Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Воронежская государственная лесотехническая академия»

Ю.С. Сербулов
Е.А. Аникеев

Методические указания
для выполнения лабораторных работ по дисциплине
«Цифровые автоматы»
для студентов специальности
230400.62 – «Информационные системы и технологии»

Воронеж 2015

Р е ц е н з е н т : начальник лаборатории ОАО НИИЭТ
к-т. техн. наук А.И. Яньков

Сербулов Ю.С., Аникеев, Е.А.

Методические указания по выполнению лабораторных работ по дисциплине «Цифровые автоматы» для студентов специальности 230400.62 – «Информационные системы и технологии» / Ю. С. Сербулов, Е.А. Аникеев – Воронеж: Воронежская государственная лесотехническая академия, 2015. – 63с.

Аникеев Евгений Александрович

Методические указания по выполнению лабораторных работ по дисциплине «Цифровые автоматы» для студентов специальности 230400.62 – «Информационные системы и технологии»

Оглавление

Правила оформления отчёта..5
Лабораторная работа №1. Системы счисления 6
Теоретическая часть..6
Варианты заданий..8
Порядок выполнения работы..8
Контрольные вопросы 9
Лабораторная работа №2. Перевод чисел из одной системы счисления в другую..9
Теоретическая часть..9
Варианты заданий..10
Порядок выполнения работы..12
Контрольные вопросы 12
Лабораторная работа №3. Законы алгебры логики 13
Теоретическая часть..13
Варианты заданий..15
Порядок выполнения работы..15
Контрольные вопросы 15
Лабораторная работа №4. Переключательные функции..16
Теоретическая часть..16
Варианты заданий..19
Порядок выполнения работы..20
Контрольные вопросы 20
Лабораторная работа №5. Минимизация логических функций..20
Теоретическая часть..20
Варианты заданий..22
Порядок выполнения работы..22
Контрольные вопросы 22
Лабораторная работа №6. Реализация логических функций в различных базисах 23
Теоретическая часть..23
Варианты заданий..28
Порядок выполнения работы..28
Контрольные вопросы 29
Лабораторная работа №7. Формы представления цифровых автоматов с памятью (автоматов Мили)..29
Теоретическая часть..29
Варианты заданий..35
Порядок выполнения работы..36
Контрольные вопросы 36
Лабораторная работа №8. Формы представления цифровых автоматов без памяти (автоматов Мура) 37
Теоретическая часть..37
Варианты заданий..39
Порядок выполнения работы..41
Контрольные вопросы 41
Лабораторная работа №9. Элементарные конечные автоматы: структура и функционирование однотактного RS-триггера 42
Теоретическая часть..42
Варианты заданий..47
Порядок выполнения работы..48
Контрольные вопросы 48
Лабораторная работа №10. Элементарные конечные автоматы: структура и функционирование двухтактного RS-триггера 49

Теоретическая часть ... 49
Варианты заданий ... 50
Порядок выполнения работы ... 50
Контрольные вопросы ... 50
Лабораторная работа №11. Элементарные конечные автоматы: структура и функционирование D-триггера, T-триггера и JK-триггера ... 50
Теоретическая часть ... 50
Варианты заданий ... 55
Порядок выполнения работы ... 55
Контрольные вопросы ... 55
Лабораторная работа №12. Синтез функций активации триггеров ... 56
Теоретическая часть ... 56
Варианты заданий ... 61
Порядок выполнения работы ... 63
Контрольные вопросы ... 64

Правила оформления отчёта

Отчёт по лабораторной работе оформляется студентом в отдельной от лекций рабочей тетради, не использующей сменные блоки листов.

Содержание отчёта:

1. Титульный лист.
2. Цель лабораторной работы.
3. Теоретическое введение.
4. Выполнение работы.
5. Вывод по лабораторной работе.

Титульный лист содержит: наименования вышестоящей организации, учебного заведения, кафедры, на которой выполняется работа, а также номер и название самой работы. Также указывается, кто выполнил работу (фамилия, инициалы и номер группы студента) и кто проверил работу (учёные степень, звание, фамилия и инициалы). В нижней части титульного листа указывается город и годы выполнения работы.

Цель выполнения приведена в описании лабораторной работы. Она размещается на странице, следующей за титульным.

Теоретическое введение содержит основные положения, на которых основана выполняемая работа. Его объем должен составлять не менее 1 страницы.

В разделе «Выполнение работы» приводятся исходные данные, вычисления и схемы, необходимые для достижения цели работы. Подробное содержание этого раздела приводится в разделе «Порядок выполнения работы» для каждой лабораторной работы.

Вывод содержит изложение результатов работы, полученных в ходе её выполнения, с учётом цели работы.

Лабораторная работа №1. Системы счисления

Цель работы: получение навыков выполнения арифметических операций с числами в двоичной системе счисления

Теоретическая часть

В современных вычислительных системах вся информация представлена в виде кодов чисел с использованием двоичной системы счисления. Изучение систем счисления необходимо для понимания основ функционирования вычислительных машин.

Система счисления – это способ изображения и именования чисел с помощью символов, имеющих количественное значение.

Существуют позиционные и непозиционные системы счисления.

В непозиционных системах значение каждого символа в числе не зависит от его местоположения. Например, в римской системе XXX – тридцать. Эти системы не получили широкого применения для расчётов, так как способы чтения этих чисел и арифметических операций над ними довольно сложны.

В позиционных системах счисления значение каждого символа в числе зависит от его местоположения в этом числе. Например: 111 – сто одиннадцать. Это число состоит из трёх единиц, количественное значение которого зависит от занимаемого места. Первая единица имеет количественное значение «сто», вторая – «десять», третья – «один».

В позиционной системе счисления любое число можно представить в виде:

$$A = \pm a_{m-1}\ a_{m-2}\ \mathbf{K}\ a_1\ a_0\ , a_1\ a_2\ \mathbf{K}\, a_k\ .$$

Позиции, пронумерованные $m-1, m, \mathbf{K}\, 0$ и $1, 2, \mathbf{K}\, k$ называются разрядами числа. Дробные разряды отделяются от целых запятой. Общее число разрядов составляет $m+k$.

Каждая цифра разрядов числа может принимать одно из значений от 0 до $N-1$. При этом число N называется основанием системы счисления. Оно численно равно количеству цифр или других символов, используемых в изображении чисел в данной системе счисления. Кроме того, основание системы счисления показывает, во сколько раз единица некоторого разряда числа больше единицы предыдущего разряда числа.

Учитывая это, любое число в позиционной системе счисления можно представить в виде:

$$A = \pm\left[a_{m-1}N^{m-1} + a_{m-2}N^{m-2} + \mathbf{K} + a_1N^1 + a_0N^0 + a_{-1}N^{-1} + \mathbf{K} + a_{-k}N^{-k}\right].$$

Например,

$$1995{,}21 = 1\cdot 10^3 + 9\cdot 10^2 + 9\cdot 10^1 + 5\cdot 10^0 + 2\cdot 10^{-1} + 1\cdot 10^{-2}.$$

Такая форма записи числа называется канонической. Основание системы счисления здесь 10. Название системы счисления определяется её основанием. Например, система, включающая в себя десять цифр $0,1,2,3,4,5,6,7,8,9$ – десятичная.

В аппаратной части современных вычислительных машин все операции выполняются в двоичной системе счисления, включающей в себя две цифры 0 и 1. Использование такой системы имеет два преимущества:

6. Для кодирования чисел можно использовать достаточно простые и надёжные элементы, имеющие два устойчивых состояния, одно из которых кодируется нулём, другое – единицей.
7. Выполнение арифметических операций в двоичной системе является наиболее простым.

Кроме двоичной системы, в программировании вычислительных машин используются восьмеричная и шестнадцатеричная системы счисления, содержащие соответственно 8 и 16 символов. Так как используемая обычно десятичная система содержит только десять символов, в шестнадцатеричной системе в качестве цифр используются буквы A, B, C, D, E, F.

Таблица 1 Таблица соответствий чисел систем счисления

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Арифметические операции с числами в любой позиционной системе счисления производятся поразрядно при помощи определения соответствующих элементарных операций.

Элементарные операции сложения в двоичной системе:

$0+0=0$

$1+0=0+1=1$

$1+1=10$

Пример сложения чисел в двоичной системе:

$$
\begin{array}{r}
+\quad 11011,101 \\
1110,001 \\
\hline
101001,110
\end{array}
$$

Элементарные операции умножения в двоичной системе:

$0\cdot 0=0$

$1\cdot 0=0\cdot 1=0$

$1\cdot 1=1$

Умножение производится, начиная с младших или старших разрядов множителей, так как при умножении любых цифр в двоичной системе счисления не возникает единиц переноса в соседний старший разряд.

Пример умножения чисел в двоичной системе:

$$
\begin{array}{r}
\times\quad 11101,101 \\
100,111 \\
\hline
11101101 \\
11101101 \\
11101101 \\
00000000 \\
00000000 \\
11101101 \\
\hline
10010000,011011
\end{array}
$$

Варианты заданий

№	Числа		№	Числа	
1	10011 110110 10011 111001	10011 110 11111 1100	11	1000 101011 1010 10101	1100 101 101010 1011
2	110 110000 101010 1110	111 100000 1101 11110	12	1010101 1010100 100101 111	11101 101001 10101 1010010
3	10000 110010 110111 10	101000 110111 110001 110101	13	110000 110011 111100 101010	1010 10101 110101 11
4	101100 10110 1111 1000110	101101 1100 110 11	14	10101 1111111 1101 1010	10110 101001 11111 101
5	101100 110100 11010 11000	100111 100001 1001011 1011	15	1101001 110011 10110 1110	111011 10101 1010000 1110
6	1010110 101 101110 111010	101 11010 101111 1011100	16	101011 101001 1100 101001	10100 1010 110101 10010101
7	10101001 100 1100 1101011	10101 11110 10110 1010101	17	10110101 101110 1101010 111010	10110 11 10101 1111
8	101010 100011 1010110 100	1010000 11111 110001 1010110	18	111100 1010001 111101 1010	101011 10110 11010 11010
9	110111 11001 10100 10100	111010 1010101 10100 111	19	1011010 110101 11010 1010011	1011011 110101 10101010 101010
10	110111 11100 111000 1110	101111 1010101 111110 1111	20	101101 11000 111001 1010101	1101010 101001 101010 1010101

Порядок выполнения работы

1. Произвести сложение и умножение чисел из двух столбцов согласно своему варианту из таблицы вариантов заданий. При проверке результата использовать калькулятор.
2. Оформить отчёт по лабораторной работе (см. пункт содержания «Правила оформления отчета»). Раздел «Выполнение работы» содержит:
   - вариант задания на лабораторную работу;
   - запись арифметических действий согласно заданию.

Контрольные вопросы

1. Что такое система счисления?
2. Какие существуют системы счисления?
3. Как формируются числа в непозиционной системе?
4. Как можно представить число в позиционной системе?
5. Как выглядит каноническая форма записи числа?
6. Каковы преимущества использования двоичной системы в аппаратной части вычислительной техники?
7. Какие ещё системы счисления используются, какие в них используются цифры?
8. Проведите сложение и умножение чисел, данных преподавателем.

Лабораторная работа №2. Перевод чисел из одной системы счисления в другую

Цель работы: получение навыков выполнения перевода чисел из одной системы счисления в другую

Теоретическая часть

Одним из распространённых методов перевода чисел является универсальный алгоритм. Кроме него, применяют перевод при помощи весов разрядов, схему Горнера и другие алгоритмы.

При использовании универсального алгоритма отдельно переводятся целая и дробная части. Для перевода целой части необходимо последовательно делить целую часть числа и образующиеся при делении целые частные, записанные в исходной системе счисления, на основание новой системы счисления, записанное в исходной системе счисления. Образующиеся на каждом шагу деления остатки представляют собой цифры разрядов числа в новой системе счисления, записанные цифрами исходной системы счисления. При этом первый остаток – это цифра младшего разряда числа, а последний остаток – цифра старшего разряда числа

Для перевода дробной части числа необходимо последовательно умножать дробную часть числа и дробные части образующихся произведений, записанные в исходной системе счисления, на основание новой системы счисления, записанное в исходной системе счисления. При этом целые части образующихся на каждом шагу умножения произведений представляют собой цифры разрядов дробной части числа в новой системе счисления, записанные в исходной системе счисления. При этом целая часть нового произведения – это старший разряд дробной части числа, стоящего справа от запятой.

Рассмотрим пример.

$A_{10} = 30{,}6$. $A_2 = ?$

$$\begin{array}{r|r|r|r|r|l}
30 & 2 & & & & \\
\underline{30} & 15 & 2 & & & \\
0 & \underline{14} & 7 & 2 & & \\
 & 1 & \underline{6} & 3 & 2 & \\
 & & 1 & \underline{2} & 1 & 2 \\
 & & & 1 & \underline{0} & 0 \\
 & & & & 1 &
\end{array}$$

1	1	1	1	0	

$$\begin{array}{r}
\times\ 0{,}\ 6 \\
\underline{2} \\
1{,}\ 2 \\
\underline{2} \\
0{,}\ 4 \\
\underline{2} \\
0{,}\ 8 \\
\underline{2} \\
1{,}\ 6
\end{array}$$

1	0	0	1		

В результате $A_2 = 11110{,}1001$.

Перевод чисел с использованием весов разрядов удобно использовать для перевода из двоичной системы счисления в десятичную. Вес разряда – это число, показывающее, во сколько раз единица данного разряда больше или меньше единицы младшего разряда целой части числа. Если в формуле канонической записи числа все цифры записать в новой системе счисления и выполнить указанные в формуле арифметические действия, то получим число в новой системе счисления.

Пример:

1	0	0	0	1	0	1	- число
64	32	16	8	4	2	1	- веса разрядов

Просуммировав ненулевые разряды, получим: 64+4+1 = 69

Перевод чисел по схеме Горнера представляет собой вычисления по преобразованной формуле канонической формы числа, например:

$$A_N = a_3N^3 + a_2N^2 + a_1N^1 + a_0N^0 = ((a_3N + a_2)N + a_1)N + a_0.$$

Для перевода числа из одной системы счисления в другую надо:

Умножить цифру старшего разряда числа, записанную в новой системе счисления, на основание исходной системы счисления, записанное в новой системе счисления. Затем к полученному результату прибавить цифру следующего разряда числа, записанного в новой системе счисления, и полученную сумму снова умножить на основание исходной системы счисления, и так далее до последнего, младшего разряда. После прибавления младшего разряда умножение на основание системы счисления не производится.

Примеры:

$A_8 = 312$. $\quad A_{10} = (3\cdot 8 + 1)\cdot 8 + 2 = 202$.

$A_{16} = 2EF$. $\quad A_{10} = (2\cdot 16 + 14)\cdot 16 + 15 = 751$.

Варианты заданий

Используя универсальный алгоритм, выполнить перевод чисел: $A_{10} \to A_K$.

№	A_{10}	K	№	A_{10}	K
1	28 111	3 15	11	34 114	3 15
2	31 201	4 14	12	32 213	4 14
3	42 141	5 13	13	25 174	5 13
4	12 112	6 12	14	11 115	6 12
5	16 205	7 11	15	18 217	7 11
6	29 152	3 15	16	26 185	3 15
7	19 113	4 14	17	14 116	4 14
8	43 209	5 13	18	29 221	5 13
9	13 163	6 12	19	35 196	6 12
10	17 113	7 11	20	23 117	7 11

2. Выполнить перевод чисел $A_2 \to A_{10}$ при помощи весов разрядов.

№	A_{10}	№	A_{10}
1	1101110 110111 11010	11	1010101 1111010 1011111
2	1101011 1011111 10111111	12	11010110 1110 101101
3	1101111 11011111 11110	13	11101101 101111 101
4	110111 110001 1101010	14	1110101010 110 11011
5	1011 1010111 110101	15	1110101 10110101 10111
6	10110111 101011 101101	16	101101011 10101 11110
7	1011110 1010111 1110101	17	1011011 101011 1011101110
8	1111 11010 11111000	18	111010 1010100 101111
9	101010110 1111010 11011011	19	1111010101 11111 11000
10	1111101100 1100 1001011	20	11110101 11010 1000001

3. Выполнить перевод чисел с использованием схемы Горнера.

№	$A_8 \to A_{10}$	$A_{16} \to A_{10}$	№	$A_8 \to A_{10}$	$A_{16} \to A_{10}$
1	111 766	11A 921	11	121 354	EF11 4213
2	123 753	21AA 7531	12	132 327	CC2 4522
3	211 676	12AB 1242	13	243 312	BCA 1562
4	223 655	11B 1543	14	254 276	A1B2 148
5	311 543	AA1C 4234	15	365 259	CE44 9654
6	323 534	A1EF 7565	16	321 246	BD23 421
7	411 472	1AFB 4213	17	432 159	AC52 655
8	423 464	22AC 142	18	443 141	FF2 754
9	511 531	D23 852	19	554 137	E3C 4587
10	523 570	13A 432	20	565 124	153A 1239

Порядок выполнения работы

1. Выполнить перевод чисел из одной системы счисления в другую согласно своему варианту из трёх таблиц вариантов заданий. При проверке результата использовать калькулятор.
2. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:
   – вариант задания на лабораторную работу;
   – запись переводов чисел согласно заданию.

Контрольные вопросы

1. Как выполняется перевод целой части числа по универсальному алгоритму?
2. Что необходимо сделать для перевода дробной части числа по универсальному алгоритму?
3. Как при переводе чисел используется схема Горнера?
4. Как выполняется перевод чисел при помощи весов разрядов?
5. Произведите перевод чисел, данных преподавателем, при помощи указанного им метода.

Лабораторная работа №3. Законы алгебры логики

Цель работы: научиться использовать основные законы алгебры логики для преобразования логических функций.

Теоретическая часть

Все элементы современных вычислительных машин представляют собой цифровые автоматы. Для описания функционирования и синтеза этих устройств используется двоичная логика. Она определена на области, состоящей из двух элементов: 0 и 1. Для них определены следующие отношения:

Название	Обозначение
Эквивалентность	$x_1 = x_2$
Дизъюнкция (логическое сложение)	$x_1 \vee x_2$
Конъюнкция (логическое умножение)	$x_1 \wedge x_2$ или $x_1 \cdot x_2$ или $x_1 x_2$
Отрицание (логическая инверсия)	$\neg x_1$ или $\overline{x_1}$

Существует множество алгебр логики, в зависимости от того, каким образом определяются эти отношения. Для используемой в вычислительных машинах алгебры логики постулируются следующие отношения при выполнении операций:

Дизъюнкция:

$0 \vee 0 = 0$, $0 \vee 1 = 1 \vee 0 = 1$, $1 \vee 1 = 1$.

Конъюнкция:

$0 \cdot 0 = 0$, $0 \cdot 1 = 1 \cdot 0 = 0$, $1 \cdot 1 = 1$.

Инверсия:

$\overline{0} = 1$, $\overline{1} = 0$.

На основании этих постулатов можно вывести законы алгебры логики:

Закон единичных элементов:

$x \vee 0 = x$, $x \cdot 0 = 0$, $x \vee 1 = 1$, $x \cdot 1 = x$.

Законы отрицания:

Двойное отрицание:

$\overline{\overline{x}} = x$

Дополнение:

$x \vee \overline{x} = 1$, $x \cdot \overline{x} = 0$.

Двойственность (закон де Моргана)

$\overline{x_1 \vee x_2} = \overline{x}_1 \cdot \overline{x}_2$,

$\overline{x_1 \cdot x_2} = \overline{x}_1 \vee \overline{x}_2$.

Законы комбинационные:

Тавтология:

$x \cdot x \cdot x \cdot \mathbf{K} = x$, $x \vee x \vee x \vee \mathbf{K} = x$.

Переместительный:

$x_1 \vee x_2 = x_2 \vee x_1$, $x_1 \cdot x_2 = x_2 \cdot x_1$.

Сочетательный:

$x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3$,

$x_1 \cdot (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3$.

Распределительный:

$x_1 \cdot (x_2 \vee x_3) = x_1 \cdot x_2 \vee x_1 \cdot x_3$,

$x_1 \vee x_2 \cdot x_3 = (x_1 \vee x_2) \cdot (x_1 \vee x_3)$.

