

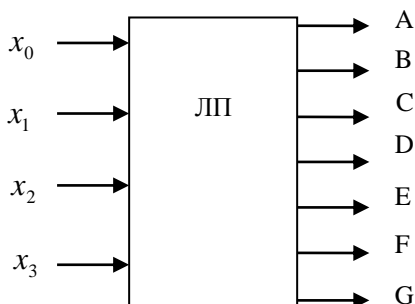
ФГБОУ ВО «Воронежский государственный
технический университет»

Кафедра автоматизированных и вычислительных систем

№101-2017

**ПРАКТИЧЕСКИЕ ЗАДАЧИ ТЕОРИИ
АВТОМАТОВ**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению лабораторных работ № 1-7
по дисциплине «Теория автоматов»
для студентов направления 09.03.01
«Информатика и вычислительная техника»,
профиля «Вычислительные машины, комплексы, системы
и сети» очной и заочной форм обучения



Воронеж 2017

Составители: канд. техн. наук Ю.С. Акинина,
канд. техн. наук С.В. Тюрин

УДК 681.518.52
ББК 3815

Практические задачи теории автоматов: методические указания к выполнению лабораторных работ № 1-7 по дисциплине «Теория автоматов» для студентов направления 09.03.01 «Информатика и вычислительная техника», профиль «Вычислительные машины, комплексы, системы и сети» очной и заочной форм обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост. Ю.С. Акинина, С.В. Тюрин. Воронеж, 2017. 57 с.

Методические указания содержат краткие теоретические сведения и задания для получения первичных навыков по практическому решению задач логического проектирования достаточно простых узлов цифровой вычислительной техники.

Предназначены для студентов третьего курса.

Табл. 19. Ил. 42. Библиогр.: 7 назв.

Рецензент канд. техн. наук, доц. А.М. Нужный

Ответственный за выпуск зав. кафедрой д-р техн. наук,
проф. С.Л. Подвальный

Печатается по решению учебно-методического совета
Воронежского государственного технического университета

© ФГБОУ ВО «Воронежский
государственный техниче-
ский университет», 2017

Лабораторная работа №1

МАТЕМАТИЧЕСКИЕ ОСНОВЫ АЛГЕБРЫ ЛОГИКИ

ЦЕЛЬ: изучение основных понятий и определений алгебры логики и получение практических навыков формального преобразования аналитически заданных логических функций.

Формальное определение алгебры логики

Булевой алгеброй является система, состоящая из множества $B = \{a, b, c, \dots\}$ и двух типов операторов «+» и «*» (логическая сумма и логическое произведение), для которых справедливы следующие пять соотношений:

$$a + b = b + a, \quad a * b = b * a \quad \text{для любых } a, b \in B$$

(коммутативность); (1)

$$a + (b * c) = (a + b) * (a + c), \quad a * (b + c) = a * b + a * c$$

для любых $a, b, c \in B$

(дистрибутивность); (2)

найдутся, $1 \in B$ и $0 \in B$ такие, что $a + 0 = a$, $a * 1 = a$
для любого $a \in B$ (единичные элементы); (3)

найдется, $\bar{a} \in B$ такой, что $a + \bar{a} = 1$, $a * \bar{a} = 0$ для
любого $a \in B$ (дополнение); (4)

$$a + b + c + \dots = \max(a, b, c, \dots);$$
$$a * b * c * \dots = \min(a, b, c, \dots);$$

(5)

с учетом того, что $1 > 0$.

В алгебре логики определены отношение эквивалентности (\Rightarrow) и три логические операции [1]: дизъюнкция (логиче-

ское сложение, операция ИЛИ); конъюнкция (логическое умножение, операция И); отрицание (инверсия, операция НЕ).

