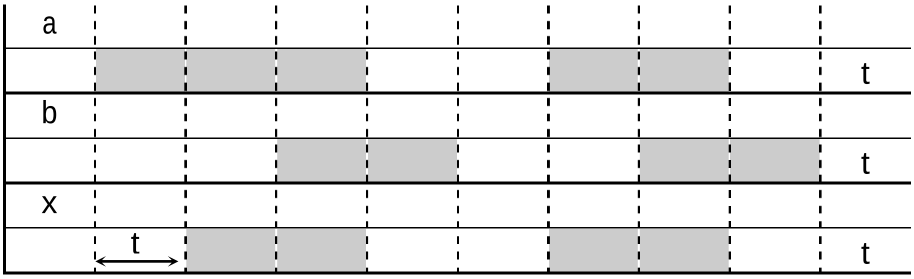

a
t
b
t
x
t

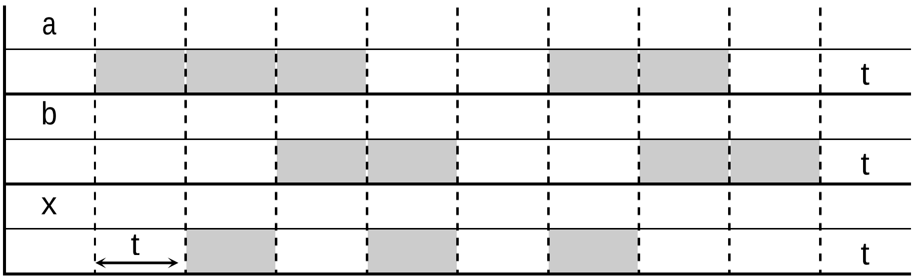

a
t
b
t
x
t

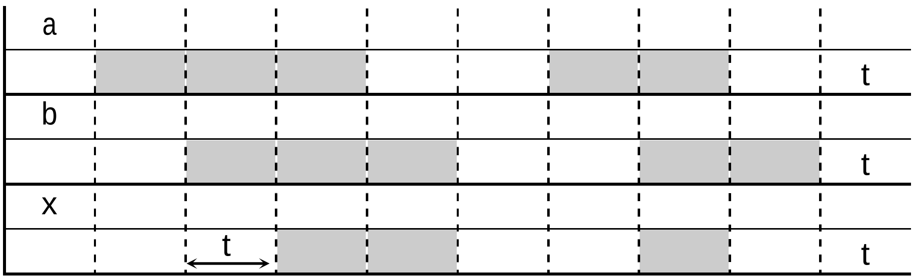
Типовые задания с временными зависимостями.