Склеивание:

$$x_1 \cdot x_2 \vee \overline{x_1} \cdot x_2 = x_2 \cdot \left(x_1 \vee \overline{x_1}\right) = x_2 .$$

Поглощение:

$$x_1 \vee x_1 \cdot x_2 = x_1 \cdot \left(1 \vee x_2\right) = x_1 .$$

При выполнении логических операций необходимо соблюдать их приоритетность. Наивысшим приоритетом обладает инверсия. Затем выполняется конъюнкция (логическое умножение), затем дизъюнкция (логическое сложение). Этот порядок может быть изменён при использовании скобок. При этом внутри скобок действия выполняются в порядке изложенного выше приоритета. Если выражение содержит множество скобок, то выполнение операций начинается с внутренних скобок.

Закон де Моргана часто применяется для преобразования логических функций с целью их упрощения. В обобщённом виде этот закон можно записать так:

$$\overline{f}\left(x_1, x_2, \ldots x_n ; \vee ; \cdot\right) = f\left(\overline{x_1}, \overline{x_2}, \ldots \overline{x_n} ; \cdot ; \vee\right).$$

Применение операций инвертирования произвольной комбинации двоичных переменных эквивалентно замене в ней исходных значений логических переменных их инверсными значениями при одновременной смене знака дизъюнкции на конъюнкцию, а конъюнкции на дизъюнкцию.

Например, преобразуем логической выражение, применив к нему двойное отрицание и обобщённый закон де Моргана:

$$\left(x_1 \vee \overline{x_2}\right) \cdot \left(\overline{x_3} \vee \overline{x_4}\right) = \overline{\overline{\left(x_1 \vee \overline{x_2}\right) \cdot \left(\overline{x_3} \vee \overline{x_4}\right)}} = \overline{\overline{x_1 \vee \overline{x_2}} \vee \overline{\overline{x_3} \vee \overline{x_4}}} = \overline{\overline{x_1} \cdot x_2 \vee x_3 \cdot x_4}$$

Проверим правильность преобразования. Для этого составим таблицу истинности этих выражений, подставляя в них все сочетания входящих в них переменных, и сравним значения функций.

№	x_1	x_2	x_3	x_4	$\left(x_1 \vee \overline{x_2}\right) \cdot \left(\overline{x_3} \vee \overline{x_4}\right)$	$\overline{\overline{x_1} \cdot x_2 \vee x_3 \cdot x_4}$
0	0	0	0	0	1	1
1	0	0	0	1	1	1
2	0	0	1	0	1	1
3	0	0	1	1	0	0
4	0	1	0	0	0	0
5	0	1	0	1	0	0
6	0	1	1	0	0	0
7	0	1	1	1	0	0
8	1	0	0	0	1	1
9	1	0	0	1	1	1
10	1	0	1	0	1	1
11	1	0	1	1	0	0
12	1	1	0	0	1	1
13	1	1	0	1	1	1
14	1	1	1	0	1	1
15	1	1	1	1	0	0

Как видно из таблицы, преобразования проведены правильно.

Варианты заданий

№	Выражения	№	Выражения
1	$\left(\overline{x_1} \vee x_2\right) \cdot \left(\overline{x_1 \vee x_1}\right) \cdot \left(x_3 \vee \overline{x_2}\right)$ $\left(x_1 \cdot x_2\right) \vee \left(x_1 \cdot x_3\right) \vee \left(x_3 \cdot x_3\right)$	11	$\left(\overline{x_1 \vee x_2}\right) \cdot \left(x_1 \vee x_1\right) \cdot \left(\overline{x_3} \vee x_2\right)$ $\left(x_1 \cdot x_2\right) \vee \left(\overline{x_1 \cdot x_3}\right) \vee \left(x_3 \cdot x_3\right)$
2	$\left(x_2 \vee x_3\right) \cdot \left(x_1 \vee x_4\right) \cdot \left(x_1 \vee x_2\right)$ $\left(\overline{x_4 \cdot x_1}\right) \vee \left(x_2 \cdot \overline{x_3}\right) \vee \left(x_4 \cdot x_1\right)$	12	$\left(\overline{x_2 \vee x_3}\right) \cdot \left(x_1 \vee x_4\right) \cdot \left(\overline{x_1} \vee \overline{x_2}\right)$ $\left(x_4 \cdot x_1\right) \vee \left(x_2 \cdot x_3\right) \vee \left(x_4 \cdot x_1\right)$
3	$\left(\overline{x_2} \vee x_4\right) \cdot \left(x_1 \vee x_3\right) \cdot \left(\overline{x_1 \vee x_2}\right)$ $\left(x_1 \cdot x_1\right) \vee \left(x_3 \cdot x_1\right) \vee \left(x_4 \cdot x_2\right)$	13	$\left(x_2 \vee x_4\right) \cdot \left(x_1 \vee x_3\right) \cdot \left(x_1 \vee x_2\right)$ $\left(\overline{x_1} \cdot x_1\right) \vee \left(\overline{x_3} \cdot \overline{x_1}\right) \vee \left(x_4 \cdot x_2\right)$
4	$\left(x_2 \vee \overline{x_3}\right) \cdot \left(x_1 \vee x_4\right) \cdot \left(\overline{x_1 \vee x_2}\right)$ $\left(x_4 \cdot x_2\right) \vee \left(x_3 \cdot x_2\right) \vee \left(x_4 \cdot x_1\right)$	14	$\left(\overline{x_2} \vee \overline{x_3}\right) \cdot \left(\overline{x_1 \vee x_4}\right) \cdot \left(x_1 \vee x_2\right)$ $\left(\overline{x_4 \cdot x_2}\right) \vee \left(x_3 \cdot x_2\right) \vee \left(x_4 \cdot \overline{x_1}\right)$
5	$\left(x_4 \vee x_1\right) \cdot \left(x_1 \vee x_4\right) \cdot \left(x_1 \vee x_2\right)$ $\left(x_4 \cdot \overline{x_4}\right) \vee \left(\overline{x_1 \cdot x_2}\right) \vee \left(x_4 \cdot \overline{x_1}\right)$	15	$\left(x_4 \vee x_1\right) \cdot \left(\overline{x_1} \vee \overline{x_4}\right) \cdot \left(x_1 \vee x_2\right)$ $\left(x_4 \cdot x_4\right) \vee \left(x_1 \cdot x_2\right) \vee \left(\overline{x_4 \cdot x_1}\right)$
6	$\left(x_2 \vee \overline{x_2}\right) \cdot \left(x_2 \vee x_3\right) \cdot \left(\overline{x_4 \vee x_2}\right)$ $\left(x_4 \cdot \overline{x_3}\right) \vee \left(x_2 \cdot x_2\right) \vee \left(\overline{x_3} \cdot \overline{x_1}\right)$	16	$\left(x_2 \vee x_2\right) \cdot \left(x_2 \vee x_3\right) \cdot \left(x_4 \vee x_2\right)$ $\left(x_4 \cdot \overline{x_3}\right) \vee \left(x_2 \cdot x_2\right) \vee \left(\overline{x_3 \cdot x_1}\right)$
7	$\left(x_2 \vee \overline{x_4}\right) \cdot \left(\overline{x_1 \vee x_3}\right) \cdot \left(x_1 \vee x_1\right)$ $\left(x_3 \cdot \overline{x_3}\right) \vee \left(\overline{x_2 \cdot x_1}\right) \vee \left(x_3 \cdot x_2\right)$	17	$\left(\overline{x_2 \vee x_4}\right) \cdot \left(x_1 \vee x_3\right) \cdot \left(x_1 \vee x_1\right)$ $\left(x_3 \cdot x_3\right) \vee \left(\overline{x_2} \cdot x_1\right) \vee \left(\overline{x_3 \cdot x_2}\right)$
8	$\left(x_4 \vee \overline{x_4}\right) \cdot \left(x_2 \vee x_3\right) \cdot \left(\overline{x_1} \vee x_3\right)$ $\left(\overline{x_3} \cdot \overline{x_4}\right) \vee \left(x_1 \cdot x_1\right) \vee \left(x_2 \cdot x_3\right)$	18	$\left(x_4 \vee x_4\right) \cdot \left(x_2 \vee x_3\right) \cdot \left(x_1 \vee x_3\right)$ $\left(\overline{x_3} \cdot x_4\right) \vee \left(x_1 \cdot x_1\right) \vee \left(\overline{x_2} \cdot x_3\right)$
9	$\left(x_1 \vee x_4\right) \cdot \left(x_1 \vee x_3\right) \cdot \left(x_1 \vee x_1\right)$ $\left(x_3 \cdot x_3\right) \vee \left(\overline{x_1 \cdot x_4}\right) \vee \left(x_3 \cdot \overline{x_2}\right)$	19	$\left(\overline{x_1 \vee x_4}\right) \cdot \left(\overline{x_1 \vee x_3}\right) \cdot \left(x_1 \vee x_1\right)$ $\left(x_3 \cdot x_3\right) \vee \left(\overline{x_1} \cdot x_4\right) \vee \left(x_3 \cdot \overline{x_2}\right)$
10	$\left(x_2 \vee \overline{x_4}\right) \cdot \left(x_2 \vee x_2\right) \cdot \left(\overline{x_3 \vee x_1}\right)$ $\left(\overline{x_3 \cdot x_1}\right) \vee \left(x_1 \cdot \overline{x_3}\right) \vee \left(x_4 \cdot x_4\right)$	20	$\left(x_2 \vee x_4\right) \cdot \left(x_2 \vee x_2\right) \cdot \left(\overline{x_3} \vee x_1\right)$ $\left(x_3 \cdot x_1\right) \vee \left(x_1 \cdot x_3\right) \vee \left(x_4 \cdot x_4\right)$

Порядок выполнения работы

1. Выполнить преобразования логических функций согласно своему варианту задания, применив к выражению двойное отрицание и обобщённый закон де Моргана. При этом учитывать законы дополнения и тавтологии.
2. Проверить правильность преобразования, составив таблицы истинности выражений.
3. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:
   - вариант задания на лабораторную работу;
   - записи преобразования выражений,
   - таблицы истинности исходных и преобразованных выражений.

Контрольные вопросы

1. Какие существуют отношения между элементами алгебры логики, как они обозначаются?
2. Напишите формулы логического сложения, умножения и инверсии.
3. В чём заключаются закон единичных элементов, двойное отрицание и дополнение?
4. Сформулируйте закон де Моргана.

5. Напишите формулы для закона тавтологии, а также переместительного и сочетательного законов.
6. Какова формулировка распределительного закона, а также склеивания и поглощения?
7. Выполните преобразование логической функции, данной преподавателем, при помощи обобщённого закона де Моргана.

Лабораторная работа №4. Переключательные функции

Цель работы: научиться задавать логические функции в канонических формах.

Теоретическая часть

Переключательной (двоичной) функцией называется двоичная переменная, значение которой зависит от значений других двоичных переменных, называемых аргументами этой функции:

$$p = p(x_1, x_2, \mathbf{K}\, x_n).$$

Двоичная функция считается заданной, если каждому набору значений аргументов поставлено в соответствие определённое значение этой функции. Переключательные функции называются разными, если их значения отличаются, по крайней мере, для одного набора значений аргументов. Если имеется n аргументов логической функции, то полное число наборов значений N этой функции составит:

$$N = 2^n.$$

Поскольку на каждом наборе аргументов логические функции могут принимать два значения, то общее число F функций n аргументов составит:

$$F = 2^{2^N}.$$

Простейшим способом задания переключательной функции является составление таблицы истинности. Например, нужно задать все функции одной переменной.

Число наборов для функции с двумя аргументами составляет:

$$N = 2^n = 2^1 = 2.$$

Число всех возможных функций для одной переменной:

$$F = 2^N = 2^2 = 4.$$

x	f_1	f_2	f_3	f_4
0	0	0	1	1
1	0	1	0	1

Эти функции можно задать в виде логических формул:

$f_1 = 0$ – константа ноль.

$f_4 = 1$ – константа единица.

$f_3 = \overline{x}$ – функция инверсии.

$f_2 = x$ – функция тождественности.

Рассмотрим формулы и таблицы истинности основных отношений в логических функциях.

Дизъюнкция:

$p = x_1 \vee x_2$ (ИЛИ).

x_1	x_2	p
0	0	0
0	1	1
1	0	1
1	1	1

Конъюнкция:

$p = x_1 \cdot x_2$ (И).

x_1	x_2	p
0	0	0
0	1	0
1	0	0
1	1	1

Исключающее или (сложение по модулю 2):

$p = \overline{x_1} \cdot x_2 \vee x_1 \cdot \overline{x_2} = x_1 \oplus x_2$.

x_1	x_2	p
0	0	0
0	1	1
1	0	1
1	1	0

Функция Шеффера (И-НЕ):

$p = \overline{x_1 \cdot x_2}$

x_1	x_2	p
0	0	1
0	1	1
1	0	1
1	1	0

Функция Пирса (ИЛИ-НЕ):

$p = \overline{x_1 \vee x_2}$

x_1	x_2	p
0	0	1
0	1	0
1	0	0
1	1	0

Эти функции широко применяются при синтезе цифровых автоматов.

Рассмотрим задание произвольной логической функции при помощи таблицы истинности и минимизацию этой функции.

В таблице истинности записываются все возможные наборы аргументов логической функции, а затем для каждого набора значений аргументов проставляются значения функции. Рассмотрим функцию трёх аргументов. Число наборов для неё составит: $N = 2^n = 2^3 = 8$

№	x_1	x_2	x_3	p
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

При проведении минимизации табличную функцию представляют в одной из двух канонических форм: совершенной нормальной дизъюнктивной форме (СДНФ) и совершенной нормальной конъюнктивной форме (СКНФ). Они образуются при помощи двух вспомогательных функций: минтерма и макстерма.

Минтерм – это функция, принимающая значение 1 только на одном наборе аргументов. Минтерм выражается через конъюнкцию аргументов, причём аргумент берётся без знака отрицания, если он равен 1, и со знаком отрицания, если он равен 0.

Макстерм – это функция, принимающая значение 0 только на одном наборе аргументов. Макстерм выражается через дизъюнкцию аргументов, причём аргумент берётся со знаком отрицания, если он равен 1, и без знака отрицания, если он равен 0.

Например, для наборов №1 и №3 получаются минтермы:

$\overline{x_1} \cdot \overline{x_2} \cdot x_3$ и $\overline{x_1} \cdot x_2 \cdot x_3$.

Для наборов №0 и №2 получаются макстермы:

$\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}$ и $x_1 \cdot \overline{x_2} \cdot x_3$.

Совершенная нормальная дизъюнктивная форма (СДНФ) – это дизъюнкция минтермов, равных 1 на тех же наборах, что и заданная логическая функция.

Совершенная нормальная конъюнктивная форма (СКНФ) – это конъюнкция макстермов, равных 0 на тех же наборах, что и заданная логическая функция.

Также СДНФ называют записью по единицам, а СКНФ – записью по нулям.

Функция p равна единице на наборах 1, 3, 5, 6. Запишем СДНФ:

$$p = \overline{x_1} \cdot \overline{x_2} \cdot x_3 \vee \overline{x_1} \cdot x_2 \cdot x_3 \vee x_1 \cdot \overline{x_2} \cdot x_3 \vee x_1 \cdot x_2 \cdot \overline{x_3}.$$

Функция p равна нулю на наборах 0, 2, 4, 7. Запишем СКНФ:

$$p = (x_1 \vee x_2 \vee x_3) \cdot (x_1 \vee \overline{x_2} \vee x_3) \cdot (\overline{x_1} \vee x_2 \vee x_3) \cdot (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}).$$

Составляющие этих функций, например $\overline{x_1} \cdot \overline{x_2} \cdot x_3$ или $(x_1 \vee x_2 \vee x_3)$, называют простыми импликантами.

Часто используют такой метод записи:

СДНФ: $p(x_1, x_2, x_3) = 1(1, 3, 5, 6)$.

СКНФ: $p(x_1, x_2, x_3) = 0(0, 2, 4, 7)$.

Варианты заданий

1. Получить канонические формы функции трёх переменных (x_1, x_2, x_3):

№	Единичные наборы	№	Единичные наборы
1	0, 1, 4, 6, 7 4, 5, 6, 7	11	0, 1, 4, 5, 0, 6, 7
2	0, 1, 2, 7 0, 1, 2, 3, 4	12	0, 1, 2, 7 0, 1, 4,
3	0, 2, 4, 6 0, 2, 5, 6, 7	13	3, 4, 5, 6, 7 0, 1, 2
4	0, 1, 3, 5, 7 0, 1, 3, 6	14	0, 4, 5, 6, 7 0, 1, 2, 3, 4, 6, 7
5	0, 1, 2, 4, 6, 5, 6, 7	15	0, 1, 2, 3, 4, 5 1, 2, 5, 6, 7
6	1, 3, 5, 6, 7 0, 6, 7	16	0, 1, 2, 3, 4 3, 4, 7
7	0, 2, 3, 4, 5, 7 0, 1, 2, 7	17	0, 1, 2, 3 0, 5, 6, 7
8	0, 1, 3, 4, 6, 7 0, 1, 2	18	0, 1, 2, 3, 4 0, 1, 2, 6, 7
9	0, 1, 2, 5, 6, 7 0, 1, 2, 7	19	2, 3, 4, 5, 6 0, 4, 6, 7
10	2, 3, 4, 5, 6, 7 0, 1, 7	20	0, 1, 2, 7 0, 1, 2, 3, 7

2. Получить канонические формы функции четырёх переменных (x_1, x_2, x_3, x_4):

№	Единичные наборы	№	Единичные наборы
1	0, 1, 4, 6, 7, 8, 10 4, 5, 6, 7, 9, 11	11	0, 1, 4, 5, , 10, 12 0, 6, 7, 9, 11
2	0, 1, 2, 7, 10, 12 0, 1, 2, 3, 14, 15,	12	0, 1, 2, 7, 10, 12 0, 1, 4, 13, 15
3	0, 2, 4, 6, 7, 13, 15 0, 2, 5, 6, 7, 10, 12	13	3, 4, 5, 6, 7 0, 1, 2, 9, 11
4	0, 1, 3, 5, 7, 14, 15 0, 1, 3, 6, 8, 9	14	0, 4, 5, 6, 7, 8, 10 0, 1, 2, 3, 4, 6, 7
5	0, 1, 2, 4, 6, 9, 11 5, 6, 7, 8, 10	15	0, 1, 2, 3, 4, 5 1, 2, 5, 6, 7, 10, 12
6	1, 3, 5, 6, 7 0, 6, 7, 9, 11	16	0, 1, 2, 3, 4 3, 4, 7, 13, 15
7	0, 2, 3, 4, 5, 7 0, 1, 2, 7, 10, 12	17	0, 1, 2, 3, 8, 10 0, 5, 6, 7, 13, 15
8	0, 1, 3, 4, 6, 7 0, 1, 2, 8, 10	18	0, 1, 2, 3, 4, 10, 12 0, 1, 2, 6, 7, 9, 11
9	0, 1, 2, 5, 6, 7 0, 1, 2, 7, 13, 15	19	2, 3, 4, 5, 6, 9, 12 0, 4, 6, 7, 10, 12
10	2, 3, 4, 5, 6, 7 0, 1, 7, 8, 10	20	0, 1, 2, 7, 10, 12 0, 1, 2, 3, 7, 10, 12

Порядок выполнения работы

1. Записать канонические формы логических функций из таблиц 1 и 2 согласно своему варианту задания.
2. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:
   - вариант задания на лабораторную работу;
   - формулы СКНФ и СДНФ для трёх и четырёх переменных.

Контрольные вопросы

1. Что называется переключательной (двоичной) функцией, какая функция считается заданной, чем они отличаются?
2. Как определить число наборов для логической функции и число логических функций?
3. Опишите все возможные функции одной переменной в табличном и символьном виде.
4. Запишите в символьном и табличном виде функции дизъюнкции, конъюнкции и сложения по модулю 2.
5. Запишите в символьном и табличном виде функции Шеффера и Пирса.
6. Составьте таблицу истинности для функции от трёх аргументов.
7. Какие существуют канонические формы записи логической функции, что такое минтерм и макстерм?
8. Что такое СДНФ и СКНФ?
9. Запишите СДНФ и СКНФ логической функции, данной преподавателем.