Аксиомы алгебры логики

$$\begin{cases} x = 0, \text{ если } x \neq 1 \\ x = 1, \text{ если } x \neq 0 \end{cases} \quad (6)$$

Аксиома (6) утверждает, что значения логических переменных и функций могут принимать всего два взаимоисключающих значения. Данную аксиому иногда называют "закон исключенного третьего". Следует помнить, что "1" и "0" - это не количественные характеристики, а качественные. Однако принимается, что $1 > 0$.

$$c = a + b = \max(a, b) = \begin{cases} 0 + 0 = 0 \\ 0 + 1 = 1 \\ 1 + 0 = 1 \\ 1 + 1 = 1 \end{cases} \quad (7)$$

Аксиома (7) определяет операцию дизъюнкции (логического сложения). Аксиома справедлива для любого числа слагаемых.

$$c = a * b = \min(a, b) = \begin{cases} 0 * 0 = 0 \\ 0 * 1 = 0 \\ 1 * 0 = 0 \\ 1 * 1 = 1 \end{cases} \quad (8)$$

Аксиома (8) определяет операцию конъюнкции (логического умножения). Аксиома справедлива для любого числа сомножителей.

$$c = \bar{a} = \begin{cases} 0, & \text{при } a = 1; \\ 1, & \text{при } a = 0. \end{cases} \quad (9)$$

Аксиома (9) определяет операцию логического отрицания (инверсию).

В алгебре логики действует закон двойственности, который определяет соотношение между логическими операциями «ИЛИ», «И» и «НЕ». Суть этого закона такова: если в уравнениях все переменные поменять на инверсные (противоположные) значения и заменить знак «ИЛИ» на знак «И» и наоборот, то получим эквивалентные уравнения.

Теоремы алгебры логики

$$x + 0 = x \quad (10) \qquad x * 0 = 0 \quad (15)$$

$$x + 1 = 1 \quad (11) \qquad x * 1 = x \quad (16)$$

$$x + x = x \quad (12) \qquad x * x = x \quad (17)$$

$$x + \bar{x} = 1 \quad (13) \qquad x * \bar{x} = 0 \quad (18)$$

$$\bar{\bar{x}} = x \quad (14) \qquad \bar{\bar{\bar{x}}} = \bar{x} \quad (19)$$

Теоремы (10 - 19) легко доказываются с помощью аксиом (6-9) и метода подстановки возможных значений аргументов x .

Законы алгебры логики

Таблица 1

Относительно логического сложения:	Относительно логического умножения:
<i>Коммутативный закон (переместительный закон)</i>	
$x + y = y + x, \quad (20 \text{ а})$	$x * y = y * x \quad (20 \text{ б})$
<i>Ассоциативный закон (сочетательный закон)</i>	
$x + y + z = y + (x + z) = (y + x) + z \quad (21 \text{ а})$	$x * y * z = y * (x * z) = (y * x) * z \quad (21 \text{ б})$
<i>Дистрибутивный закон (распределительный закон)</i>	
$x + (y * z) = (x + y) * (x + z) \quad (22 \text{ а})$	$x * (y + z) = (x * y) + (x * z) \quad (22 \text{ б})$

<i>Законы (правила) де Моргана</i>	
Законы де Моргана называют также законами двойственности	
$\overline{x_1 x_2 \dots x_n} = \overline{x_1} + \overline{x_2} + \dots + \overline{x_n}$ (23 а)	$\overline{x_1 + x_2 + \dots + x_n} = \overline{x_1} * \overline{x_2} * \dots * \overline{x_n}$ (23 б)

Как и в обычной алгебре в алгебре логики определен порядок выполнения логических операций: первой выполняется операция инверсии; второй - конъюнкции; третьей - дизъюнкции.

Табличная форма представления логических функций

Любую логическую функцию можно представить в *табличной форме (таблицей истинности или картой Карно)*. Рассмотрим принцип построения таблиц истинности.

Таблица истинности (ТИ) — представление логической функции в виде таблицы. Таблица истинности содержит $(n + k)$ столбцов и 2^n строк, где n — число логических переменных, а k - число логических функций от n переменных.

В таблице истинности перечисляются (записываются) все возможные входные наборы логических переменных и указываются значения логических функций на каждом из входных наборов логических переменных. Для того, чтобы правильно перечислить все возможные входные наборы логических переменных, можно придерживаться следующего правила заполнения таблицы истинности: все логические переменные, определяющие логическую функцию, нумеруются (начиная справа) целыми числами j от 0 до $n-1$. Каждой логической переменной в таблице истинности, начиная сверху, присваивают чередующиеся группы значений (начиная с 0) через 2^j элементов в группе, где j — номер логической переменной (табл. 2).

Пример заполнения таблицы истинности для одной логической функции n аргументов

2^{n-1}	2^{n-2}	2^j	2^2	2^1	2^0	
x_{n-1}	x_{n-2}	x_j	x_2	x_1	x_0	y
a	b	z	p	q	
			0	0	0	
			0	0	1	
			0	1	0	
			0	1	1	
			1	0	0	
			1	0	1	
			1	1	0	
			1	1	1	
...	