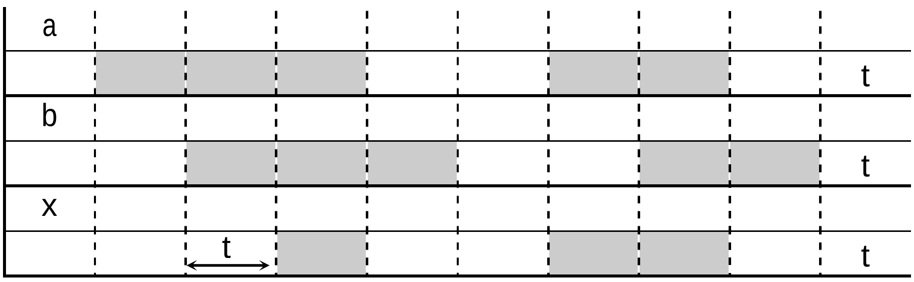


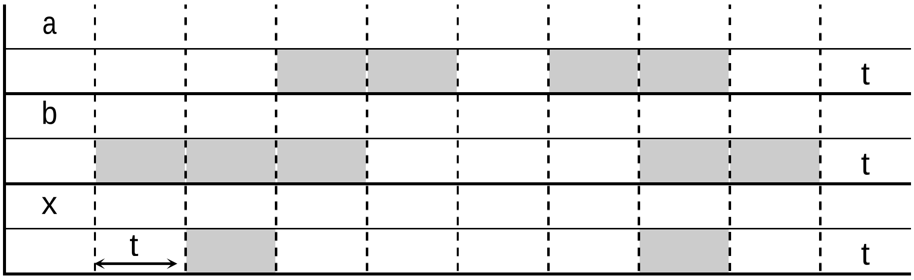


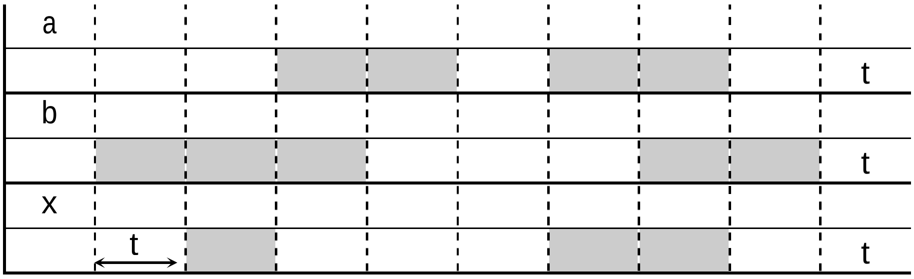


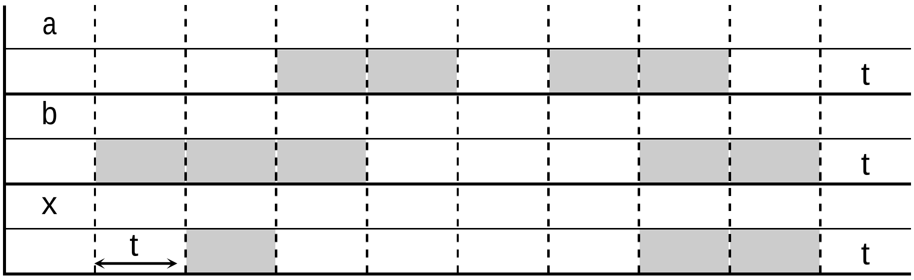


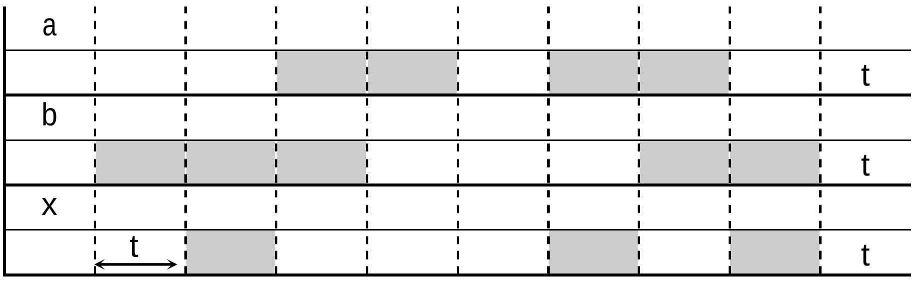


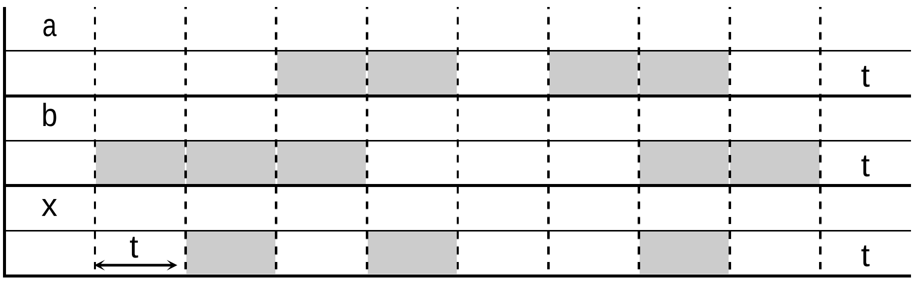


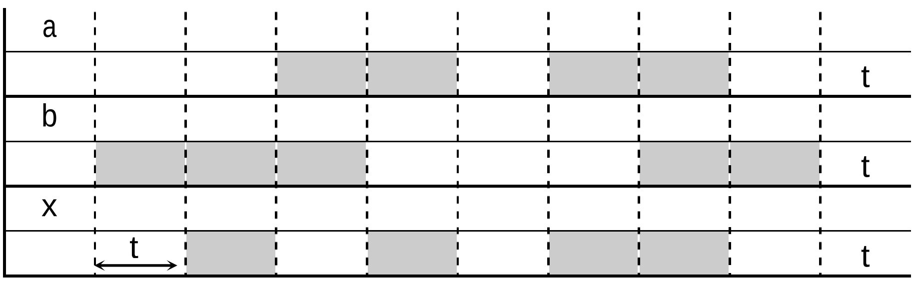

a
t
b
t
x
t
t

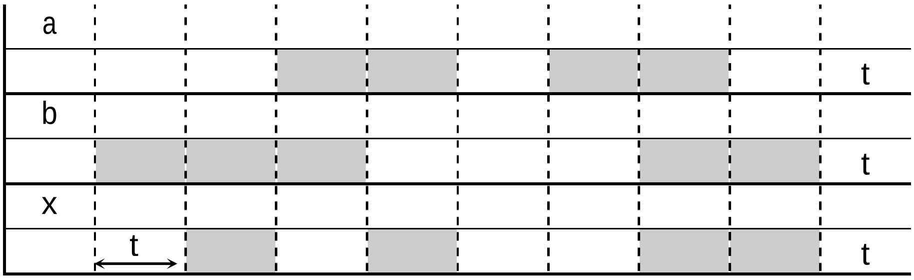

a
t
b
t
x
t
t

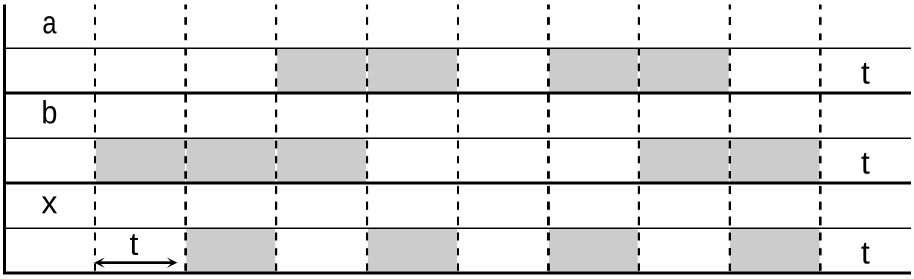

a
t
b
t
x
t
t

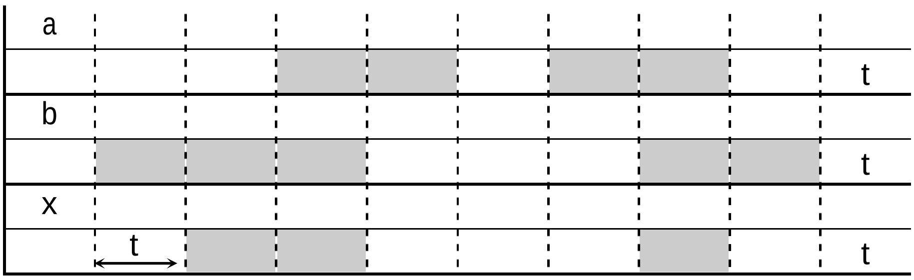

a
t
b
t
x
t
t

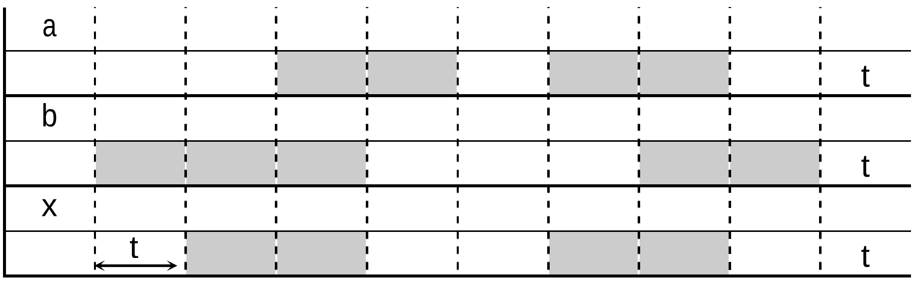

a
t
b
t
x
t
t

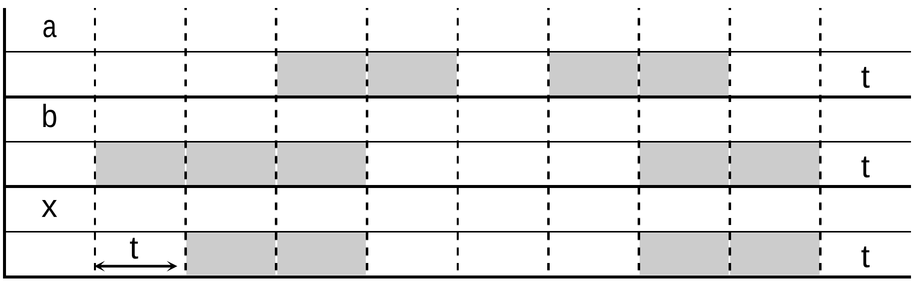

a
t
b
t
x
t
t

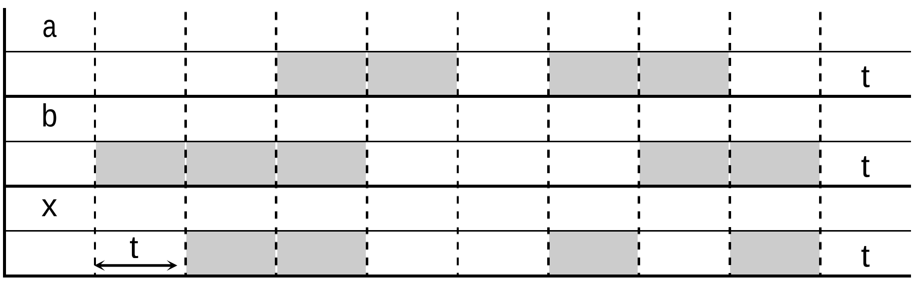

a
t
b
t
x
t
t

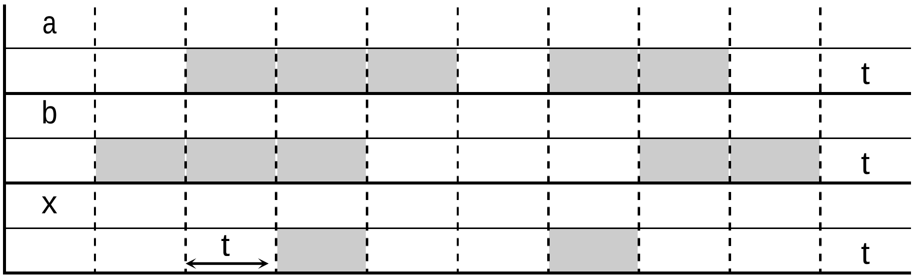

a
t
b
t
x
t
t

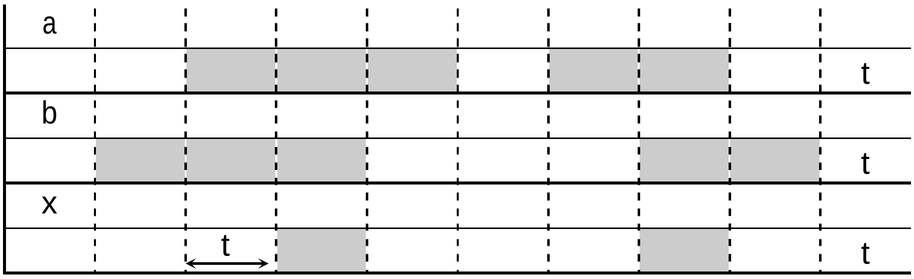

a
t
b
t
x
t
t

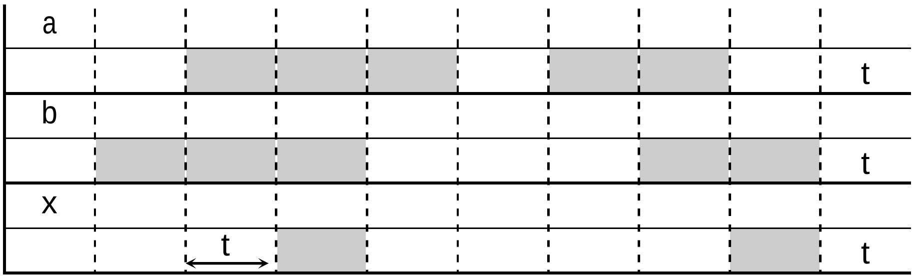

a
t
b
t
x
t
t

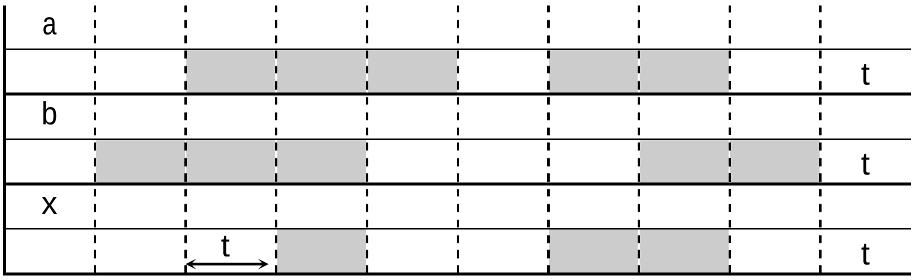

a
t
b
t
x
t
t

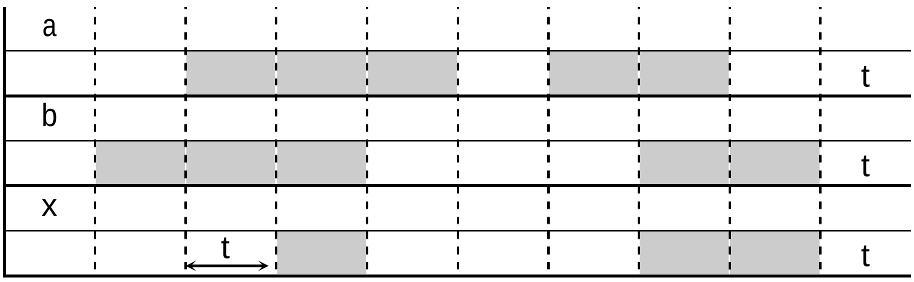

a
t
b
t
x
t
t

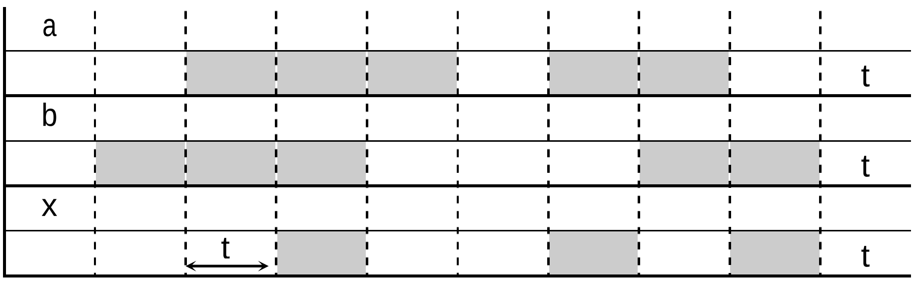

a
t
b
t
x
t
t

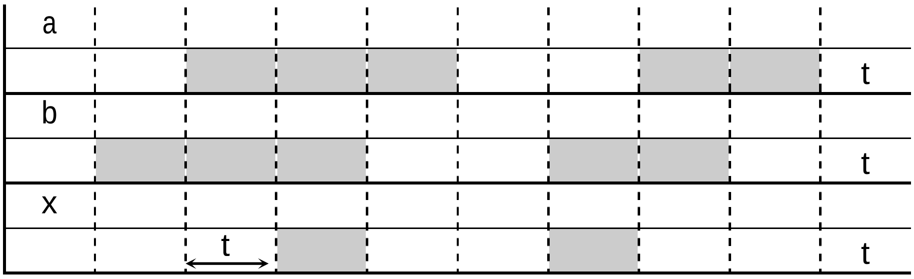

a
t
b
t
x
t
t

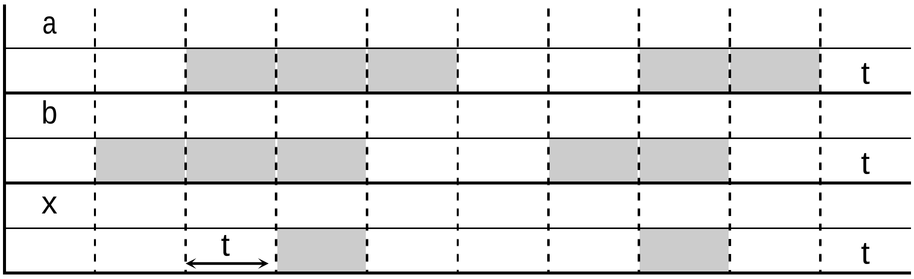

a
t
b
t
x
t
t

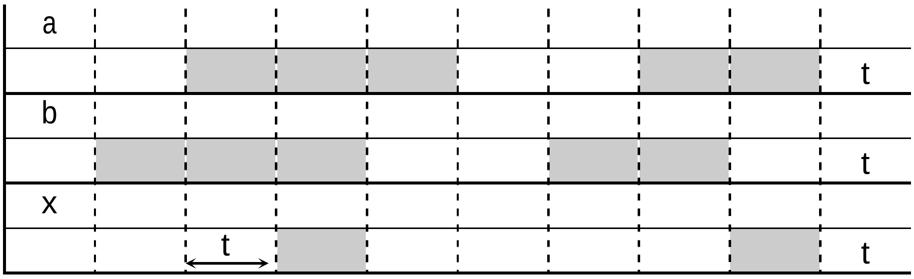

a
t
b
t
x
t
t

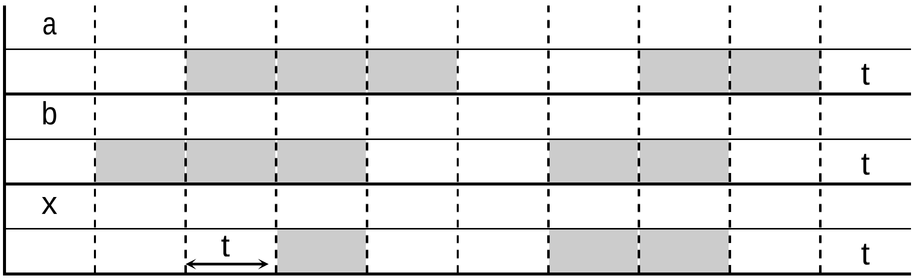

a
t
b
t
x
t
t