Лабораторная работа №5. Минимизация логических функций

Цель работы: изучить применение карт Карно при поиске минимальных форм логических функций.

Теоретическая часть

Минимизацией логической функции называется представление функции в аналитической форме с минимальным количеством переменных и логических операций.

Одним из широко используемых методов минимизации является применение карт Карно. Этот способ минимизации функций был предложен в 1952 Эдвардом В. Вейчем и усовершенствован в 1953 Морисом Карно. В карту Карно двоичные переменные передаются из таблицы истинности и упорядочиваются с помощью кода Грея, в котором каждое следующее число отличается от предыдущего только одним разрядом. Основным методом минимизации логических функций, представленных в виде СДНФ или СКНФ, является операция попарного неполного склеивания и элементарного поглощения. Операция попарного склеивания осуществляется между двумя импликантами, содержащими одинаковые переменные, вхождения которых (прямые и инверсные) совпадают для всех переменных, кроме одной. В этом случае все переменные, кроме одной, можно вынести за скобки, а оставшиеся в скобках прямое и инверсное вхождение одной переменной подвергнуть склейке.

Исходной информацией для работы с картой Карно является таблица истинности минимизируемой функции. Фактически, карта Карно – это представление таблицы истинности в двумерном виде. Рассмотрим пример для функции трёх переменных.

№	x_1	x_2	x_3	p
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Преобразуем таблицу истинности в карту Карно.

		$x_1\ x_2$			
		00	01	11	10
x_3	0	1	0	1	1
	1	0	0	0	1

Склейку клеток карты Карно можно осуществлять по единицам, если необходимо получить дизъюнктивную форму (ДНФ), или по нулям (если требуется КНФ). Склеивать можно только прямоугольные области с числом единиц (нулей) 2^n, где n – целое число. Кроме того, область, которая подвергается склейке, должна содержать только единицы (нули). Крайние клетки каждой горизонтали и каждой вертикали считаются смежными и могут объединяться. Все единицы (нули) должны попасть в какую-либо область. Одна ячейка карты может входить сразу в несколько областей (это следует из закона тавтологии).

Проведём склейку ячеек по единицам для получения ДНФ. Объединяем соседние ячейки $x_1\ x_2$ = 00, x_3 = 0 и $x_1\ x_2$ = 10, x_3 = 0. В этом случае переменные x_2 и x_3 не меняют своего значения, а x_1 принимает значения 0 и 1. При этом x_2 = 0 и x_3 = 0. Запишем импликант: $\overline{x_2}\ \overline{x_3}$. При склеивании соседних ячеек $x_1\ x_2$ = 11, x_3 = 0 и $x_1\ x_2$ = 10, x_3 = 0, получаем импликант $x_1\overline{x_3}$. Аналогично, при объединении $x_1\ x_2$ = 10, x_3 = 0 и $x_1\ x_2$ = 10, x_3 = 1 получим $x_1\overline{x_2}$. Таким образом, минимальная дизъюнктивная форма функции:

$$p = \overline{x_2}\ \overline{x_3} \vee x_1\overline{x_3} \vee x_1\overline{x_2}\ .$$

Минимальная форма логической функции, получаемая при помощи карты Карно, не является единственной, в отличие от СКНФ и СДНФ. Возможно несколько эквивалентных друг другу ДНФ (КНФ), которые соответствуют разным способам покрытия карты Карно прямоугольными областями.

Для получения конъюнктивной формы произведём объединение ячеек по нулям. При этом рассматриваем клетки с нулями, неменяющиеся переменные в пределах одной области объединяем в дизъюнкции (инверсии проставляем над единичными переменными), а дизъюнкции областей объединяем в конъюнкцию.:

$x_1\ x_2$ = 01, x_3 = 0 и $x_1\ x_2$ = 01, x_3 = 1,

$x_1\ x_2$ = 00, x_3 = 1 и $x_1\ x_2$ = 01, x_3 = 1,

$x_1\ x_2$ = 01, x_3 = 1 и $x_1\ x_2$ = 11, x_3 = 1.

Получаем:

$$p = \left(x_1 \vee \overline{x_2}\right)\cdot\left(x_1 \vee \overline{x_3}\right)\cdot\left(\overline{x_2} \vee \overline{x_3}\right).$$

Рассмотрим пример для четырёх переменных.

		$x_1\ x_2$			
		00	01	11	10
$x_3\ x_4$	00	0	1	1	0
	01	0	0	0	0
	11	1	0	0	1
	10	0	1	1	0

Для получения минимальной ДНФ объединим ячейки по единицам.

$x_1\ x_2\ = 01,\ x_1\ x_2\ = 11,\ x_3\ x_4\ = 00,\ x_3\ x_4\ = 10.$

$x_1\ x_2\ = 00,\ x_1\ x_2\ = 10,\ x_3\ x_4\ = 11.$

В результате:

$p = x_2\overline{x_4} \vee \overline{x_2}x_3x_4\,.$

Получим минимальную КНФ, для чего произведём объединение ячеек по нулям.

$x_1\ x_2\ = 01,\ x_1\ x_2\ = 11,\ x_3\ x_4\ = 01,\ x_3\ x_4\ = 11.$

$x_1\ x_2\ = 00,\ x_1\ x_2\ = 10,\ x_3\ x_4\ = 00,\ x_3\ x_4\ = 01.$

$x_1\ x_2\ = 00,\ x_1\ x_2\ = 10,\ x_3\ x_4\ = 10.$

В результате:

$p = \left(\overline{x_2} \vee \overline{x_4}\right)\cdot\left(x_2 \vee x_3\right)\cdot\left(x_2 \vee \overline{x_3} \vee x_4\right).$

Для карт Карно с числом переменных более четырёх могут получаться более сложные области.

Варианты заданий

В данной работе варианты заданий совпадают с заданиями в работе №4.

Порядок выполнения работы

1. Используя таблицу истинности, при помощи карты Карно получить дизъюнктивные и конъюнктивные минимальные формы логических функций согласно своему варианту задания.
2. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:
   - вариант задания на лабораторную работу;
   - карты Карно для получения минимальных форм своего варианта задания;
   - дизъюнктивные и конъюнктивные минимальные формы логических функций.

Контрольные вопросы

1. Что понимают под минимизацией логической функции?
2. Какая существует связь между таблицей истинности и картой Карно?
3. Как получить дизъюнктивную минимальную форму логической функции?
4. Как получить конъюнктивную минимальную форму логической функции?
5. Запишите минимальную форму логической функции, данной преподавателем.

Лабораторная работа №6. Реализация логических функций в различных базисах

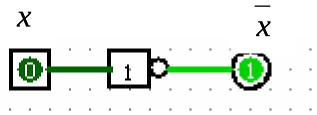
Цель работы: изучить способы реализаций логических функций в основном базисе, а также в базисах «И-НЕ» и «ИЛИ-НЕ».

Теоретическая часть

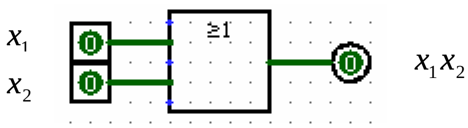
Для изображения логических схем применяются условные изображения логических элементов.

Входы у всех элементов располагаются слева, выходы – справа. В левом верхнем углу прямоугольника расположен символ операции, выполняемой элементом. Входы и выходы могут быть прямыми и инверсными. Если вход инверсный. То элемент выполняет свою логическую операцию над инверсным значением переменной, переданной на этот вход. Если выход элемента инверсный, то на выход элемента подаётся инверсное значение выполняемой элементом логической операции.

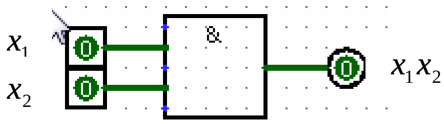
Инвертор: НЕ



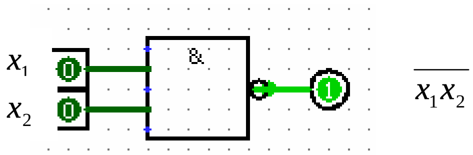
Дизъюнкция: ИЛИ



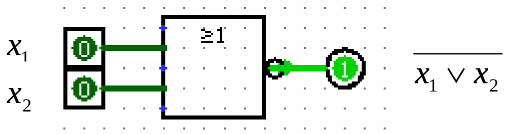
Конъюнкция: И



Функция Шеффера И-НЕ:



Функция Пирса: ИЛИ-НЕ



Все эти элементы имеют статические входы, то есть реагируют на уровень входного сигнала (с 0 на 1). У более сложных логических элементов могут использоваться динамические входы, которые реагируют не на уровень входного сигнала, а на переход входного сигнала с одного уровня на другой. Эти элементы выполняют свою функцию только в те моменты, когда входной сигнал меняет своё значение.

При помощи условных обозначений изображается реализация функций в различных базисах.

Функционально полная система – это такой набор логических функций, с помощью которых можно выразить любую логическую функцию. Такой набор функций также называют базисом.

Система логических элементов, реализующих функционально полную систему логических функций, называют функционально полной системой логических элементов.

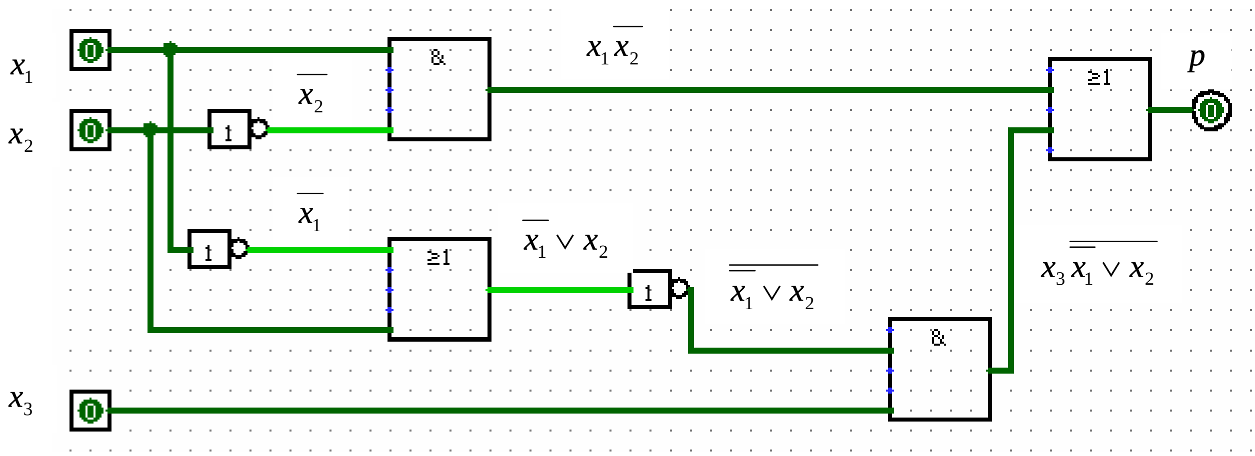
Основной базис включает в себя операции НЕ, И, ИЛИ.

Базис Шеффера (И-НЕ) содержит операцию И-НЕ.

Базис Пирса (ИЛИ-НЕ) содержит операцию ИЛИ-НЕ.

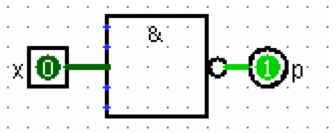
Рассмотрим пример реализации логической функции в основном базисе.

$p = x_1\overline{x_2} \vee x_3\overline{\overline{x_1} \vee x_2}$.



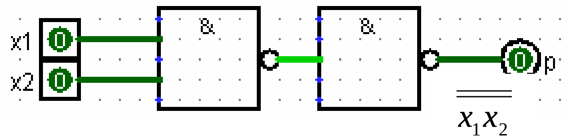
Для реализации функций в базисе И-НЕ соответствующие элементы получают, используя закон де Моргана.

Элемент НЕ: $\overline{x}$



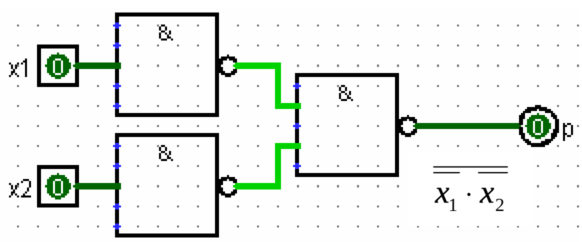
x	p
0	1
1	0

Элемент И: $x_1x_2 = \overline{\overline{x_1x_2}}$.



x1	x2	p
0	0	0
0	1	0
1	0	0
1	1	1

Элемент ИЛИ: $x_1 \vee x_2 = \overline{\overline{x_1 \vee x_2}} = \overline{\overline{x_1} \cdot \overline{x_2}}$.

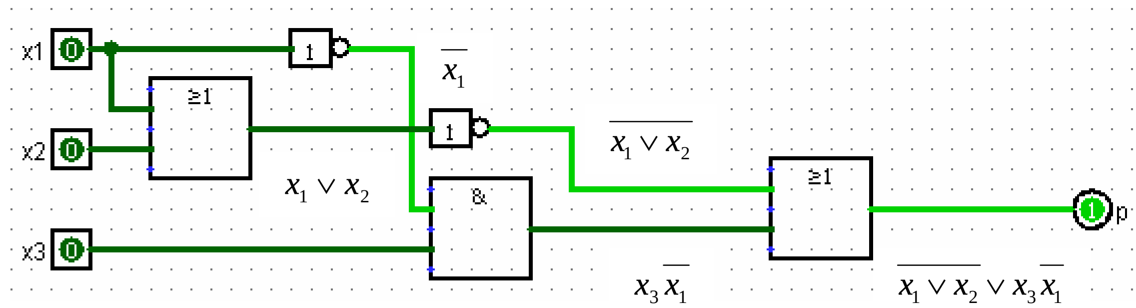


x1	x2	p
0	0	0
0	1	1
1	0	1
1	1	1

Рассмотрим пример.

$p = \overline{x_1 \vee x_2} \vee x_3\overline{x_1}$.

В основном базисе:



x1	x2	x3	p
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Для преобразования выражения $p = \overline{x_1 \vee x_2} \vee x_3\overline{x_1}$ в базис И-НЕ произведём замены:

$z_1 = \overline{x_1 \vee x_2}$,

$z_2 = x_3\overline{x_1}$.

Таким образом,

$p = z_1 \vee z_2$.

Проведём преобразования:

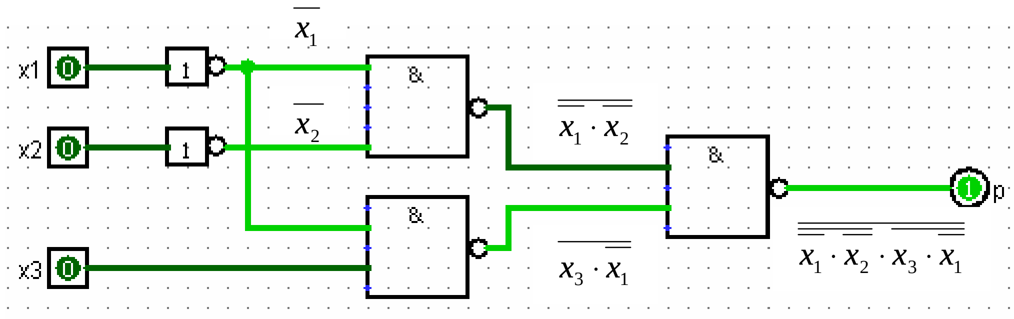
$z_1 = \overline{x_1 \vee x_2} = \overline{\overline{\overline{x_1} \cdot \overline{x_2}}}$,

$z_2 = x_3\overline{x_1} = \overline{\overline{x_3\overline{x_1}}}$,

$p = z_1 \vee z_2 = \overline{\overline{z_1} \cdot \overline{z_1}}$.

Произведём подстановку преобразованных z_1 и z_2, а также сократим чётное число инверсий:

$p = z_1 \vee z_2 = \overline{\overline{\overline{\overline{\overline{x_1} \cdot \overline{x_2}}}} \cdot \overline{\overline{\overline{x_3\overline{x_1}}}}} = \overline{\overline{\overline{x_1} \cdot \overline{x_2}} \cdot \overline{x_3\overline{x_1}}}$.

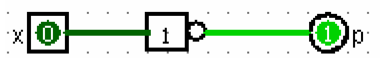


x1	x2	x3	p
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Эта функция содержит только операции И-НЕ.

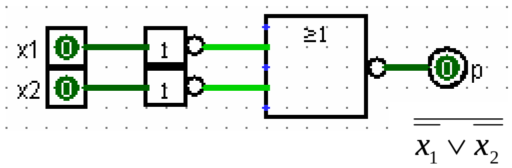
Рассмотрим ту же функцию в базисе ИЛИ-НЕ. Для реализации функций в этом базисе соответствующие элементы получают, используя закон де Моргана.

Элемент НЕ: $\overline{x}$



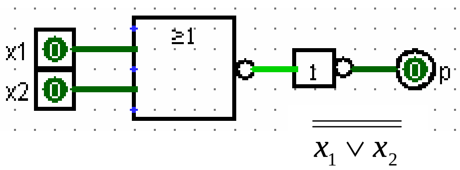
x	p
0	1
1	0

Элемент И: $x_1 x_2 = \overline{\overline{x_1 x_2}} = \overline{\overline{x_1} \vee \overline{x_2}}$.



x1	x2	p
0	0	0
0	1	0
1	0	0
1	1	1

Элемент ИЛИ: $x_1 \vee x_2 = \overline{\overline{x_1 \vee x_2}}$.



x1	x2	p
0	0	0
0	1	1
1	0	1
1	1	1

Для преобразования выражения $p = \overline{x_1 \vee x_2} \vee x_3\overline{x_1}$ в базис ИЛИ-НЕ произведём замены:

$z_1 = \overline{x_1 \vee x_2}$,

$z_2 = x_3\overline{x_1}$.

Таким образом,

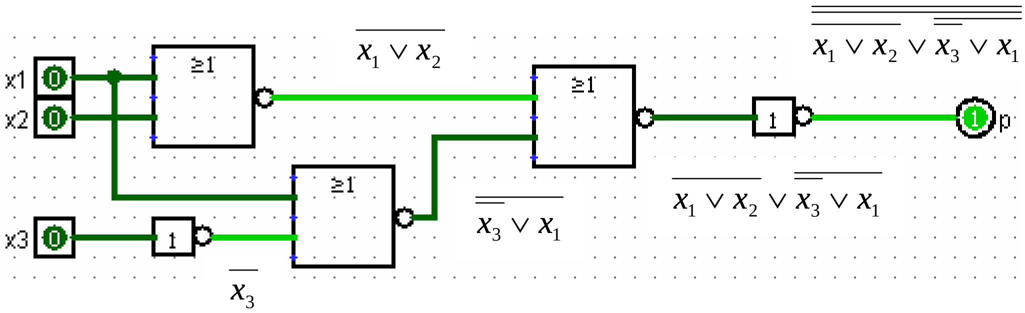
$p = z_1 \vee z_2$.

Функция z_1 уже представлена в нужном базисе. Преобразуем z_2:

$z_2 = x_3\overline{x_1} = \overline{\overline{x_3\overline{x_1}}} = \overline{\overline{x_3} \vee \overline{\overline{x_1}}} = \overline{\overline{x_3} \vee x_1}$.

В результате:

$p = \overline{x_1 \vee x_2} \vee \overline{\overline{x_3} \vee x_1} = \overline{\overline{\overline{x_1 \vee x_2} \vee \overline{\overline{x_3} \vee x_1}}}$.



x1	x2	x3	p
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Эта функция содержит только операции ИЛИ-НЕ.

Если логическая функция задана таблицей истинности, то её можно реализовать в любом базисе. Запись СДНФ или СКНФ является реализацией функции в основном базисе. Для того, чтобы получить ту же функцию в базисе И-НЕ, надо записать СДНФ этой функции, а затем взять от всего выражения двойное отрицание и применить обобщённый закон де Моргана. Если же нужно получить реализацию функции в базисе ИЛИ-НЕ, то надо записать СКНФ, а затем взять двойное отрицание и применить закон де Моргана.