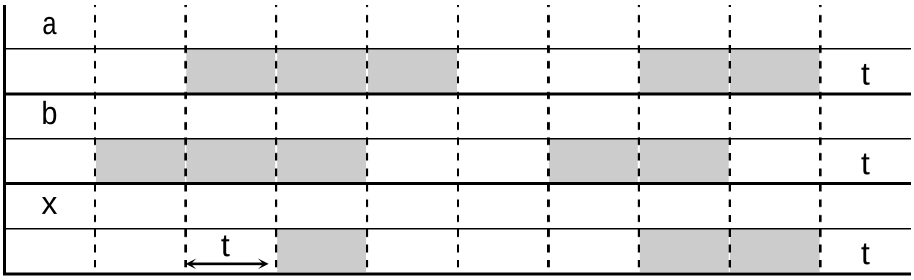

a
t
b
t
x
t
t

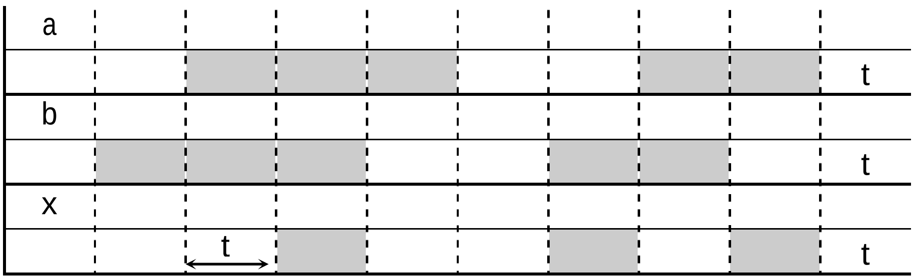

a
t
b
t
x
t
t


a
t
b
t
x
t
t


a
t
b
t
x
t
t


a
t
b
t
x
t
t


a
t
b
t
x
t
t


a
t
b
t
x
t
t

4. **ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ**

Вспомогательный раздел УМК представлен учебной программой по дисциплине «Проектирование дискретных систем управления» для студентов специальности 1-53 01 05 «Автоматизированные электроприводы».

Белорусский национальный технический университет

УТВЕРЖДАЮ
Декан факультета информационных технологий и робототехники
________________ Е.Е Трофименко
____
Регистрационный № УД-________/р.

Проектирование дискретных систем управления

Учебная программа для специальности:
1 53 01 05 «Автоматизированные электроприводы»

Факультет информационных технологий и робототехники

Кафедра "Электропривод и автоматизация промышленных установок и технологических комплексов»

Дневное обучение		Заочное обучение	
Курс	3	Курс	3, 4
Семестр	5	Семестр	5, 6
Лекции	50 часов	Лекции	12 часов
Практические занятия	18 часов	Практические занятия	4 часа
Курсовая работа	5 семестр	Курсовая работа	7 семестр
Зачет	5 семестр	Зачет	6 семестр
Всего аудиторных часов по дисциплине:	68		16
Всего часов по дисциплине:	170		

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная программа «Проектирование дискретных систем управления» разработана для специальности 1-53 01 05 «Автоматизированные электроприводы».

Цель изучения и преподавания дисциплины «Проектирование дискретных систем управления» заключается в овладении студентами знаний и умений для описания, проектирования и наладки дискретных систем управления.

Полученные знания и умения студенты в дальнейшем применяют в изучаемых далее дисциплинах учебного плана «Системы управления электроприводами», «Наладка и диагностика электропривода», «Проектирование систем управления робототехническими комплексами», в курсовом и дипломном проектировании.

Задача дисциплины «Проектирование дискретных систем управления» состоит в получении студентами основных теоретических знаний о методах описания и преобразования дискретных управляющих устройств, синтеза комбинационных и дискретных автоматов с памятью, повышения надёжности дискретных управляющих устройств.

Теоретические сведения должны быть закреплены на практике в ходе проведения практических занятий и выполнения курсовой работы.

В результате освоения дисциплины студент должен:

знать:

- логические основы описания и преобразования структуры дискретных управляющих устройств;
- методы задания и синтеза дискретных управляющих устройств;
- способы повышения надёжности дискретных управляющих устройств.

уметь:

- давать логическое описание алгоритма функционирования дискретного управляющего устройства;
- синтезировать оптимальную структуру дискретного управляющего устройства;
- реализовывать полученную структуру на различной элементной базе;
- выполнять наладку и обслуживание дискретных управляющих устройств.

приобрести навыки:

- описания и преобразования дискретных управляющих устройств с использованием аппарата алгебры логики;
- использования формальных методов при синтезе дискретных управляющих устройств;
- обеспечения надёжности дискретных управляющих устройств.

Методы (технологии) обучения

Основными методами (технологиями) обучения, отвечающими целям изучения дисциплины, являются:

- методы словесного обучения, реализуемые на лекционных занятиях;

– элементы учебно-исследовательской деятельности, реализация творческого подхода, реализуемые на практических занятиях и при самостоятельной работе;

– проектные технологии, используемые при проектировании конкретного объекта, реализуемые при выполнении курсовой работы.