Рассмотрим пример. Пусть логическая функция задана таблицей истинности:

№	x_1	x_2	x_3	p
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Для её реализации в базисе И-НЕ запишем СДНФ:

$$p = \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \vee \bar{x}_1 \cdot x_2 \cdot x_3 \vee x_1 \cdot \bar{x}_2 \cdot x_3 \vee x_1 \cdot x_2 \cdot \bar{x}_3 .$$

Применим двойное отрицание и закон де Моргана:

$$p = \overline{\overline{\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \vee \bar{x}_1 \cdot x_2 \cdot x_3 \vee x_1 \cdot \bar{x}_2 \cdot x_3 \vee x_1 \cdot x_2 \cdot \bar{x}_3}} = \overline{\overline{\bar{x}_1 \cdot \bar{x}_2 \cdot x_3} \cdot \overline{\bar{x}_1 \cdot x_2 \cdot x_3} \cdot \overline{x_1 \cdot \bar{x}_2 \cdot x_3} \cdot \overline{x_1 \cdot x_2 \cdot \bar{x}_3}} .$$

Для реализации этой функции в базисе ИЛИ-НЕ запишем СКНФ:

$$p = \left(x_1 \vee x_2 \vee x_3\right) \cdot \left(x_1 \vee \bar{x}_2 \vee x_3\right) \cdot \left(\bar{x}_1 \vee x_2 \vee x_3\right) \cdot \left(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3\right).$$

Применим двойное отрицание и закон де Моргана:

$$p = \overline{\overline{\left(x_1 \vee x_2 \vee x_3\right) \cdot \left(x_1 \vee \bar{x}_2 \vee x_3\right) \cdot \left(\bar{x}_1 \vee x_2 \vee x_3\right) \cdot \left(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3\right)}} =$$

$$= \overline{\overline{\left(x_1 \vee x_2 \vee x_3\right)} \vee \overline{\left(x_1 \vee \bar{x}_2 \vee x_3\right)} \vee \overline{\left(\bar{x}_1 \vee x_2 \vee x_3\right)} \vee \overline{\left(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3\right)}}.$$

Варианты заданий

1. Функции трёх переменных $\left(x_1, x_2, x_3\right)$:

№	Единичные наборы	№	Единичные наборы
1	0, 1, 4, 6, 7	11	0, 1, 4, 5,
2	0, 1, 2, 3, 4	12	0, 1, 4,
3	0, 2, 5, 6, 7	13	3, 4, 5, 6, 7
4	0, 1, 3, 5, 7	14	0, 1, 2, 3, 4, 6, 7
5	5, 6, 7	15	0, 1, 2, 3, 4, 5
6	1, 3, 5, 6, 7	16	3, 4, 7
7	0, 2, 3, 4, 5, 7	17	0, 5, 6, 7
8	0, 1, 3, 4, 6, 7	18	0, 1, 2, 3, 4
9	0, 1, 2, 5, 6, 7	19	2, 3, 4, 5, 6
10	2, 3, 4, 5, 6, 7	20	0, 1, 2, 7

2. Функции четырёх переменных $\left(x_1, x_2, x_3, x_4\right)$:

№	Единичные наборы	№	Единичные наборы
1	0, 1, 4, 6, 7, 8, 10	11	0, 1, 4, 5, , 10, 12
2	0, 1, 2, 3, 14, 15,	12	0, 1, 2, 7, 10, 12
3	0, 2, 5, 6, 7, 10, 12	13	3, 4, 5, 6, 7
4	0, 1, 3, 5, 7, 14, 15	14	0, 4, 5, 6, 7, 8, 10
5	0, 1, 2, 4, 6, 9, 11	15	1, 2, 5, 6, 7, 10, 12
6	0, 6, 7, 9, 11	16	0, 1, 2, 3, 4
7	0, 1, 2, 7, 10, 12	17	0, 5, 6, 7, 13, 15
8	0, 1, 2, 8, 10	18	0, 1, 2, 3, 4, 10, 12
9	0, 1, 2, 7, 13, 15	19	2, 3, 4, 5, 6, 9, 12
10	0, 1, 7, 8, 10	20	0, 1, 2, 3, 7, 10, 12

Порядок выполнения работы

1. Используя таблицу истинности, при помощи карты Карно получить дизъюнктивные и конъюнктивные минимальные формы логических функций согласно своему варианту задания.
2. Реализовать полученные функции в основном базисе.

3. Преобразовать полученные минимальные формы в базисы И-НЕ (для дизъюнктивной формы) и ИЛИ-НЕ (для конъюнктивной формы).
4. Запустить программу Logisim и изучить «Пособие начинающего» в меню «Справка».
5. Реализовать полученные функции в основном базисе, а также в базисах И-НЕ и ИЛИ-НЕ, используя Logisim.
6. Построить таблицу истинности: меню Проект – Анализировать схему – вкладка Таблица.
7. Составить таблицы истинности функций, данные в варианте задания, с полученными в Logisim. Проверить правильность преобразования при помощи сравнения этих таблиц. В случае несовпадения результатов вернуться к п.1.
8. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:
   - вариант задания на лабораторную работу;
   - карты Карно для получения минимальных форм своего варианта задания;
   - дизъюнктивные и конъюнктивные минимальные формы логических функций.
   - схемы полученных минимальных форм в основном базисе, а также в базисах И-НЕ и ИЛИ-НЕ;
   - таблицы истинности функций.

Контрольные вопросы

1. Каковы общие принципы изображения логических элементов?
2. Как изображаются элементы инверсии, дизъюнкции, конъюнкции, Шеффера и Пирса?
3. Какие типы входов существуют у логических элементов, как они изображаются?
4. Что такое базис, какие базисы существуют, какие операции они содержат?
5. Как осуществляется переход между базисами?
6. Запишите в базисе И-НЕ (ИЛИ-НЕ) логическую функцию, данную преподавателем.

Лабораторная работа №7. Формы представления цифровых автоматов с памятью (автоматов Мили)

Цель работы: изучить различные способы представления цифровых автоматов с памятью: табличный, при помощи графов и матричный.

Теоретическая часть

Цифровой автомат (ЦА) – это формальная модель любого информационного устройства дискретного действия. Модель предназначена для представления устройства на логическом уровне, т.е. в ней не рассматриваются физические характеристики процессов, происходящих при работе реального устройства.

Любой дискретный узел, входящий в состав вычислительной техники, представляет собой цифровой автомат, осуществляющий преобразование исходных последовательностей цифровых кодов в результирующие последовательности по определённому алгоритму.

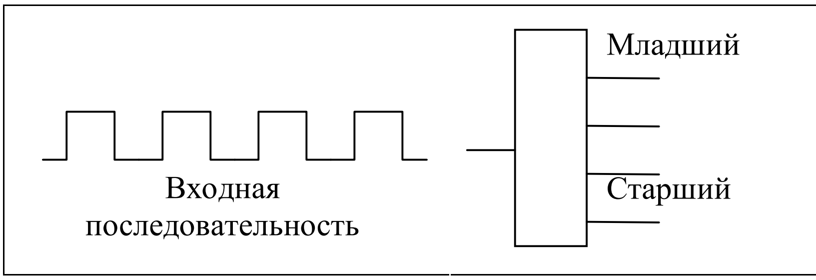
Каждый цифровой автомат характеризуется количеством состояний, в которых он может находиться в процессе работы. Обычно это количество конечно, поэтому цифровые автоматы такого типа называют конечными автоматами. Бесконечное число состояний для цифрового автомата означает его бесконечную сложность, так как иначе нельзя обеспечить отличие одного состояния от другого. Можно представить себе бесконечный ЦА, как автомат с неограниченно наращиваемой памятью состояний, и такой автомат представляет определенный теоретический интерес, но в данном курсе рассматриваются только конечные автоматы. Термины «конечный автомат» и «цифровой автомат» часто используются как синонимы.

Конечные автоматы бывают двух типов:

1. Комбинационные (автоматы Мура, автоматы без памяти);
2. Последовательные (автоматы Мили, автоматы с памятью).

У комбинационных автоматов каждой комбинации входных сигналов соответствует одна комбинация выходных сигналов. Такие автоматы называют также комбинационными схемами. В них отсутствуют элементы памяти.

В отличие от комбинационных, в последовательные автоматы обладают элементами памяти, которые хранят предысторию автомата. Поэтому выходные сигналы последовательного автомата зависят не только от того, какие входные сигналы были поданы на вход автомата, но и от того, какая последовательность сигналов была подана на вход до этого. То есть, в отличие от комбинационных, последовательные автоматы могут неоднозначно реагировать на одну и ту же последовательность входных сигналов. Примером последовательного автомата является двоичный четырёхразрядный счётчик.



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

Типичными автоматами без памяти являются цифровые комбинационные схемы: сумматоры, дешифраторы, схемы сравнения, мультиплексоры и т.п.

Методы синтеза и минимизации комбинационных автоматов были рассмотрены в предыдущих лабораторных работах. Последовательные автоматы являются более сложными устройствами, чем комбинационные, но в конечном счёте их синтез сводится к синтезу комбинационных схем, выбору элементов памяти и соединению их определённым образом.

Работа ЦА осуществляется в автоматном времени, определяемом числом периодов поступления входных сигналов. При анализе и синтезе последовательных автоматов предполагается, что входные и выходные сигналы появляются не непрерывно, а только в некоторые фиксированные моменты времени t. Эти моменты обычно обозначаются цифрами натурального ряда (0, 1, 2, …). При этом поведение автомата в промежутке между соседними моментами времени является несущественным и не рассматривается.

Существует два способа задания автоматного времени. По этому признаку выделяют два типа последовательных автоматов: асинхронные и синхронные.

В синхронных автоматное время задаётся при помощи источника тактирующих сигналов, который является внешним по отношению к схеме автомата. У асинхронных автоматов моменты автоматного времени будут являться моментами изменения сигнала на выходе. У синхронного автомата очередной символ входной последовательности будет воспринят только в том случае, если в тот же момент подан тактирующий сигнал. Поэтому моменты автоматного времени для синхронного автомата являются моментами подачи тактирующего сигнала.

Существуют различные модели последовательных автоматов. Одной из наиболее употребительных является модель Клини.

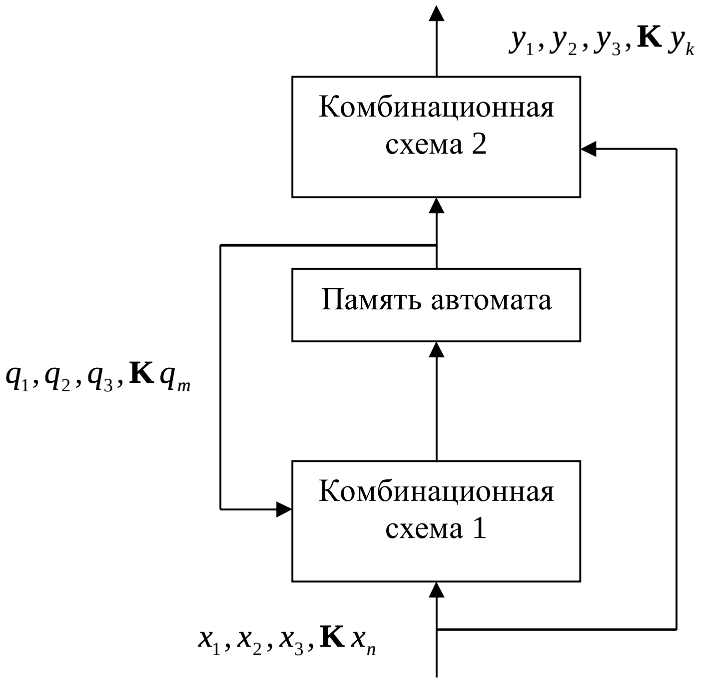


Рисунок 1 – Модель последовательного автомата

В этой модели $x_1, x_2, x_3, \mathbf{K}\, x_n$ входные сигналы (переменные), $q_1, q_2, q_3, \mathbf{K}\, q_m$ - внутренние сигналы (переменные), $y_1, y_2, y_3, \mathbf{K}\, y_k$ - выходные сигналы (переменные). Все возможные наборы входных, внутренних и выходных переменных образуют входной X, внутренний Q и выходной Y алфавиты автомата. Конечный автомат задаётся этими тремя алфавитами. Если числа n, m, k конечны, то автомат является абстрактным конечным. На практике применяются именно конечные автоматы. Они осуществляют преобразование входного алфавита в выходной. Это преобразование определяется характеристическими функциями переходов и выходов.

Функция переходов определяет состояние автомата в момент времени $t+1$ в зависимости от его входного и внутреннего состояний в момент времени t.

$q(t+1) = d\left(q(t), x(t)\right).$

Функция выходов определяет значение выходного сигнала $y(t)$. Эта функция может быть задана двумя способами.

Если выходное состояние автомата в момент времени t однозначно определятся его внутренним состоянием в тот же момент (автомат без памяти):

$y(t) = l\left(q(t)\right).$

Если выходное состояние автомата определяется внутренним и входным состоянием автомата в тот же момент времени (автомат с памятью):

$y(t) = l\left(q(t), x(t)\right).$

Кроме внутреннего и внешнего состояния, на выходной сигнал влияет также и начальное состояние q_0 автомата.

Существует несколько способов задания автоматов Мили (с памятью) и Мура (без памяти). В данной работе будут рассмотрены способы:

- при помощи таблиц переходов и выходов,
- с использованием графов,
- матрица переходов и выходов.
- табличная форма матрицы.

Все эти способы являются формами представления модели абстрактного конечного автомата Мили или Мура и однозначно связаны между собой.

Таблицы переходов и выходов автомата Мили содержит столько строк. Сколько входных состояний у автомата, и столько столбцов, сколько у него внутренних состояний. Таким образом, задаются все возможные пары входных x_i и внутренних q_j состояний в момент времени t. На пересечении i-й строки и j-го столбца записывается то внутреннее состояние q_{ij}, в которое автомат переходит в момент времени $t+1$, если в предыдущий момент времени t он находился во входном состоянии x_i и во внутреннем состоянии q_j.

Рассмотрим в качестве примера абстрактный конечный последовательный автомат, имеющий три входных состояния x_1, x_2, x_3 и три внутренних состояния q_1, q_2, q_3, и заданный следующей таблицей переходов:

	q_1	q_2	q_3
x_1	q_1	q_3	q_2
x_2	q_3	q_1	q_2
x_3	q_1	q_2	q_1

Эта таблица показывает, например, что при если автомат находится в состоянии q_1, то при подаче на вход автомата состояния x_2, автомат перейдёт в состояние q_3.

Таблица выходов автомата Мили имеет столько же строк и столбцов, сколько таблица переходов. Но на пересечениях проставляются выходные сигналы y_i автомата, которые появляются на выходе автомата для данного перехода.

	q_1	q_2	q_3
x_1	y_1	y_1	y_2
x_2	y_1	y_2	y_1
x_3	y_2	y_2	y_1

Часто эти две таблицы объединяют в одну, называемую таблицей переходов и выходов.

	q_1	q_2	q_3
x_1	q_1 / y_1	q_3 / y_1	q_2 / y_2
x_2	q_3 / y_1	q_1 / y_2	q_2 / y_1
x_3	q_1 / y_2	q_2 / y_2	q_1 / y_1

Если переходы автомата определены не полностью, то он называется не полностью определённым или частичным конечным автоматом. В клетках таблицы переходов, соответствующих неопределённым переходам, ставятся прочерки.

Другим способом представления конечных автоматов является задание их при помощи графов. Для обоих типов автоматов их состояние представляется вершинами графа, а переходы из одного состояния в другое изображаются при помощи направленных дуг. Состояния, вызывающие переходы, приписываются соответствующим дугам графа. Для автомата Мили (с памятью) выходной сигнал ставится в соответствие дуге графа. Если переход автомата из одного состояния в другое может происходить под воздействием двух различных комбинаций входных сигналов, то дуги, соответствующие этим переходам, можно заменить одной дугой, которой соответствует дизъюнкция этих состояний.

Ниже приведёны фрагменты автомата Мили (а, б).

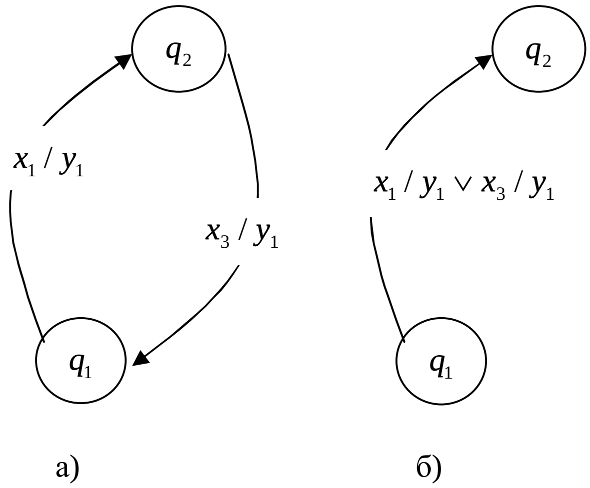


Рисунок 2 Представление конечных автоматов при помощи графов

Матричное представление абстрактного автомата является таблицей, отражающей его структурные особенности. Матрица автомата Мили представляет собой квадратную таблицу, строки которой соответствуют текущим внутренним состояниям автомата, т.е. состояниям во время t, а столбцы соответствуют внутренним состояниям автомата в следующий момент времени, т.е. в момент $t+1$. На пересечениях строк и столбцов указываются условия, вызывающие этот переход. Эти условия представляют собой входное состояние с записанным через косую черту выходным состоянием. При этом некоторые пересечения могут быть пустыми. Это означает, что соответствующий переход отсутствует. В клетках матрицы переходов содержатся записи, относящиеся к дугам графовой модели.

Рассмотрим в качестве примера автомат Мили, имеющий три входных состояния x_1, x_2, x_3 и три внутренних состояния q_1, q_2, q_3, и заданный следующей таблицей переходов.

q_{t+1} / q_t	q_1	q_2	q_3
q_1	–	–	x_2 / y_1
q_2	x_2 / y_2	x_3 / y_2	x_1 / y_1
q_3	x_3 / y_1	$x_1 / y_2 \vee x_2 / y_1$	–

Из матрицы следует, что из состояния q_1 невозможно сразу перейти в состояние q_2.

Построим граф этого автомата.

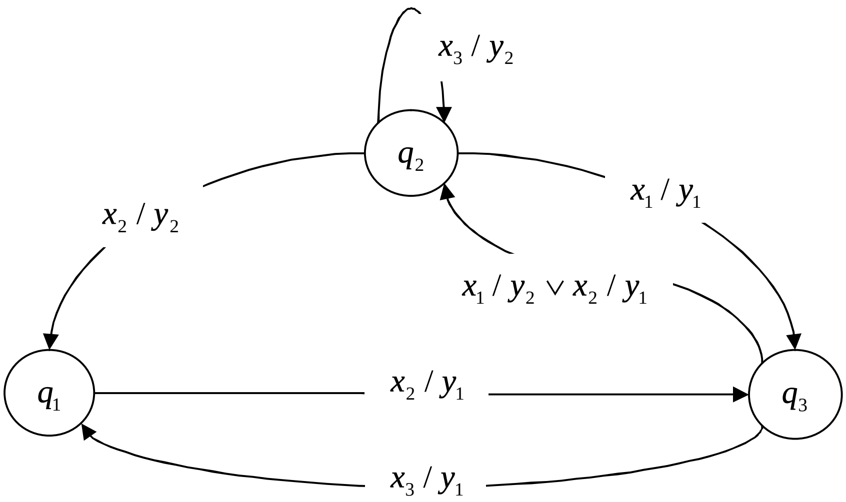


Рисунок 3 Представление автомата Мили при помощи графа

Кроме представления в виде таблицы переходов и выходов, графа или матрицы модель цифрового автомата Мили или Мура можно задать в виде табличной формы матрицы переходов. В случае автомата с памятью она состоит из трёх столбцов: состояние в текущий момент времени q_t, состояние в следующий момент времени q_{t+1} и условие перехода из одного состояния в другое.