Организация самостоятельной работы студентов

При изучении дисциплины используются следующие формы самостоятельной работы:

– контролируемая самостоятельная работа в виде решения индивидуальных задач в аудитории во время проведения практических занятий под контролем преподавателя в соответствии с расписанием;

– подготовка курсовой работы по индивидуальным заданиям, в том числе разноуровневым заданиям.

Диагностика компетенций студента

Оценка уровня знаний студента производится по десятибалльной шкале.

Для оценки достижений студента используется следующий диагностический инструментарий:

– защита курсовой работы;

– сдача зачета по дисциплине.

СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Раздел I. ЛОГИЧЕСКИЕ ОСНОВЫ ОПИСАНИЯ И ПРЕОБРАЗОВАНИЯ СТРУКТУРЫ ДИСКРЕТНЫХ УСТРОЙСТВ

Тема 1.1 Введение. Задачи, решаемые при проектировании дискретных систем управления

Выбор оптимальных вариантов структур отдельных устройств и всей системы в целом. Обеспечение высокой надежности проектируемого устройства. Сокращение времени проектирования, изготовления и наладки устройства, возможность реализации устройств на различных элементных базах. Использование автоматизированного проектирования.

Развитие методов проектирования дискретных устройств.

Тема 1.2 Основные определения. Способы задания функций алгебры логики

Понятие о дискретных устройствах. Алгебра логики. Логическая переменная. Логическая функция. Способы задания функций алгебры логики (табличный, координатный, числовой, аналитический).

Тема 1.3 Логические функции одной и двух переменных и их реализация

Общее число наборов логической функции. Число функций от n аргументов. Логические функции одной переменной и их реализация на контактных и бесконтактных элементах. Логические функции двух переменных и их реализация.

Тема 1.4 Аксиомы и теоремы алгебры логики

Основные аксиомы алгебры логики (дистрибутивности, коммутативности, ассоциативности, операций с значениями переменных). Законы алгебры логики (нулевого и универсального множества, сохранения степени, двойной инверсии, поглощения, склеивания). Разложение булевых функций: первая и вторая формулы разложения. Правила упрощения схем.

Тема 1.5 Нормальные и совершенные нормальные формы логических функций

Элементарные дизъюнкция и конъюнкции. Нормальные дизъюнктивная и конъюнктивная формы функций. Конституанты единицы и нуля. Совершенные дизъюнктивная и конъюнктивная нормальные формы функций. Переход от нормальных форм к совершенным нормальным формам. Запись совершенных нормальных форм функций по таблицам истинности.

Тема 1.6 Понятие о базисе. Базис функции Шеффера и функции Пирса

Понятие о базисе. Базис функции Шеффера: свойства функции Шеффера. Переход от нормальных форм булевых функций к записи в базисе функции Шеффера. Базис функции Пирса. Свойства функции Пирса. Переход от нбормальных форм булевых функций к записи в базисе функции Вебба. Логические элементы, реализующие функцию Шеффера и функцию Пирса.

Тема 1.7 Минимизация функций алгебры логики

Постановка задачи. Методы минимизации. Алгебраический метод минимизации с помощью теоремы Квайна. Алгебраический метод минимизации с помощью теоремы Блэка. Метод минимизации с помощью карт Карно. Процедура минимизации. Скобочные формы булевых функций.

Тема 1.8 Понятие о булевых матрицах

Мостиковые структуры. Булевы матрицы. Основные свойства булевых матриц. Запись логических функций проводимости цепи между узлами мостиковой структуры.

Раздел II. СИНТЕЗ КОМБИНАЦИОННЫХ АВТОМАТОВ

Тема 2.1 Постановка задачи синтеза комбинационных автоматов. Запись условий работы комбинационных автоматов

Определение комбинационных автоматов. Способы задания комбинационных автоматов.

Тема 2.2 Синтез структур комбинационных автоматов на контактных и бесконтактных логических элементах

Порядок синтеза комбинационных автоматов. Минимизация комбинационных автоматов. Синтез контактных схем. Синтез структур комбинационных автоматов на бесконтактных логических элементах. Особенности построения структур комбинационных автоматов на бесконтактных логических элементах.

Тема 2.3 Учёт недоопределённости комбинационного автомата при его задании и синтезе

Полностью и не полностью определённые комбинационные автоматы. Обязательные, запрещённые и условные наборы. Доопределение комбинационного автомата с целью получения оптимальной структуры.

Тема 2.4 Минимизация логических функций, описывающих комбинационный автомат со многими выходами

Постановка задачи. Метод многовыходной функции. Порядок минимизации. Особенности минимизации при реализации комбинационного автомата на
контактных и бесконтактных элементах.

Раздел III. СИНТЕЗ ДИСКРЕТНЫХ АВТОМАТОВ С ПАМЯТЬЮ

Тема 3.1 Дискретный автомат с памятью. Общие понятия и определения

Дискретный автомат. Функции выходов и переходов. Математическая модель дискретного автомата с памятью. Модель Мура и Мили. Синхронный и асинхронный автоматы.

Тема 3.2 Способы задания дискретных автоматов с памятью

Переход от словесной формулировки условий работы к автоматному описанию. Способы задания дискретного автомата с памятью: таблицы переходов и выходов, графы переходов и выходов, матрицы переходов.

Тема 3.3 Постановка задачи синтеза дискретного автомата с памятью

Полностью и неполностью определенные автоматы. Допустимые входные последовательности. Ограничения на входные последовательности. Неопределенные внутренние состояния. Абстрактный синтез. Структурный синтез дискретных автоматов с память. Последовательность синтеза.

Тема 3.4 Минимизация памяти автоматов

Задачи минимизации. Метод треугольной таблицы. Последовательность минимизации. Совместимые и условно совместимые состояния. Группы и классы совместимости. Определение минимального класса совместимости. Построение минимизированной автоматной таблицы.