Представим рассмотренный выше автомат Мили в такой форме.

q_t	q_{t+1}	Условие перехода
q_1	q_1	–
q_1	q_2	–
q_1	q_3	x_2 / y_1
q_2	q_1	x_2 / y_2
q_2	q_2	x_3 / y_2
q_2	q_3	x_1 / y_1
q_3	q_1	x_3 / y_1
q_3	q_2	$x_1 / y_2 \vee x_2 / y_1$
q_3	q_3	–

Количество строк в табличной форме матрицы переходов определяется числом внутренних состояний q автомата Мили и составляет m^m, где m – число внутренних состояний. Таким образом, для большого числа внутренних состояний такая форма представления модели будет слишком большой.

Варианты заданий

q_t	q_{t+1}	Условие перехода			
q_1	q_1	$x_1 / y_1 \vee x_3 / y_2$	x_3 / y_1	x_2 / y_2	–
q_1	q_2	–	$x_1 / y_2 \vee x_2 / y_1$	x_3 / y_2	x_2 / y_1
q_1	q_3	x_2 / y_1	–	x_1 / y_1	x_2 / y_2
q_2	q_1	x_2 / y_2	$x_1 / y_1 \vee x_3 / y_2$	x_3 / y_1	x_3 / y_2
q_2	q_2	x_3 / y_2	–	$x_1 / y_2 \vee x_2 / y_1$	x_1 / y_1
q_2	q_3	x_1 / y_1	x_2 / y_1	–	x_3 / y_1
q_3	q_1	x_3 / y_1	x_2 / y_2	$x_1 / y_1 \vee x_3 / y_2$	$x_1 / y_2 \vee x_2 / y_1$
q_3	q_2	$x_1 / y_2 \vee x_2 / y_1$	x_3 / y_2	–	–
q_3	q_3	–	x_1 / y_1	x_2 / y_1	$x_1 / y_1 \vee x_3 / y_2$
Вариант		**1**	**2**	**3**	**4**

q_t	q_{t+1}	Условие перехода			
q_1	q_1	$x_2 / y_1 \vee x_1 / y_2$	–	x_2 / y_2	x_2 / y_2
q_1	q_2	x_2 / y_2	x_2 / y_2	x_1 / y_2	x_1 / y_1
q_1	q_3	x_1 / y_2	x_1 / y_1	–	x_2 / y_1
q_2	q_1	–	x_2 / y_1	$x_2 / y_2 \vee x_3 / y_1$	$x_2 / y_1 \vee x_1 / y_2$
q_2	q_2	$x_2 / y_2 \vee x_3 / y_1$	$x_2 / y_1 \vee x_1 / y_2$	–	x_2 / y_2
q_2	q_3	–	x_2 / y_2	x_2 / y_2	x_1 / y_2
q_3	q_1	x_2 / y_2	x_1 / y_2	x_1 / y_1	–
q_3	q_2	x_1 / y_1	–	x_2 / y_1	$x_2 / y_2 \vee x_3 / y_1$
q_3	q_3	x_2 / y_1	$x_2 / y_2 \vee x_3 / y_1$	$x_2 / y_1 \vee x_1 / y_2$	–
Вариант		**5**	**6**	**7**	**8**

q_t	q_{t+1}	Условие перехода			
q_1	q_1	$x_3 / y_2 \vee x_2 / y_1$	x_3 / y_2	$x_3 / y_2 \vee x_2 / y_1$	x_3 / y_2
q_1	q_2	x_1 / y_2	x_2 / y_1	x_1 / y_2	x_2 / y_1
q_1	q_3	–	–	–	–
q_2	q_1	x_3 / y_2	$x_2 / y_2 \vee x_3 / y_1$	x_3 / y_2	$x_2 / y_2 \vee x_3 / y_1$
q_2	q_2	$x_3 / y_2 \vee x_1 / y_1$	$x_3 / y_2 \vee x_2 / y_1$	$x_3 / y_2 \vee x_1 / y_1$	$x_3 / y_2 \vee x_2 / y_1$
q_2	q_3	x_3 / y_2	x_1 / y_2	x_3 / y_2	x_1 / y_2
q_3	q_1	x_2 / y_1	–	x_2 / y_1	–
q_3	q_2	–	x_3 / y_2	–	x_3 / y_2
q_3	q_3	$x_2 / y_2 \vee x_3 / y_1$	$x_3 / y_2 \vee x_1 / y_1$	$x_2 / y_2 \vee x_3 / y_1$	$x_3 / y_2 \vee x_1 / y_1$
Вариант		**9**	**10**	**11**	**12**

q_t	q_{t+1}	Условие перехода			
q_1	q_1	–	x_1 / y_2	–	$x_1 / y_2 \vee x_3 / y_1$
q_1	q_2	x_3 / y_2	$x_1 / y_1 \vee x_2 / y_1$	x_2 / y_1	$x_3 / y_1 \vee x_1 / y_2$
q_1	q_3	x_1 / y_1	–	x_1 / y_2	–
q_2	q_1	$x_1 / y_2 \vee x_3 / y_1$	x_3 / y_2	$x_1 / y_1 \vee x_2 / y_1$	x_2 / y_1
q_2	q_2	$x_3 / y_1 \vee x_1 / y_2$	x_1 / y_1	–	x_1 / y_2
q_2	q_3	–	$x_1 / y_2 \vee x_3 / y_1$	x_3 / y_2	$x_1 / y_1 \vee x_2 / y_1$
q_3	q_1	x_2 / y_1	$x_3 / y_1 \vee x_1 / y_2$	x_1 / y_1	–
q_3	q_2	x_1 / y_2	–	$x_1 / y_2 \vee x_3 / y_1$	x_3 / y_2
q_3	q_3	$x_1 / y_1 \vee x_2 / y_1$	x_2 / y_1	$x_3 / y_1 \vee x_1 / y_2$	x_1 / y_1
Вариант		**13**	**14**	**15**	**16**

q_t	q_{t+1}	Условие перехода			
q_1	q_1	x_1 / y_1	$x_3 / y_2 \vee x_1 / y_2$	x_3 / y_1	x_3 / y_2
q_1	q_2	x_3 / y_1	x_1 / y_1	$x_3 / y_2 \vee x_1 / y_2$	x_3 / y_1
q_1	q_3	x_2 / y_2	x_3 / y_1	x_1 / y_1	$x_3 / y_2 \vee x_1 / y_2$
q_2	q_1	$x_2 / y_2 \vee x_1 / y_1$	x_2 / y_2	x_3 / y_1	x_1 / y_1
q_2	q_2	–	$x_2 / y_2 \vee x_1 / y_1$	x_2 / y_2	x_3 / y_1
q_2	q_3	$x_1 / y_1 \vee x_3 / y_2$	–	$x_2 / y_2 \vee x_1 / y_1$	x_2 / y_2
q_3	q_1	x_3 / y_2	$x_1 / y_1 \vee x_3 / y_2$	–	$x_2 / y_2 \vee x_1 / y_1$
q_3	q_2	x_3 / y_1	x_3 / y_2	$x_1 / y_1 \vee x_3 / y_2$	–
q_3	q_3	$x_3 / y_2 \vee x_1 / y_2$	x_3 / y_1	x_3 / y_2	$x_1 / y_1 \vee x_3 / y_2$
Вариант		**17**	**18**	**19**	**20**

Порядок выполнения работы

1. Используя табличную форму представления матрицы переходов, получить модель конечного автомата с памятью в виде таблицы переходов и выходов, в виде графа, а также в виде матрицы.
2. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:
   – вариант задания на лабораторную работу;
   – таблицу переходов и выходов;
   – граф автомата;
   – матрицу автомата.

Контрольные вопросы

1. Что называют цифровым автоматом?
2. Чем характеризуется цифровой автомат, что такое конечный автомат?
3. Каковы типы конечных автоматов?
4. Дайте определения двух типов конечных автоматов.
5. В чём состоит понятие автоматного времени?
6. Какие бывают конечные автоматы по способу задания времени?

7. Как понимается автоматное время в разных типах конечных автоматов?
8. Опишите модель Клини.
9. Что такое функция переходов и выходов у автоматов с памятью?
10. Каковы способы задания автомата Мили?
11. Какие столбцы содержит табличная форма матрицы перехода автомата Мили и сколько в ней строк?
12. Преобразуйте модель автомата Мили из одной формы представления в другую, заданную преподавателем.

Лабораторная работа №8. Формы представления цифровых автоматов без памяти (автоматов Мура)

Цель работы: изучить различные способы представления цифровых автоматов без памяти: табличный, при помощи графов и матричный.

Теоретическая часть

В случае синтеза автомата без памяти (автомата Мура) строится только таблица переходов. Так как в комбинационном автомате выходные состояния однозначно соответствуют внутренним, то в таблице имеется дополнительная строка, отражающая это соответствие. В качестве примера рассмотрим автомат Мура с входными состояниями x_1, x_2, x_3, внутренними q_1, q_2, q_3, q_4 и выходными u_1, u_2. Зададим таблицу переходов:

	u_1	u_1	u_2	u_1
	q_1	q_2	q_3	q_4
x_1	q_1	q_3	q_4	q_2
x_2	q_4	q_2	q_1	q_3
x_3	q_2	q_1	q_4	q_2

Другим способом представления конечных автоматов является задание их при помощи графов. Для обоих типов автоматов их состояние представляется вершинами графа, а переходы из одного состояния в другое изображаются при помощи направленных дуг. Состояния, вызывающие переходы, приписываются соответствующим дугам графа. Для автомата Мура (без памяти) выходной сигнал ставится в соответствие вершине графа. Если переход автомата из одного состояния в другое может происходить под воздействием двух различных комбинаций входных сигналов, то дуги, соответствующие этим переходам, можно заменить одной дугой, которой соответствует дизъюнкция этих состояний. Ниже приведёны фрагменты автомата Мили (а, б) и Мура (в).

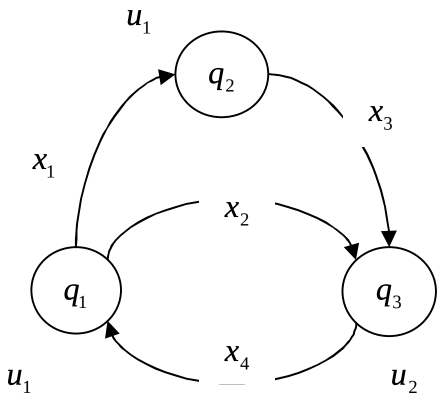


Рисунок 3 Представление конечных автоматов при помощи графов

Матричное представление автомата Мура имеет один дополнительный столбец, с помощью которого задаётся соответствие между внутренними состояниями и выходными сигналами автомата.

$q_t \backslash q_{t+1}$	q_1	q_2	q_3	q_4	u_t
q_1	x_1	x_3	–	x_2	u_1
q_2	x_3	x_2	x_1	–	u_1
q_3	x_2	–	–	$x_1 \vee x_3$	u_2
q_4	–	$x_1 \vee x_3$	x_2	–	u_1

Построим граф этого автомата Мура.

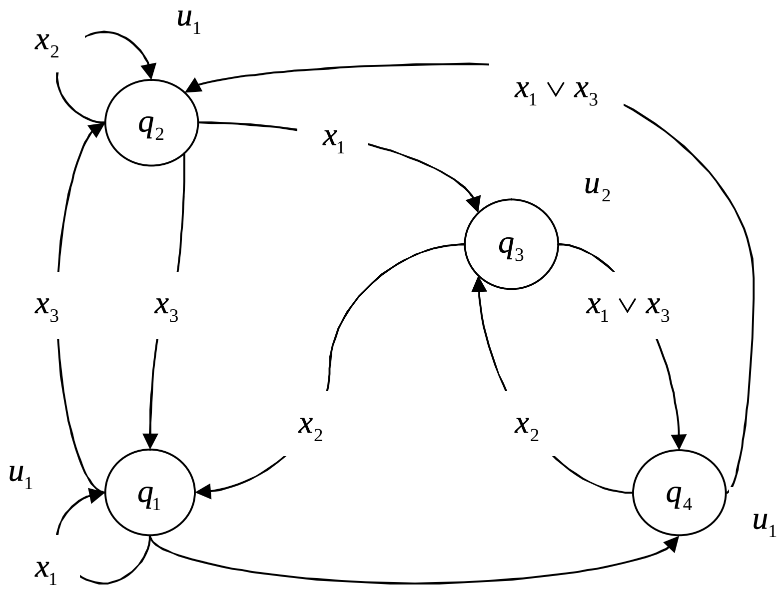


Рисунок 4 Представление автомата Мура при помощи графа

Построим теперь табличную форму матрицы переходов для автомата Мура, рассмотренного выше. В этом случае добавляется столбец, отражающий соответствия переходов внутренних состояний автомата смене выходных состояний u_t / u_{t+1}.

q_t	q_{t+1}	Условие перехода	u_t / u_{t+1}
q_1	q_1	x_1	u_1 / u_1
q_1	q_2	x_3	u_1 / u_1
q_1	q_3	–	u_1 / u_2
q_1	q_4	x_2	u_1 / u_1
q_2	q_1	x_3	u_1 / u_1
q_2	q_2	x_2	u_1 / u_1
q_2	q_3	x_1	u_1 / u_2
q_2	q_4	–	u_1 / u_1
q_3	q_1	x_2	u_2 / u_1
q_3	q_2	–	u_2 / u_1
q_3	q_3	–	u_2 / u_2
q_3	q_4	$x_1 \vee x_3$	u_2 / u_1
q_4	q_1	–	u_1 / u_1
q_4	q_2	$x_1 \vee x_3$	u_1 / u_1
q_4	q_3	x_2	u_1 / u_2
q_4	q_4	–	u_1 / u_1

Количество строк в табличной форме матрицы переходов определяется числом внутренних состояний q автомата Мура и составляет m^m, где m – число внутренних состояний.

Варианты заданий

q_t	q_{t+1}	u_t / u_{t+1}	**Условие перехода**				
q_1	q_1	u_1 / u_1	$x_1 \vee x_2$	–	x_3	x_2	$x_1 \vee x_2$
q_1	q_2	u_1 / u_1	x_3	$x_1 \vee x_3$	–	$x_2 \vee x_3$	x_3
q_1	q_3	u_1 / u_2	–	x_2	$x_1 \vee x_2$	x_1	–
q_1	q_4	u_1 / u_1	x_2	–	–	–	x_2
q_2	q_1	u_1 / u_1	x_2	$x_1 \vee x_2$	–	x_3	x_2
q_2	q_2	u_1 / u_1	$x_2 \vee x_3$	x_3	$x_1 \vee x_3$	–	$x_2 \vee x_3$
q_2	q_3	u_1 / u_2	x_1	–	x_2	$x_1 \vee x_2$	x_1
q_2	q_4	u_1 / u_1	–	x_2	–	–	–
q_3	q_1	u_2 / u_1	x_3	x_2	$x_1 \vee x_2$	–	x_3
q_3	q_2	u_2 / u_1	–	$x_2 \vee x_3$	x_3	$x_1 \vee x_3$	–
q_3	q_3	u_2 / u_2	$x_1 \vee x_2$	x_1	–	x_2	$x_1 \vee x_2$
q_3	q_4	u_2 / u_1	–	–	x_2	–	–
q_4	q_1	u_1 / u_1	–	x_3	x_2	$x_1 \vee x_2$	–
q_4	q_2	u_1 / u_1	$x_1 \vee x_3$	–	$x_2 \vee x_3$	x_3	$x_1 \vee x_3$
q_4	q_3	u_1 / u_2	x_2	$x_1 \vee x_2$	x_1	–	x_2
q_4	q_4	u_1 / u_1	–	–	–	x_2	–
Вариант			**1**	**2**	**3**	**4**	**5**

q_t	q_{t+1}	u_t / u_{t+1}	Условие перехода				
q_1	q_1	u_1 / u_1	$x_3 \vee x_1$	$x_2 \vee x_3$	$x_1 \vee x_3$	x_2	x_2
q_1	q_2	u_1 / u_1	–	–	x_3	–	$x_1 \vee x_2$
q_1	q_3	u_1 / u_2	x_3	x_1	–	–	x_1
q_1	q_4	u_1 / u_1	–	$x_3 \vee x_1$	$x_2 \vee x_3$	$x_1 \vee x_3$	x_2
q_2	q_1	u_1 / u_1	x_2	–	–	x_3	–
q_2	q_2	u_1 / u_1	$x_1 \vee x_2$	x_3	x_1	–	–
q_2	q_3	u_1 / u_2	x_1	–	$x_3 \vee x_1$	$x_2 \vee x_3$	$x_1 \vee x_3$
q_2	q_4	u_1 / u_1	x_2	x_2	–	–	x_3
q_3	q_1	u_2 / u_1	–	$x_1 \vee x_2$	x_3	x_1	–
q_3	q_2	u_2 / u_1	–	x_1	–	$x_3 \vee x_1$	$x_2 \vee x_3$
q_3	q_3	u_2 / u_2	$x_1 \vee x_3$	x_2	x_2	–	–
q_3	q_4	u_2 / u_1	x_3	–	$x_1 \vee x_2$	x_3	x_1
q_4	q_1	u_1 / u_1	–	–	x_1	–	$x_3 \vee x_1$
q_4	q_2	u_1 / u_1	$x_2 \vee x_3$	$x_1 \vee x_3$	x_2	x_2	–
q_4	q_3	u_1 / u_2	–	x_3	–	$x_1 \vee x_2$	x_3
q_4	q_4	u_1 / u_1	x_1	–	–	x_1	–
Вариант			**6**	**7**	**8**	**9**	**10**

q_t	q_{t+1}	u_t / u_{t+1}	Условие перехода				
q_1	q_1	u_1 / u_1	x_1	x_2	x_3	–	–
q_1	q_2	u_1 / u_1	–	$x_2 \vee x_3$	–	–	$x_1 \vee x_3$
q_1	q_3	u_1 / u_2	$x_2 \vee x_3$	x_1	x_2	x_3	–
q_1	q_4	u_1 / u_1	–	–	$x_2 \vee x_3$	–	–
q_2	q_1	u_1 / u_1	$x_1 \vee x_3$	$x_2 \vee x_3$	x_1	x_2	x_3
q_2	q_2	u_1 / u_1	x_3	–	–	$x_2 \vee x_3$	–
q_2	q_3	u_1 / u_2	$x_1 \vee x_2$	$x_1 \vee x_3$	$x_2 \vee x_3$	x_1	x_2
q_2	q_4	u_1 / u_1	x_1	x_3	–	–	$x_2 \vee x_3$
q_3	q_1	u_2 / u_1	–	$x_1 \vee x_2$	$x_1 \vee x_3$	$x_2 \vee x_3$	x_1
q_3	q_2	u_2 / u_1	$x_1 \vee x_3$	x_1	x_3	–	–
q_3	q_3	u_2 / u_2	–	–	$x_1 \vee x_2$	$x_1 \vee x_3$	$x_2 \vee x_3$
q_3	q_4	u_2 / u_1	–	$x_1 \vee x_3$	x_1	x_3	–
q_4	q_1	u_1 / u_1	x_3	–	–	$x_1 \vee x_2$	$x_1 \vee x_3$
q_4	q_2	u_1 / u_1	–	–	$x_1 \vee x_3$	x_1	x_3
q_4	q_3	u_1 / u_2	x_2	x_3	–	–	$x_1 \vee x_2$
q_4	q_4	u_1 / u_1	$x_2 \vee x_3$	–	–	$x_1 \vee x_3$	x_1
Вариант			**11**	**12**	**13**	**14**	**15**

q_t	q_{t+1}	u_t / u_{t+1}	**Условие перехода**				
q_1	q_1	u_1 / u_1	–	x_1	x_2	x_3	x_1
q_1	q_2	u_1 / u_1	x_1	–	–	–	$x_2 \vee x_3$
q_1	q_3	u_1 / u_2	x_2	x_3	x_1	$x_1 \vee x_3$	$x_2 \vee x_3$
q_1	q_4	u_1 / u_1	–	–	$x_2 \vee x_3$	$x_1 \vee x_2$	$x_1 \vee x_3$
q_2	q_1	u_1 / u_1	x_1	$x_1 \vee x_3$	$x_2 \vee x_3$	–	–
q_2	q_2	u_1 / u_1	$x_2 \vee x_3$	$x_1 \vee x_2$	$x_1 \vee x_3$	–	x_1
q_2	q_3	u_1 / u_2	$x_2 \vee x_3$	–	–	x_1	–
q_2	q_4	u_1 / u_1	$x_1 \vee x_3$	–	x_1	x_2	x_3
q_3	q_1	u_2 / u_1	–	x_1	–	–	–
q_3	q_2	u_2 / u_1	x_1	x_2	x_3	x_1	$x_1 \vee x_3$
q_3	q_3	u_2 / u_2	–	–	–	$x_2 \vee x_3$	$x_1 \vee x_2$
q_3	q_4	u_2 / u_1	x_3	x_1	$x_1 \vee x_3$	$x_2 \vee x_3$	–
q_4	q_1	u_1 / u_1	–	$x_2 \vee x_3$	$x_1 \vee x_2$	$x_1 \vee x_3$	–
q_4	q_2	u_1 / u_1	$x_1 \vee x_3$	$x_2 \vee x_3$	–	–	x_1
q_4	q_3	u_1 / u_2	$x_1 \vee x_2$	$x_1 \vee x_3$	–	x_1	x_2
q_4	q_4	u_1 / u_1	–	–	x_1	–	–
Вариант			**16**	**17**	**18**	**19**	**20**

Порядок выполнения работы

1. Используя табличную форму представления матрицы переходов, получить модель конечного автомата без памяти в виде таблицы переходов и выходов, в виде графа, а также в виде матрицы.
2. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:
   – вариант задания на лабораторную работу;
   – таблицу переходов и выходов;
   – граф автомата;
   – матрицу автомата.

Контрольные вопросы

1. Как составляется таблица переходов для синтеза автомата Мура?
2. Как задать автомат Мура при помощи графа?
3. Опишите матричное представление автомата Мура.
4. Какие столбцы содержит табличная форма матрицы перехода автомата Мура и сколько в ней строк?
5. Преобразуйте модель автомата Мура из одной формы представления в другую, заданную преподавателем.

Лабораторная работа №9. Элементарные конечные автоматы: структура и функционирование однотактного RS-триггера

Цель работы: изучить виды однотактных RS -триггеров и их работу.

Теоретическая часть

Под синтезом комбинационных схем понимают процесс создания цифровых схем, реализующих функции заданных абстрактных цифровых автоматов. Процесс синтеза заключается в выборе типов запоминающих устройств и синтезе связанных с ними комбинационных схем. По этой причине в этой лабораторной работе будет рассмотрено функционирование элементов памяти – элементарных конечных автоматов, или триггеров.

Триггеры предназначены для запоминания двоичной информации. Использование триггеров позволяет реализовывать устройства оперативной памяти. Также триггеры могут использоваться и для построения цифровых устройств с памятью, таких как счётчики, преобразователи последовательного кода в параллельный и т.д.

Триггеры являются автоматами с двумя устойчивыми состояниями, обладающие полнотой переходов и выходов. Полнота переходов определяется наличием хотя бы одного входного сигнала, который переводит автомат из одного состояния в другое. Полнота выхода означает, что двум различным внутренним состояниям соответствуют два различных выходных сигнала.

При синтезе конечного автомата с памятью в качестве элементарных автоматов используются триггеры, схемы которых отличаются друг от друга как числом входов, так и логикой функционирования. При этом все триггеры являются автоматами Мура (без памяти), а автоматы, использующие триггеры – автоматами Мили (с памятью).

Рассмотрим некоторые особенности описания триггеров.

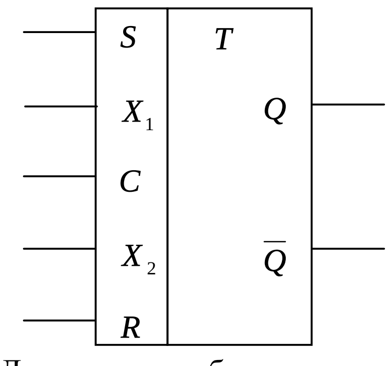
Внутреннее состояние триггера обозначается буквой Q (или q). При Q=1 триггер находится в единичном состоянии, при Q=0 – в нулевом состоянии.

Выходной сигнал полностью повторяет внутреннее состояние триггера и также обозначается Q.

Триггер имеет два выхода, сигналы на которых взаимно инверсны: Q и $\overline{Q}$.

Если триггер является синхронным, то импульсы синхронизации подаются на специальный синхронизирующий вход и обозначаются C.

Все триггеры, как правило, имеют два типа входов: рабочие и установочные. Вход установки триггера в единичное состояние обозначается S (Set), вход установки триггера в нулевое состояние обозначается R ($Reset$). Установочные входы всегда асинхронны. Условное графическое обозначение триггера:



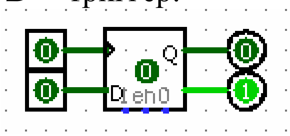
Детали этого обозначения могут меняться. Тип триггера, обозначенный на данном рисунке как T, обозначает одноступенчатый (однотактный) триггер. Двухступенчатое (двухтактное) устройство такого типа обозначается TT.

Рабочие входы обозначены X_1 и X_2. Эти обозначения меняются в зависимости от типа триггера. Наиболее распространёнными типами являются D (*delay*, триггер задержки), T (счётный) RS (*ResetSet*, он же SR) и JK (универсальный).

Входы S и R являются установочными, вход C - синхронизирующим. Эти обозначения, как правило, не меняются.

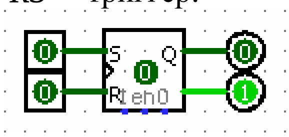
В программе моделирования логических схем Logisim используются немного другие обозначения элементов.

D – триггер:

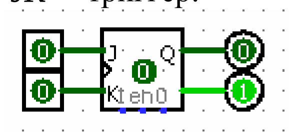


T – триггер:

RS – триггер:



JK – триггер:



На этих обозначениях установочные входы расположены снизу.

Основным триггером, на котором базируются все остальные, является RS-триггер. Он имеет два логических входа:

R - установка 0 (от слова reset);

S - установка 1 (от слова set).

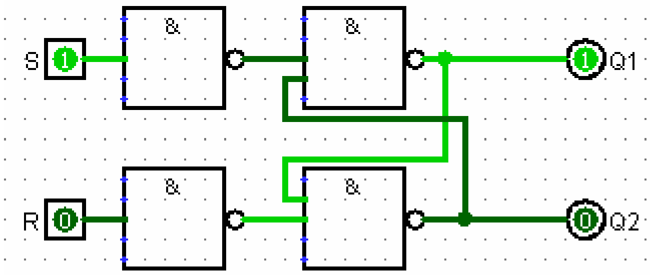
RS-триггер имеет два выхода:

Q - прямой;

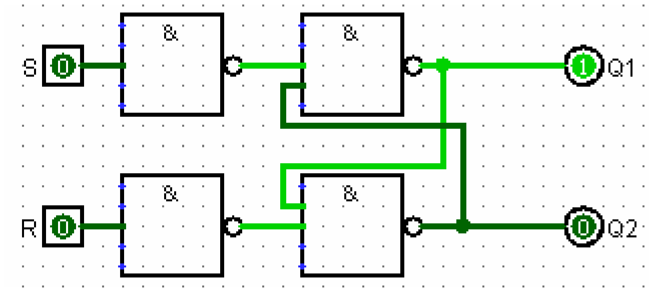
$\overline{Q}$ - обратный (инверсный).

Состояние триггера определяется состоянием прямого выхода. Простейший асинхронный RS-триггер состоит из двух логических элементов, связанных перекрёстной положительной обратной связью:

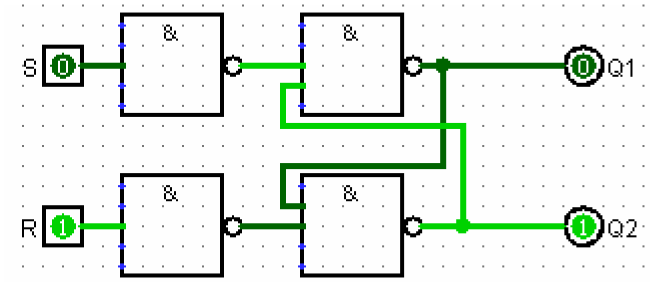
Рассмотрим работу однотактного асинхронного триггера. Пусть рабочие входы триггера находятся в состоянии $R=0$, $S=1$. Нижний двухвходовой логический элемент выполняет логическую функцию И-НЕ, т.е. единица на любом его входе приводит к тому, что на его выходе будет логическая единица $Q2=1$. Но в то же время на оба входа верхнего элемента И-НЕ поданы нули, что приводит к появлению единицы на его выходе $Q1=1$. В то же время эта единица приходит на вход нижнего элемента, поэтому на его выходе получаем $Q2=0$. Триггер находится в единичном состоянии $Q1=1$, $Q2=0$.



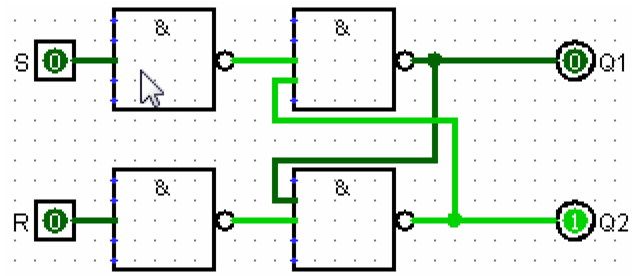
Если теперь на оба входа подать 0 (S=0, R=0), то на выходе ситуация не изменится, т.к. несмотря на то, что на нижний вход нижнего логического элемента будет поступать 0, на его верхний вход поступает 1 с выхода верхнего логического элемента. Триггер будет находиться в единичном состоянии, пока на вход R не поступит сигнал сброса R=1.



Пусть теперь R=1, S=0. Тогда $Q1$=0, а $Q2$=1. Триггер переключился в нулевое состояние.



Если после этого убрать сигнал сброса (R=0, S=0), то триггер не изменит своего состояния.



Для описания работы триггера используют таблицу состояний (переходов). Обозначим:

$Q(t)$ – состояние триггера до поступления управляющих сигналов (изменения на входах R и S);

$Q(t+1)$ – состояние триггера после изменения на входах R и S.

R	S	$Q(t)$	$Q(t+1)$	Режимы
0	0	0	0	Хранения информации
0	0	1	1	
0	1	0	1	Режим установки единицы: $S=1$
0	1	1	1	
1	0	0	0	Режим установки нуля: $R=1$
1	0	1	0	
1	1	0	*	Запрещённый: $R=1,\ S=1$
1	1	1	*	

Граф RS-триггера:

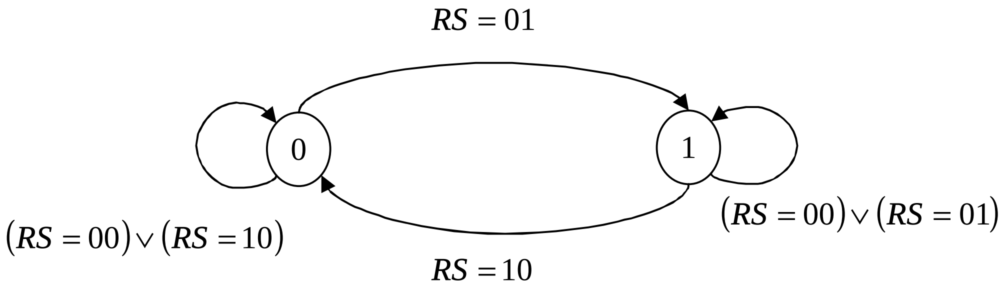


Рисунок 3 Представление RS-триггера в виде графа

Представим таблицу переходов RS-триггера в следующем виде:

$Q(t)$		0	1
$R(t)$	$S(t)$		
0	0	0	1
0	1	1	1
1	1	*	*
1	0	0	0

Используем таблицу переходов как карту Карно. Так как комбинация 11 является запрещённой, можно поставить в эти клетки нули или единицы. Поставим единицы. Получим выражение для функции переходов RS-триггера:

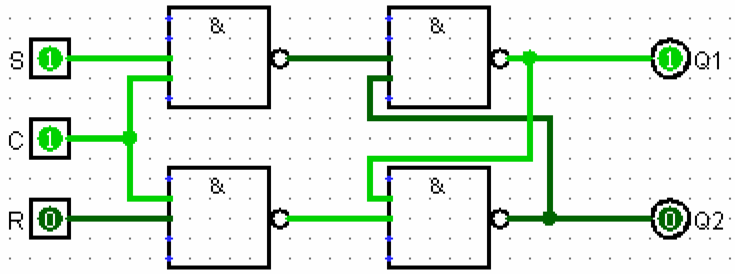
$$Q(t+1)=S(t)\vee\overline{R(t)}\cdot Q(t).$$

Используя граф триггера, можно получить табличную форму матрицы переходов. При этом можно упростить условия переходов, используя законы алгебры логики, например: $\overline{R}\cdot\overline{S}\vee R\cdot\overline{S}=\overline{S}\cdot(\overline{R}\vee R)=\overline{S}$.

$Q(t)$	$Q(t+1)$	Условие перехода	Упрощённое условие
0	0	$\overline{R}\cdot\overline{S}\vee R\cdot\overline{S}$	$-\overline{S}$
0	1	$\overline{R}\cdot S$	$\overline{R}\cdot S$
1	0	$R\cdot\overline{S}$	$R\cdot\overline{S}$
1	1	$\overline{R}\cdot\overline{S}\vee\overline{R}\cdot S$	$\overline{R}-$

Запись $-\overline{S}$ означает, что на вход S подаётся сигнал 0, а на вход R – безразлично.

Схема RS-триггера позволяет запоминать состояние логической схемы. При изменении входных сигналов может возникать переходный процесс (этот процесс называется «гонки сигналов»). Поэтому запоминать состояния логической схемы нужно только в определённые моменты времени, когда все переходные процессы закончены, и сигнал на выходе комбинационной схемы соответствует выполняемой ею функции. Поэтому большинство цифровых схем требуют сигнала синхронизации (тактового сигнала). Все переходные процессы в комбинационной логической схеме должны закончиться за время периода синхросигнала, подаваемого на входы триггеров. Триггеры, запоминающие входные сигналы только в момент времени, определяемый сигналом синхронизации, называются синхронными. Принципиальная схема однотактного синхронного RS-триггера приведена на рисунке.

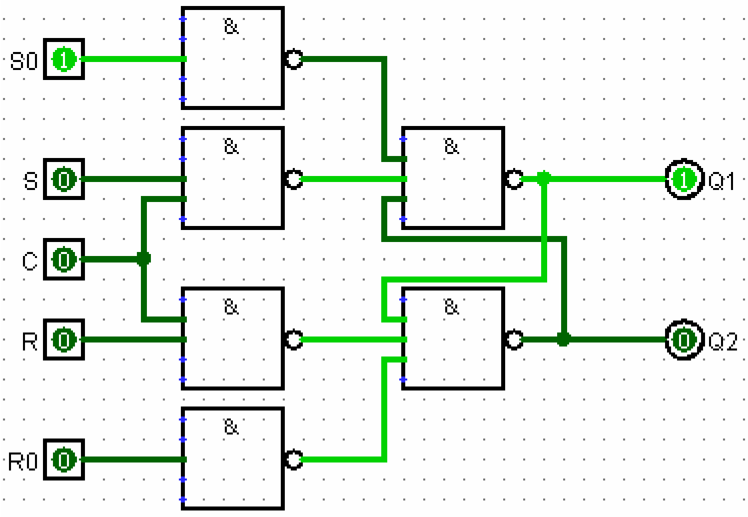


На вход C подаётся синхросигнал. Состояние триггера меняется только в моменты его подачи. Без синхроимпульса триггер сохраняет своё состояние неизменным.

Таблица переходов синхронного RS-триггера:

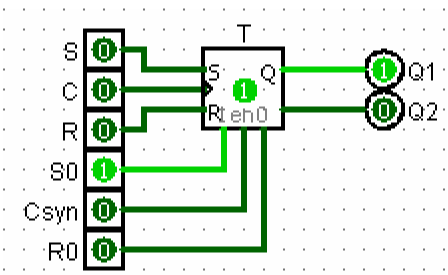
R	S	C	$Q(t)$	$Q(t+1)$	Режимы
0	0	1	0	0	Хранения информации
0	0	1	1	1	
0	1	1	0	1	Установки единицы: $S=1$
0	1	1	1	1	
1	0	1	0	0	Установки нуля: $R=1$
1	0	1	1	0	
1	1	1	0	*	Запрещённый: $R=1$, $S=1$
1	1	1	1	*	

Для предварительной установки триггера в определённое состояние применяют установочные входы ($R0$ и $S0$ на рисунке).



Установочные входы всегда асинхронные.

Обозначение RS -триггера в программе Logisim:



Входы S и R являются рабочими входами триггера. Значение, установленное на этих входах, становится значением внутреннего состояния триггера в момент срабатывания тактирующего входа C.

На вход C (на символе устройства обозначен треугольником) подаётся тактовый сигнал. В момент, когда значение на этом входе меняется с 0 на 1 (передний фронт), значение триггера будет обновлено в соответствии входами S и R. Пока значение на этом входе остаётся 0 или 1, входы S и R не имеют эффекта.

Входы $R0$ (на символе устройства обозначен «1») и $S0$ (на символе устройства обозначен «0») являются установочными. Пока $S0$ =1, состояние триггера равно на 1. Это происходит асинхронно - то есть вне зависимости от текущего значения на тактовом входе. Пока на этом входе 1, другие входы не имеют эффекта, за исключением входа $R0$, который имеет приоритет. Пока $R0$ =1, значение триггера фиксировано на 0. Это происходит асинхронно - то есть вне зависимости от текущего значения на тактовом входе. Пока на нём 1, другие входы не имеют эффекта.

Вход *Csyn* (на символе устройства обозначен «eh») является синхронизатором тактового входа. Когда на этом входе 0, срабатывания тактового входа игнорируются. Текущий бит по-прежнему поступает на выход. Срабатывания тактового входа включаются, когда значение этого входа 1 или не определено.

Прямой выход триггера обозначен на символе устройства обозначен Q.

Выходы триггера: $Q1$ прямой и $Q2$ инверсный.

Варианты заданий

Варианты заданий для этой лабораторной работы отсутствуют.

Порядок выполнения работы

1. Используя Logisim, создать модели RS -триггеров в базисе И-НЕ:
   - однотактного асинхронного;
   - однотактного синхронного;
   - однотактного синхронного с установочными входами;
2. Для каждой схемы составить таблицу переходов.
3. В разделе «память» программы Logisim найти модель RS -триггера и, моделируя его работу, составить его таблицу переходов.
4. Сравнить все полученные таблицы переходов. При обнаружении таблицы, не совпадающей с правильной, проверить работу соответствующей схемы и исправить ошибку.
5. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:
   - схемы RS -триггеров;
   - таблицы переходов триггеров.

Контрольные вопросы

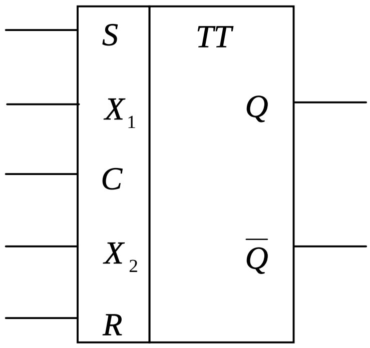
1. Что понимают под синтезом комбинационных схем?
2. Для чего предназначены триггеры?
3. Какими цифровыми автоматами являются триггеры, что такое полнота переходов и выходов?
4. Какие бывают состояния триггера, как они обозначаются?
5. Какие типы входов есть у триггера?
6. Каково условное графической обозначение триггера?
7. Какие существуют типы триггеров по выполняемой функции?
8. Нарисуйте схему однотактного асинхронного RS -триггера в базисе И-НЕ и расскажите о её работе.
9. В каких режимах может работать однотактный асинхронный RS -триггер?
10. Нарисуйте таблицу переходов однотактного асинхронного RS -триггера.
11. Нарисуйте граф однотактного асинхронного RS -триггера.
12. Выведите выражение для функции переходов RS -триггера.
13. Нарисуйте табличную форму матрицы переходов RS -триггера.
14. Нарисуйте схему однотактного синхронного RS -триггера в базисе И-НЕ и расскажите о её работе.
15. Нарисуйте таблицу переходов однотактного синхронного RS -триггера.
16. Нарисуйте схему однотактного синхронного RS -триггера с установочными входами в базисе И-НЕ и расскажите о её работе.