Тема 3.5 Кодирование состояний автоматов с памятью

Задача кодирования. Основы формализации процесса кодирования. Разбиения внутренних состояний, блоки разбиений. Внешнее и внутреннее разбиения состояний автомата. Множества порядка 1. Порядок кодирования. Однозначность кодирования.

Тема 3.6 Устранение критических состязаний элементов памяти

Причины появления состязаний элементов памяти. Выявление состязаний. Допустимые и недопустимые состязания. Меры устранения недопустимых состязаний.

Тема 3.7 Реализация памяти петлями обратных связей

Порядок синтеза автомата с реализацией памяти петлями обратных связей. Запись логических выражений для функции перехода. Реализация памяти автомата на контактных и бесконтактных элементах.

Тема 3.8 Реализация памяти автомата на триггерах

Основные типы триггеров, логика их функционирования. Порядок синтеза автомата с реализацией памяти на триггерах. Определение функций возбуждения памяти для входов триггера.

Тема 3.9 Синтез комбинационной схемы выходов автомата

Порядок синтеза комбинационной схемы выходов. Определение логических выражений для выходных функций. Реализация комбинационной схемы выходов на контактных и бесконтактных элементах.

Тема 3.10 Синтез дискретных автоматов с временными зависимостями

Особенности описания дискретных автоматов с памятью при наличии временных зависимостей. Порядок синтеза. Реализация дискретных автоматов с временными зависимостями на контактных и бесконтактных элементах.

Тема 3.11 Синтез динамических дискретных автоматов

Понятие о динамическом автомате. Способы описания динамических автоматов. Порядок синтеза.

СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ

Целью курсовой работы является применение формальных методов проектирования дискретных управляющих устройств в практике инженерного проектирования. При выполнении курсовой работы выполняется синтез схемы управления перемещением механизма между тремя положениями по заданному циклу с наличием временных зависимостей (с различными циклами). Объём курсовой работы составляет порядка 40 печатных листов, для выполнения работы необходимо затратить не менее 20 часов.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА

Номер раздела, темы, занятия	Название раздела, темы, занятия; перечень изучаемых вопросов	Количество аудиторных часов			Материальное обеспечение занятия (наглядные, методические пособия и др.)	Литература	Форма контроля знаний
		Лекции	практические (семинарские) занятия	Лабораторные занятия			
1.	**Логические основы описания и преобразования структуры дискретных устройств. (22 ч.)**	16	6				
1.1	Введение. Задачи, решаемые при проектировании дискретных систем управления. 1. Цели и задачи курса. Понятие дискретного автомата. 2. Классификация дискретных управляющих устройств. Достоинства формальных методов проектирования.	2/2				[1]	зачёт
1.2	Основные определения. Способы задания функций алгебры логики. 1. Алгебра логики. Логическая переменная. Логическая функция. 2. Способы задания функций алгебры логики (табличный, координатный, числовой, аналитический).	2	1			[1] [2]	зачёт
1.3	Логические функции одной и двух переменных и их реализация. 1. Общее число наборов логической функции. Число функций от *n* аргументов. Логические функции одной переменной и их реализация на контактных и бесконтактных элементах. 2. Логические функции двух переменных и их реализация.	2				[1] [2]	зачёт
1.4	Аксиомы и теоремы алгебры логики. 1. Основные аксиомы алгебры логики (дистрибутивности, коммутативности, ассоциативности, операций с значениями переменных). 2. Законы алгебры логики (нулевого и универсального множества, сохранения степени, двойной инверсии, поглощения 9 ия). 3. Разложение булевых функций: первая и вторая формулы разложения. Правила упрощения схем.	2	1			[1] [2]	зачёт

1.5	Нормальные и совершенные нормальные формы логических функций. 1. Элементарные дизъюнкция и конъюнкции. Нормальные дизъюнктивная и конъюнктивная формы функций. 2. Конституанты единицы и нуля. Совершенные дизъюнктивная и конъюнктивная нормальные формы функций. Переход от нормальных форм к совершенным нормальным формам. Запись совершенных нормальных форм функций по таблицам истинности.	2	1			[1] [2]	зачёт
1.6	Понятие о базисе. Базис функции Шеффера и функции Пирса. 1. Понятие о базисе. Базис функции Шеффера: свойства функции Шеффера. Переход от нормальных форм булевых функций к записи в базисе функции Шеффера. 2. Базис функции Пирса. Свойства функции Пирса. Переход от нбормальных форм булевых функций к записи в базисе функции Вебба. Логические элементы, реализующие функцию Шеффера и функцию Пирса.	2	1			[2]	зачёт
1.7	Минимизация функций алгебры логики. 1. Постановка задачи. Методы минимизации. 2. Алгебраический метод минимизации с помощью теоремы Квайна. 3. Алгебраический метод минимизации с помощью теоремы Блэка. 4. Метод минимизации с помощью карт Карно. Процедура минимизации. Скобочные формы булевых функций.	2/2	2			[1] [2]	зачёт
1.8	Понятие о булевых матрицах. 1. Мостиковые структуры. Булевы матрицы. Основные свойства булевых матриц. Запись логических функций проводимости цепи между узлами мостиковой структуры.	2				[3]	зачёт
2.	**Синтез комбинационных автоматов**. (10 **ч.)**	8	2				
2.1	Постановка задачи синтеза комбинационных автоматов. Запись условий работы комбинационных автоматов. 1. Определение комбинационных автоматов. Способы задания комбинационных автоматов.	2/2				[2]	зачёт
2.2	Синтез структур комбинационных автоматов на конта 10 он-тактных логических элементах. 1. Порядок синтеза комбинационных автоматов.	2	2			[2]	зачёт

	2. Минимизация комбинационных автоматов. 3. Синтез контактных схем. 4. Синтез структур комбинационных автоматов на бесконтактных логических элементах. Особенности построения структур комбинационных автоматов на бесконтактных логических элементах.						
2.3	Учёт недоопределённости комбинационного автомата при его задании и синтезе. 1. Полностью и не полностью определённые комбинационные автоматы. Обязательные, запрещённые и условные наборы. 2. Доопределение комбинационного автомата с целью получения оптимальной структуры.	2				[1] [2]	зачёт
2.4	Минимизация логических функций, описывающих комбинационный автомат со многими выходами. 1. Постановка задачи. Метод многовыходной функции. Порядок минимизации. 2. Особенности минимизации при реализации комбинационного автомата на контактных и бесконтактных элементах.	2				[1] [2]	зачёт
3.	**Синтез дискретных автоматов с памятью** (36 **ч.)**	26	10				
3.1	Дискретный автомат с памятью. Общие понятия и определения. 1. Дискретный автомат. Функции выходов и переходов. 2. Математическая модель дискретного автомата с памятью. Модель Мура и Мили. Синхронный и асинхронный автоматы.	2/1				[1] [2]	зачёт
3.2	Способы задания дискретных автоматов с памятью. 1. Переход от словесной формулировки условий работы к автоматному описанию. 2. Способы задания дискретного автомата с памятью: таблицы переходов и выходов, графы переходов и выходов, матрицы переходов.	2/1	1		Учебно-методическое пособие	[1] [2] [6]	зачёт
3.3	Постановка задачи синтеза дискретного автомата с памятью. 1. Полностью и неполностью определенные автомать 11 мые входные последовательности. Ограничения на вхо, едовательности. Неопределенные внутренние состояния. 2. Абстрактный синтез. Структурный синтез дискретных автоматов с память. Последовательность синтеза.	2				[1] [2]	зачёт

3.4	Минимизация памяти автоматов. 1. Задачи минимизации. Метод треугольной таблицы. Последовательность минимизации. Совместимые и условно совместимые состояния. 2. Группы и классы совместимости. Определение минимального класса совместимости. 3. Построение минимизированной автоматной таблицы.	4/2	2/1		Учебно-методическое пособие	[1] [2] [6]	зачёт
3.5	Кодирование состояний автоматов с памятью. 1. Задача кодирования. Основы формализации процесса кодирования. 2. Разбиения внутренних состояний, блоки разбиений. Внешнее и внутреннее разбиения состояний автомата. Множества порядка 1. Порядок кодирования. Однозначность кодирования.	2/1	1/1		Учебно-методическое пособие	[1] [2] [6]	зачёт
3.6	Устранение критических состязаний элементов памяти. 1. Причины появления состязаний элементов памяти. Выявление состязаний. Допустимые и недопустимые состязания. 2. Меры устранения недопустимых состязаний.	2	1		Учебно-методическое пособие	[1] [2] [6]	зачёт
3.7	Реализация памяти петлями обратных связей. 1. Порядок синтеза автомата с реализацией памяти петлями обратных связей. Запись логических выражений для функции перехода. 2. Реализация памяти автомата на контактных и бесконтактных элементах.	2	1/1		Учебно-методическое пособие	[1] [5] [6]	зачёт
3.8	Реализация памяти автомата на триггерах. 1. Основные типы триггеров, логика их функционирования. 2. Порядок синтеза автомата с реализацией памяти на триггерах. Определение функций возбуждения памяти для входов триггера.	2/1	1/1		Учебно-методическое пособие	[1] [5] [6]	зачёт
3.9	Синтез комбинационной схемы выходов автомата. 1. Порядок синтеза комбинационной схемы выходов 12 ение логических выражений для выходных функций. 2. Реализация комбинационной схемы выходов на контактных и бесконтактных элементах.	2	1		Учебно-методическое пособие	[1] [2] [6]	зачёт
3.10	Синтез дискретных автоматов с временными зависимостями. 1. Особенности описания дискретных автоматов с памятью при наличии временных зависимостей. Порядок синтеза.	4	2		Учебно-методическое	[1] [4] [6]	зачёт

	2. Реализация дискретных автоматов с временными зависимостями на контактных и бесконтактных элементах.				пособие		
3.11	Синтез динамических дискретных автоматов. 1. Понятие о динамическом автомате. Способы описания динамических автоматов. 2. Порядок синтеза.	2				[4]	зачёт

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

ОСНОВНАЯ ЛИТЕРАТУРА

1. Ершова Е.Б., Рогинский В.Н., Маркин Н.П. Основы дискретной автоматики в электросвязи. – М.: Связь, 1980. – 231 с.
2. Г. Р. Грейнер. Проектирование бесконтактных управляющих устройств промышленной автоматики. Москва “Энергия”, 1977. – 384 с.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

3. Трачик В. Дискретные устройства автоматики. Пер. с польского./ Под ред. Д.С. Поспелова. – М.: Энергия, 1978. – 456 с.
4. Баранов С.И. Синтез микропрограммных аппаратов. – Л.: Энергия, 1979. - 232 с.
5. Цифровые интегральные микросхемы: Справочник/ М. И. Богданович, И. Н. Грель, С. А. Дубина и др. – 2-е изд., перераб. и доп. – Мн.: Беларусь, Полымя. 1996. – 596 с.
6. Проектирование дискретных систем управления: методическое пособие по выполнению курсовой работы для студентов специальности 1-53 01 05 «Автоматизированные электроприводы» / А. А. Мигдалёнок. – Минск: БНТУ, 2011. – 55 с.