Лабораторная работа №10. Элементарные конечные автоматы: структура и функционирование двухтактного RS-триггера

Цель работы: изучить работу триггеров: задержки (D), счётного (T), а также RS-триггера и JK- триггера.

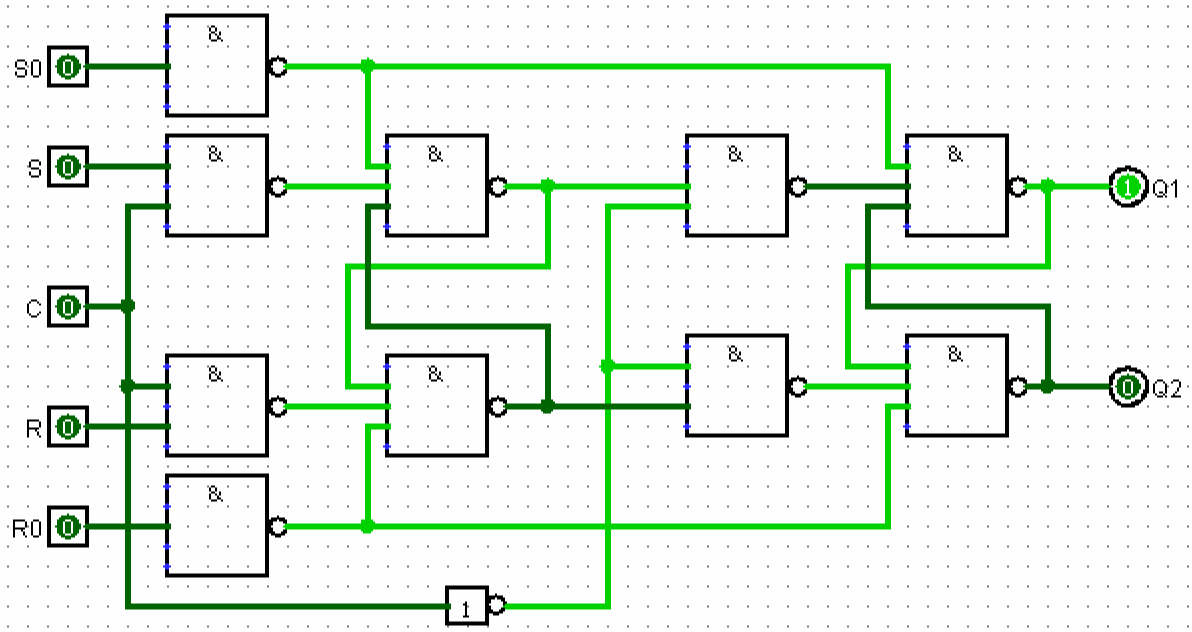
Теоретическая часть

Во многих схемах, например, в регистрах сдвига, устойчивая работа триггера возможна только в том случае, если занесение в него новой информации осуществляется после передачи информации о его состоянии в следующий триггер. В этих случаях используют двухтактные триггеры. Кроме этого, на двухтактных триггерах строятся некоторые другие типы триггеров.

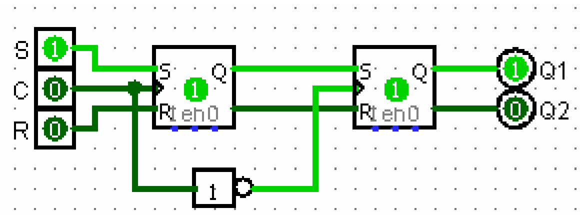


Условное обозначение двухтактного триггера содержит буквы *TT* в поле обозначения триггера. Символы и функции остальных входов аналогичны однотактному.

Двухтактный *RS*-триггер содержит два однотактных триггера и инвертор в цепи синхронизации. При подаче синхросигнала (C=1) входная информация заносится на первый триггер, а во втором триггере еще сохраняется старая информация, гарантируя ее передачу на следующий триггер. После окончания активного C=1, становится активным сигнал синхронизации с выхода инвертора, который записывает входную информацию (с задержкой на время действия C=1) на второй триггер, который и является элементом хранения.



В Logisim нет отдельного элемента для двухтактного триггера, он строится на двух однотактных.



Варианты заданий

Варианты заданий для этой лабораторной работы отсутствуют.

Порядок выполнения работы

1. Используя Logisim, построить схему двухтактного RS-триггера в базисе И-НЕ.
2. Используя Logisim, построить схему двухтактного RS-триггера, применив однотактные триггеры из меню «Память»
3. Построить таблицы переходов обоих устройств и сравнить их. При обнаружении различий проверить схемы.
4. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:
   - схемы двухтактных RS-триггеров;
   - таблицы переходов триггеров.

Контрольные вопросы

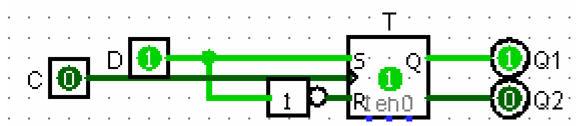
1. Для чего применяют двухтактные триггеры?
2. Нарисуйте схему двухтактного RS-триггера в базисе И-НЕ и расскажите о её работе.
3. Нарисуйте схему двухтактного RS-триггера на базе двух однотактных и расскажите о её работе.

Лабораторная работа №11. Элементарные конечные автоматы: структура и функционирование D-триггера, T-триггера и JK-триггера

Цель работы: изучить работу триггеров: задержки (D), счётного (T), а также JK-триггера.

Теоретическая часть

Триггер задержки (D-триггер) можно построить на базе синхронного RS-триггера.



Этот триггер имеет один информационный вход (D-вход). Существуют только синхронные D-триггеры. Состояние информационного входа передаётся на выход под действием синхроимпульса (вход C). Если на входе D=1, то по приходу синхроимпульса C=1 состояние выхода $Q1$=1. Если D=0, то по приходу синхроимпульса C=1 состояние выхода $Q1$=0.

Для описания работы триггера используем таблицу состояний (переходов).

C	D	$Q(t)$	$Q(t+1)$	Режимы
0	*	0	0	Хранения информации
0	*	1	1	
1	0	*	0	Записи информации
1	1	*	1	

Граф D-триггера:

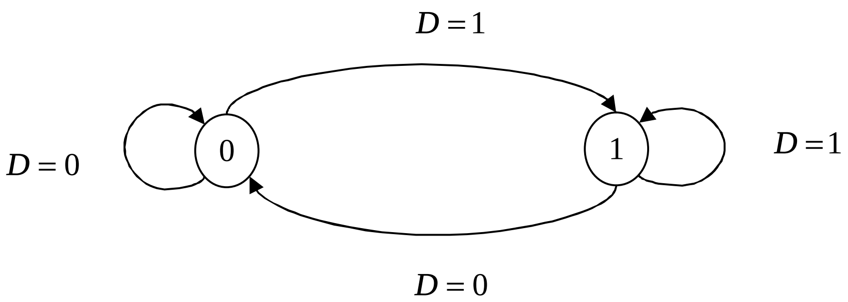


Рисунок 4 Представление D-триггера в виде графа

Используя граф триггера, можно получить табличную форму матрицы переходов D-триггера:

$R(t)$ \ $Q(t)$	0	1
0	0	0
1	1	1

Интерпретировав таблицу переходов как карту Карно, запишем минимизированное выражение для функции перехода D-триггера.

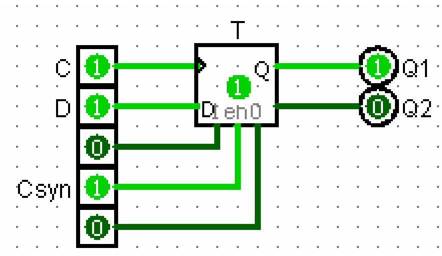
$$Q(t+1) = D(t).$$

Для описания работы триггера используем матрицу переходов.

$Q(t)$	$Q(t+1)$	Условие перехода
0	0	$\overline{D}$
0	1	D
1	0	$\overline{D}$
1	1	D

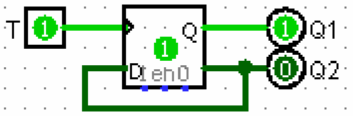
Этот триггер иногда называют «защёлкой» ($lock$) и используют для хранения поданного на его вход бита.

Обозначение D-триггера в программе Logisim:

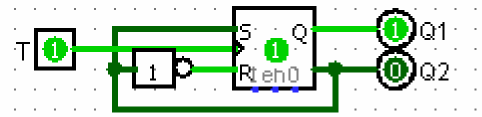


Счётные триггеры бывают асинхронные и синхронные.

Асинхронный T-триггер можно построить на базе D-триггера, если его вход D соединить с инверсным выходом $Q2$.



Этот T-триггер имеет один счётный информационный вход T. Если в момент t на его вход поступает сигнал логической единицы, то в момент времени $t+1$ триггер переключается в противоположное состояние. Также этот триггер можно построить на базе RS-триггера.



Этот триггер предназначен для подсчёта входных сигналов в двоичной системе счисления.

Для описания работы триггера используем таблицу состояний (переходов).

T	$Q(t)$	$Q(t+1)$	Режимы
0	0	0	Хранения информации
1	1	1	
0	0	1	Записи информации
1	1	0	

Граф T-триггера:

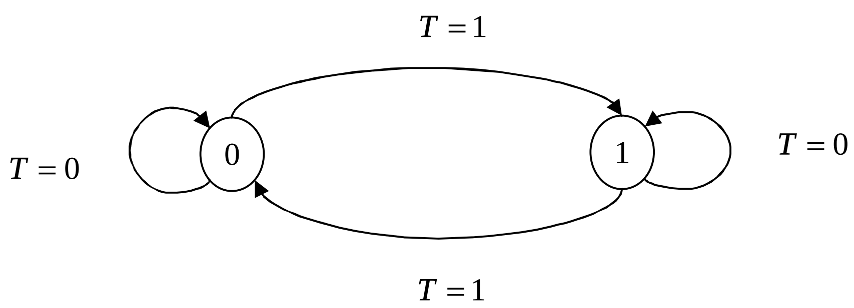


Рисунок 4 Представление T-триггера в виде графа

Используя граф триггера, можно получить табличную форму матрицы переходов T-триггера:

$R(t)$ \ $Q(t)$	0	1
0	0	1
1	1	0

Интерпретировав таблицу переходов как карту Карно, запишем минимизированное выражение для функции переходов D-триггера.

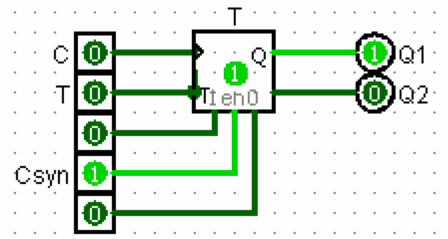
$$Q(t+1) = \overline{T(t)} \cdot Q(t) \vee T(t) \cdot \overline{Q(t)}.$$

Для описания работы триггера используем матрицу переходов.

$Q(t)$	$Q(t+1)$	Условие перехода
0	0	$\overline{T}$
0	1	T
1	0	T
1	1	$\overline{T}$

В программе Logisim представлен синхронный T-триггер. Когда его тактовый вход C срабатывает, значение, хранящееся в триггере, меняется или остаётся прежним в зависимости от того, какое значение на входе T: 1 или 0.

Обозначение в программе Logisim:



Универсальный триггер (JK-триггер) применяется для построения любых типов триггеров. Он является двухтактным и строится на основе RS-триггера, от которого отличается тем, что комбинация $SR = 11$ не является для него запрещённой. Такой триггер имеет информационные входы J и K, которые по своему влиянию аналогичны входам S и R синхронного RS-триггера.

Рассмотрим его работу. При JK=10 триггер по тактовому импульсу устанавливается в состояние Q=1. Если JK=01 - переключается в состояние Q=0. При JK=00 триггер хранит ранее принятую информацию. При комбинации JK=11, запрещённой для RS-триггера, переводит JK-триггер в противоположное состояние.

Таблица состояний:

K	J	C	$Q(t)$	$Q(t+1)$	Режимы
0	0	1	0	0	Хранения информации
0	0	1	1	1	
0	1	1	0	1	Установки единицы: $J = 1$
0	1	1	1	1	
1	0	1	0	0	Установки нуля: $K = 1$
1	0	1	1	0	
1	1	1	0	1	Инверсии: $R = 1,\ S = 1$
1	1	1	1	0	

Граф JK-триггера:

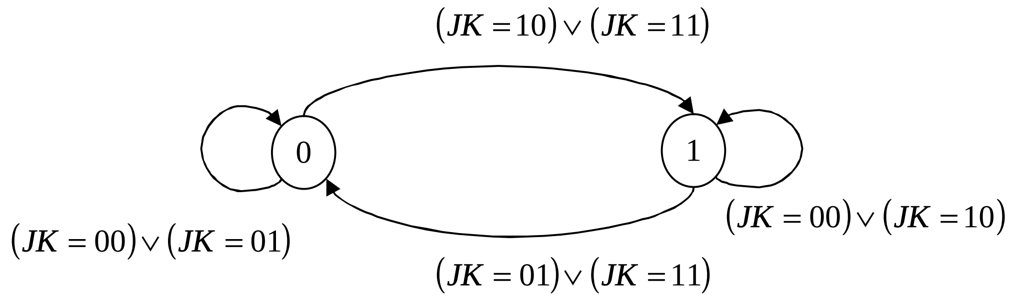


Рисунок 3 Представление JK-триггера в виде графа

Используя граф триггера, можно получить табличную форму матрицы переходов. При этом можно упростить условия переходов, используя законы алгебры логики.

$Q(t)$	$Q(t+1)$	Условие перехода	Упрощённое условие
0	0	$\overline{J}\cdot\overline{K}\vee\overline{J}\cdot K$	$\overline{J}\ -$
0	1	$J\cdot\overline{K}\vee J\cdot K$	$J\ -$
1	0	$\overline{J}\cdot K\vee J\cdot K$	$-\ K$
1	1	$\overline{J}\cdot\overline{K}\vee J\cdot\overline{K}$	$-\ \overline{K}$

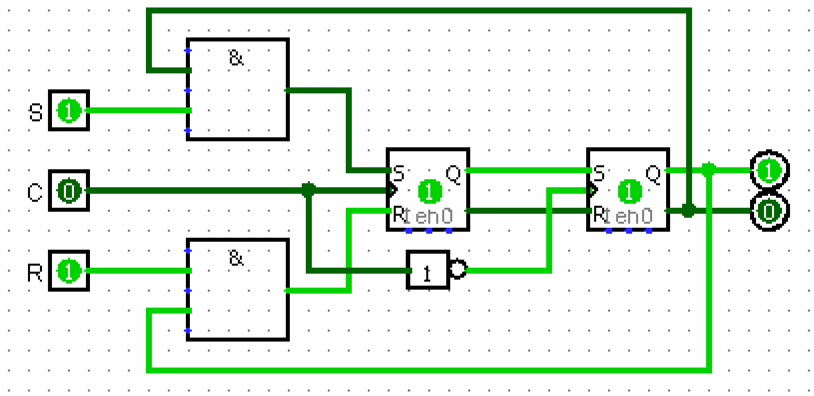
Запись $\overline{J}$ – означает, что на вход J подаётся сигнал 0, а на вход K – безразлично.
Представим таблицу переходов JK-триггера в виде карты Карно:

$Q(t)$		0	1
$J(t)$	$K(t)$		
0	0	0	1
0	1	0	0
1	1	1	0
1	0	1	1

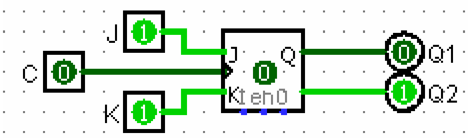
Получим выражение для функции переходов JK-триггера:

$$Q(t+1)=J(t)\overline{Q(t)}\vee\overline{K(t)}\cdot Q(t).$$

Схематически JK- триггер строится на базе двухтактного RS-триггера за счёт обратной связи.



Обозначение в программе Logisim:



Кроме рассмотренных типов триггеров, существуют и другие: S, R, E-триггеры. Они отличаются от базового RS реакцией на комбинацию 11. При подаче такой комбинации сигналов на входы S и R, S-триггер переходит в состояние Q=1, R-триггер переходит в состояние Q=0, а E-триггер сохраняет своё предыдущее состояние.

Варианты заданий

Варианты заданий для этой лабораторной работы отсутствуют.

Порядок выполнения работы

1. Используя Logisim, построить схемы D, T и JK-триггеров.
2. Построить таблицы состояний (переходов) этих схем.
3. Используя Logisim, проверить работу созданных схем, применяя готовые D, T и JK-триггеры из меню «Память»
4. Построить таблицы переходов этих схем этих устройств и сравнить их с построенными в п. 2. При обнаружении различий проверить схемы.
5. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:
   - схемы D, T и JK-триггеров, построенные в п.1;
   - изображения триггеров (п. 3);
   - таблицы переходов триггеров.

Контрольные вопросы

1. Как построить D-триггер на базе синхронного RS-триггера? Расскажите о его работе.
2. Нарисуйте таблицу состояний (переходов) D-триггера.
3. Нарисуйте граф D-триггера.
4. Нарисуйте табличную форму матрицы переходов D-триггера и получите из неё выражение для функции переходов.
5. Нарисуйте матрицу переходов D-триггера.
6. Как построить асинхронный T-триггер на базе D-триггера и RS-триггера? Расскажите о его работе.
7. Нарисуйте таблицу состояний (переходов) T-триггера.
8. Нарисуйте граф T-триггера.
9. Нарисуйте табличную форму матрицы переходов T-триггера и получите из неё выражение для функции переходов.
10. Нарисуйте матрицу переходов T-триггера.
11. Для чего применяется JK-триггер, чем он отличается от RS-триггера?
12. Нарисуйте таблицу состояний (переходов) JK-триггера.
13. Нарисуйте граф JK-триггера.
14. Нарисуйте табличную форму матрицы переходов JK-триггера с условиями переходов.
15. Нарисуйте табличную форму матрицы переходов JK-триггера и получите из неё выражение для функции переходов.
16. Постройте JK-триггер на базе двухтактного RS-триггера за счёт использования обратной связи.

17. Расскажите об S, R, E-триггерах.

Лабораторная работа №12. Синтез функций активации триггеров

Цель работы: изучить метод синтеза функций активации триггеров.

Теоретическая часть

Кодирование внутренних состояний абстрактного конечного автомата заключается в установлении соответствия между внутренними состояниями автомата и комбинациями состояний элементарных комбинационных схем (автоматов Мура), входящих в синтезируемый последовательный автомат (автомат Мили). В общем случае кодирование может быть произвольным, но на практике выбор того или иного варианта кодирования влияет на сложность функций активации его элементарных автоматов.

Функцией активации (возбуждения) элементарного конечного автомата называется логическое выражение, описывающее зависимость входных сигналов элементарного автомата от состояний элементарного автомата и входных сигналов синтезируемого конечного автомата.

Предполагается, что для всех типов триггеров, используемых при синтезе, функции активации определяются функциями внешних переходов этих автоматов.

Функция внешних переходов элементарного конечного автомата – это логическое выражение, определяющее зависимость состояния элементарного автомата в момент $t+1$ от комбинаций состояний всех элементарных автоматов и входных сигналов синтезируемого конечного автомата в момент времени t.

Рассмотрим этапы кодирования.

Конечный автомат может быть задан несколькими способами (лабораторная работа №8). Пусть в данном случае он задан таблицей состояний (переходов), а также кодами этих состояний и входных воздействий.

x \ q	1	2	3
1	2	3	1
2	2	3	2
3	1	1	2
4	3	1	1

В развёрнутом виде эта таблица выглядит так:

Внешние воздействия x \ Внутренние состояния q	q_1	q_2	q_3
x_1	q_2	q_3	q_1
x_2	q_2	q_3	q_2
x_3	q_1	q_1	q_2
x_4	q_3	q_1	q_1

Вначале необходимо определить число N триггеров синтезируемого конечного автомата по формуле:

$N = [\log_2 m]_{\text{ббц}}$,

где ббц - ближайшее большее целое,

m – число внутренних переменных абстрактного конечного автомата,

N – число триггеров.

В рассматриваемом случае имеется три внутренних переменных q:

$N = [\log_2 3]_{\text{ббц}} = 2$

Обозначим триггеры Q_1 и Q_2.

Теперь установим соответствия между внутренними состояниями конечного автомата и комбинациями состояний триггеров, входящих в синтезируемый автомат.

Определим выходные состояния элементарных автоматов (триггеров Q_1 и Q_2) с учётом того, что у каждого из них имеется прямой и инверсный вход:

$q_1 = \overline{Q_1} \cdot \overline{Q_2} = 00$,

$q_2 = \overline{Q_1} \cdot Q_2 = 01$,

$q_3 = Q_1 \cdot Q_2 = 11$,

$q_4 = Q_1 \cdot \overline{Q_2} = 10$.

В процессе кодирования получилось четыре состояния. Но в заданной выше таблице используется только три внутренних состояния q синтезируемого автомата. Следовательно, из четвёртого состояния автомат будет переходить в неопределённое.

Найдём число входных переменных синтезируемого автомата: $N = [\log_2 4]_{\text{ббц}} = 2$.

Обозначим их z_1 и z_2. Закодируем эти состояния:

$x_1 = \overline{z_1} \cdot \overline{z_2} = 00$,

$x_2 = \overline{z_1} \cdot z_2 = 01$,

$x_3 = z_1 \cdot z_2 = 11$,

$x_4 = z_1 \cdot \overline{z_2} = 10$.

Заменим в заданной выше таблице переходов синтезируемого автомата обозначения состояний q и входных воздействий x их кодами. Например, $q_1 = 00$, $x_2 = 01$ и так далее.

$x \backslash q$	00	01	11	10
00	01	11	00	–
01	01	11	01	–
11	00	00	01	–
10	11	00	00	–

Разделим эту таблицу на две. В первой оставим только те цифры внутренних состояний, которые соответствуют триггеру Q_1 (левые), а во второй – те, которые соответствуют триггеру Q_2 (правые).

Для триггера Q_1:

$x \backslash q$	00	01	11	10
00	0	1	0	–
01	0	1	0	–
11	0	0	0	–
10	1	0	0	–

Для триггера Q_2:

q \ x	00	01	11	10
00	1	1	0	–
01	1	1	1	–
11	0	0	1	–
10	1	0	0	–

Зададимся типом триггера, который будет использован при синтезе конечного автомата. Пусть это RS-триггер. Рассмотрим приведённую выше закодированную таблицу переходов, предназначенную для определения функций внешних переходов входящих в автомат триггеров. Если в её клетках проставить не состояния элементарного автомата, а условия, вызывающие переход в эти состояния, то по такой таблице можно определить функции активации соответствующего триггера. Условия, вызывающие переход, берутся из табличной формы матрицы переходов выбранного типа триггера. В рассматриваемом случае следует воспользоваться соответствующей таблицей RS-триггера:

$Q(t)$	$Q(t+1)$	Упрощённое условие
0	0	$-\overline{S}$
0	1	$\overline{R}\cdot S$
1	0	$R\cdot\overline{S}$
1	1	$\overline{R}-$

Состояние $Q(t)$ отмечено в строке «Внутренние состояния», а состояние $Q(t+1)$ - в клетках таблицы.

Для триггера Q_1:

q \ x	00	01	11	10
00	$-\overline{S_1}$	$\overline{R_1}\cdot S_1$	$R_1\cdot\overline{S_1}$	–
01	$-\overline{S_1}$	$\overline{R_1}\cdot S_1$	$R_1\cdot\overline{S_1}$	–
11	$-\overline{S_1}$	$-\overline{S_1}$	$R_1\cdot\overline{S_1}$	–
10	$\overline{R_1}\cdot S_1$	$-\overline{S_1}$	$R_1\cdot\overline{S_1}$	–

Найдём выражение функции активации для входа R_1 триггера Q_1. Для этого поставим в эту таблицу в выражения состояний $Q(t+1)$ значения $R_1=1$ и $S_1=0$:

$-\overline{S_1}=-\cdot\overline{0}=-\cdot 1=-$ (неопределённость),

$\overline{R_1}\cdot S_1=\overline{1}\cdot 0=0$,

$R_1\cdot\overline{S_1}=1\cdot\overline{0}=1\cdot 1=1$.

Получим:

$x \backslash q$	00	01	11	10
00	–	0	1	–
01	–	0	1	–
11	–	–	1	–
10	0	–	1	–

Используя эту таблицу как карту Карно, получим логическое выражение функции активации для входа R_1 триггера Q_1:

$R_1 = Q_1$.

Совершим те же операции S_1 триггера Q_1, положив $R_1 = 0$ и $S_1 = 1$.

$-\overline{S_1} = -\cdot\bar{1} = 0$,

$\overline{R_1} \cdot S_1 = \bar{0} \cdot 1 = 1$,

$R_1 \cdot \overline{S_1} = 0 \cdot \bar{1} = 0$.

Используем получившуюся таблицу как карту Карно:

$x \backslash q$	00	01	11	10
00	0	1	0	–
01	0	1	0	–
11	0	0	0	–
10	1	0	0	–

$S_1 = \overline{z_1}\,\overline{Q_1}Q_2 \vee z_1\overline{z_2}\,\overline{Q_2}$.

Для триггера Q_2 используем правые части кодов исходной таблицы, подставив условия перехода для RS-триггера:

$x \backslash q$	00	01	11	10
00	$\overline{R} \cdot S$	$\overline{R}-$	$R \cdot \overline{S}$	–
01	$\overline{R} \cdot S$	$\overline{R}-$	$\overline{R}-$	–
11	$-\overline{S}$	$R \cdot \overline{S}$	$\overline{R}-$	–
10	$\overline{R} \cdot S$	$R \cdot \overline{S}$	$R \cdot \overline{S}$	–

Выражение для R_2 ($R_1 = 1$ и $S_1 = 0$):

$x \backslash q$	00	01	11	10
00	0	0	1	–
01	0	0	0	–
11	–	1	0	–
10	0	1	1	–

$R_2 = \overline{z_2}Q_1 \vee z_1\overline{Q_1}Q_2$.

Выражение для S_2 ($R_2 = 0$ и $S_2 = 1$):

q / x	00	01	11	10
00	1	–	0	–
01	1	–	–	–
11	0	0	–	–
10	1	0	0	–

$$S_2 = \bar{z}_1\bar{Q}_1 \vee \bar{z}_2\bar{Q}_2\,.$$

Итак, мы имеем выражения для активации входов триггеров Q_1 и Q_2:

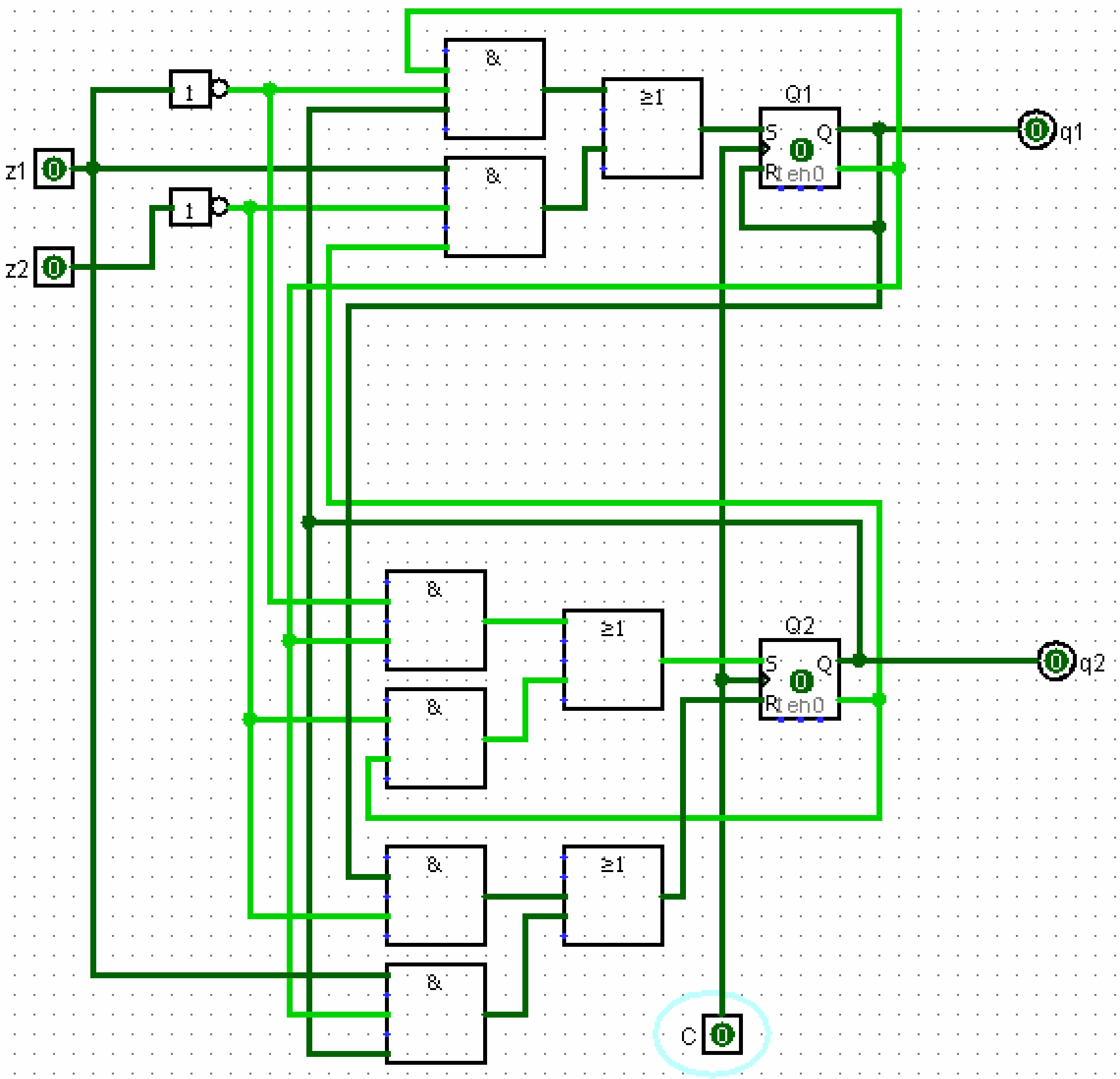
$R_1 = Q_1$,

$S_1 = \bar{z}_1\bar{Q}_1 Q_2 \vee z_1\bar{z}_2\bar{Q}_2$,

$R_2 = \bar{z}_2 Q_1 \vee z_1\bar{Q}_1 Q_2$

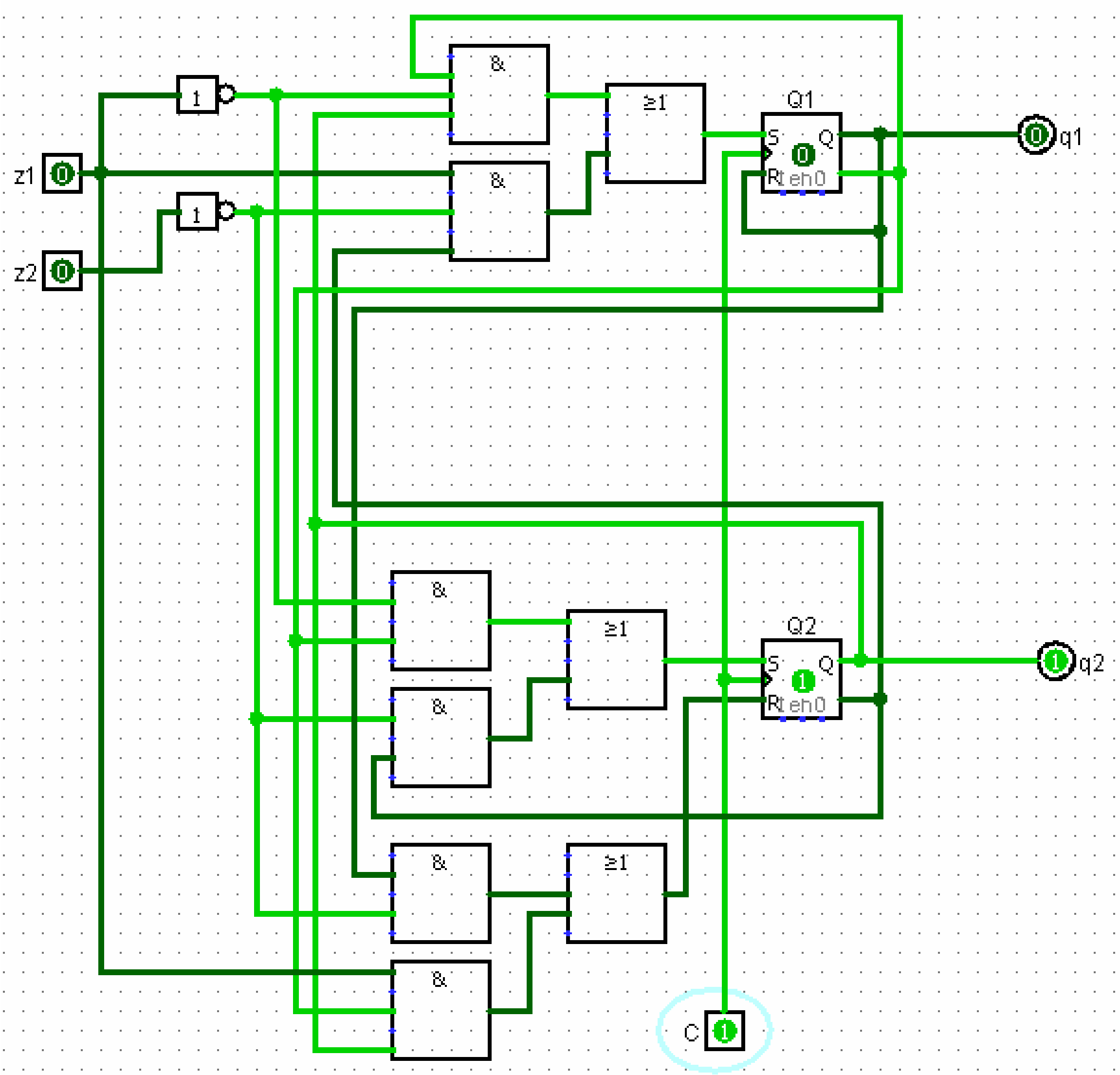
$S_2 = \bar{z}_1\bar{Q}_1 \vee \bar{z}_2\bar{Q}_2$.

Теперь необходимо построить эту схему в Logisim и проверить её работу. Так как триггеры в Logisim синхронные, добавим вход синхронизации (вход C внизу на схеме).



Состояние триггеров будет меняться только при подаче единицы на вход синхронизации. Исходное состояние триггера можно задавать щелчком мыши по его изображению. Для проверки используем исходную таблицу состояний (переходов) и коды состояний.

Согласно таблице переходов, если $Q_1Q_2 = 00$, то при воздействии $z_1z_2 = 00$ автомат должен перейти в состояние $Q_1Q_2 = 01$ (логический ноль). Установив триггеры в состояние $Q_1Q_2 = 00$ и входы в состояние $z_1z_2 = 00$ (рисунок выше), подадим сигнал синхронизации $C = 1$. Получим переход автомата в состояние $Q_1Q_2 = 01$:



Аналогично проверяются другие состояния.

Варианты заданий

Вариант 1

x \ q	1	2	3
1	1	1	1
2	1	2	3
3	2	1	1
4	3	3	3
Триггер	*RS*		

Вариант 2

x \ q	1	2	3
1	1	1	2
2	2	1	3
3	2	1	2
4	3	3	2
Триггер	*T*		

Вариант 3

$x \backslash q$	1	2	3
1	1	1	3
2	2	1	3
3	2	1	3
4	3	3	1
Триггер	*JK*		

Вариант 4

$x \backslash q$	1	2	3
1	1	2	1
2	1	2	3
3	2	2	1
4	3	2	3
Триггер	*RS*		

Вариант 5

$x \backslash q$	1	2	3
1	1	2	2
2	1	2	3
3	2	2	2
4	3	2	2
Триггер	*T*		

Вариант 6

$x \backslash q$	1	2	3
1	1	2	3
2	2	1	3
3	2	2	3
4	3	2	1
Триггер	*JK*		

Вариант 7

$x \backslash q$	1	2	3
1	1	3	1
2	2	1	3
3	2	3	1
4	3	1	3
Триггер	*RS*		

Вариант 8

$x \backslash q$	1	2	3
1	1	3	2
2	1	2	3
3	2	3	2
4	3	1	2
Триггер	*T*		

Вариант 9

$x \backslash q$	1	2	3
1	1	3	3
2	1	2	3
3	2	3	3
4	3	1	1
Триггер	*JK*		

Вариант 10

$x \backslash q$	1	2	3
1	1	1	1
2	1	2	3
3	2	1	1
4	3	3	3
Триггер	*T*		

Вариант 11

$x \backslash q$	1	2	3
1	1	1	2
2	2	1	3
3	2	1	2
4	3	3	2
Триггер	*JK*		

Вариант 12

$x \backslash q$	1	2	3
1	1	1	3
2	2	1	3
3	2	1	3
4	3	3	1
Триггер	*RS*		

Вариант 13

q / x	1	2	3
1	1	2	1
2	1	2	3
3	2	2	1
4	3	2	3
Триггер	*T*		

Вариант 14

q / x	1	2	3
1	1	2	2
2	1	2	3
3	2	2	2
4	3	2	2
Триггер	*JK*		

Вариант 15

q / x	1	2	3
1	1	2	3
2	2	1	3
3	2	2	3
4	3	2	1
Триггер	*RS*		

Вариант 16

q / x	1	2	3
1	1	3	1
2	2	1	3
3	2	3	1
4	3	1	3
Триггер	*T*		

Вариант 17

q / x	1	2	3
1	1	3	2
2	1	2	3
3	2	3	2
4	3	1	2
Триггер	*JK*		

Вариант 18

q / x	1	2	3
1	1	3	3
2	1	2	3
3	2	3	3
4	3	1	1
Триггер	*RS*		

Вариант 19

q / x	1	2	3
1	3	1	2
2	2	3	2
3	1	3	2
4	1	2	3
Триггер	*T*		

Вариант 20

q / x	1	2	3
1	3	1	3
2	2	3	1
3	1	3	1
4	2	1	3
Триггер	*JK*		

Порядок выполнения работы

1. Используя свой вариант задания, построить развёрнутую таблицу состояний (переходов) конечного автомата.
2. Определить число триггеров автомата.
3. Определить число внутренних и входных переменных, закодировать состояния этих переменных.
4. Построить закодированную таблицу состояний (переходов) автомата.
5. Построить закодированные таблицы состояний для каждого триггера.
6. Согласно типу триггера, определённому в варианте задания, построить таблицы активаций триггеров синтезируемого автомата.

7. Используя таблицы активаций, получить логические выражения функций активаций триггеров синтезируемого автомата.
8. Используя Logisim, построить синтезированный автомат.
9. Используя Logisim, проверить работу автомата, построив таблицу его состояний.
10. Проверить совпадение полученной таблицы с вариантом задания. При несовпадении проверить синтезированные функции активации.
11. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:
    - вариант задания на лабораторную работу;
    - расчёт числа триггеров автомата;
    - коды внутренних и входных переменных;
    - выбор типа триггера и его таблицу переходов;
    - закодированную таблицу состояний;
    - закодированные таблицы состояний для триггеров;
    - таблицы активации триггеров;
    - логические выражения функций активации;
    - схему синтезированного автомата;
    - таблицу состояний полученного автомата, построенную в Logisim.

Контрольные вопросы

1. В чём заключается кодирование внутренних состояний автомата?
2. Что такое функция активации?
3. Что такое функция внешних переходов?
4. Как определяется число триггеров в синтезируемом автомате?
5. Как кодируется внутренние состояния автомата?
6. Как определяется число внешних переменных автомата и как кодируются их состояния?
7. Как составляется закодированная таблица переходов для автомата и для каждого триггера?
8. Как получить таблицы активации триггеров синтезируемого автомата?
9. Как из таблицы активаций получить функции активации для различных входов триггера?
10. Синтезируйте абстрактный автомат по таблице переходов, заданной преподавателем.