Иными словами, для правильного заполнения таблицы истинности можно представить каждый входной набор значений логических переменных двоичным n -разрядным числом, и записать все возможные двоичные числа в порядке возрастания их значений, начиная со значения, равного нулю.

В правый крайний столбец (столбцы) таблицы истинности записываются значения логической функции.

В табл. 3 представлена в табличной форме логическая функция $f = \overline{x_0} + x_1$.

Таблица 3

Таблица истинности функции f

x_1	x_0	f
0	0	1
0	1	0
1	0	0
1	1	0

ПРОЕКТИРОВАНИЕ ЛОГИЧЕСКИХ СХЕМ С ПОМОЩЬЮ ФУНКЦИЙ АЛГЕБРЫ ЛОГИКИ

Для формального синтеза комбинационных автоматов закон их функционирования должен быть описан каким-либо аналитическим способом. К таким аналитическим способам относят описание работы комбинационного автомата в виде совершенной дизъюнктивной нормальной формы (СДНФ) и совершенной конъюнктивной нормальной формы (СКНФ). Каждая логическая функция имеет одну СДНФ и одну СКНФ.

СДНФ логической функции $f(x_1, \dots, x_n)$ называется дизъюнкция элементарных конъюнкций максимального ранга (n), соответствующих наборам из всех входных переменных, на которых значение логической функции равно 1.

СКНФ логической функции $f(x_1, \dots, x_n)$ называется конъюнкция элементарных дизъюнкций максимального ранга (n), соответствующих наборам из всех входных переменных, на которых значение логической функции равно 0.

Ранг элементарной конъюнкции/дизъюнкции – это число входных переменных в элементарной конъюнкции/дизъюнкции.

Представление логических функций в виде дизъюнктивных и конъюнктивных нормальных форм называют структурными формулами, так как на их основе осуществляется формальный синтез структуры комбинационного автомата.

Логической схемой называется совокупность *логических электронных элементов* (рис. 1), соединенных между собой таким образом, чтобы выполнялся заданный закон функционирования схемы, иначе говоря, выполнялась заданная логическая функция.

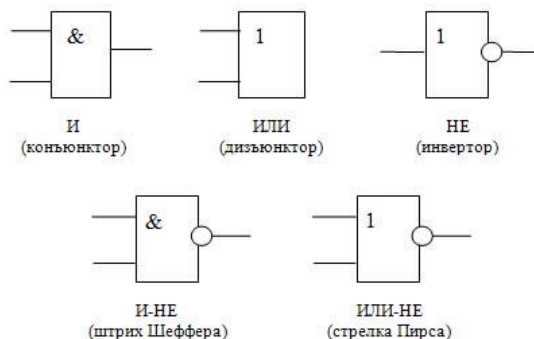


Рис. 1. Условное графическое обозначение основных логических элементов

ХОД ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ:

1. Используя законы, аксиомы и теоремы алгебры логики преобразовать аналитические выражения, задающие логические функции, так, чтобы в преобразованное аналитическое выражение входили только прямые и инверсные переменные (т.е., чтобы в аналитическом выражении отсутствовали “длинные” инверсии) (по варианту):

1. $f_1(x_1, x_2, x_3) = \overline{\overline{(x_1 x_2 + x_1 x_2)}} * x_3$;
2. $f_2(x_1, x_2, x_3) = \overline{\overline{x_1 + (x_1 + x_2)}} x_3$;
3. $f_3(x_1, x_2, x_3) = \overline{\overline{(x_1 x_2 + x_2 x_3)}} * \overline{\overline{x_1 x_2}}$;
4. $f_4(x_1, x_2, x_3) = \overline{\overline{\overline{(x_1 x_2 x_3 * x_1)} + (x_1 + x_3)}}$;
5. $f_5(x_1, x_2, x_3) = \overline{\overline{(x_1 + x_2) + x_2 x_3}} * \overline{\overline{x_1 x_2}}$;
6. $f_6(x_1, x_2, x_3) = \overline{\overline{(x_1 x_2 + x_1 x_2)}} * \overline{\overline{x_3}}$;
7. $f_7(x_1, x_2, x_3) = \overline{\overline{(x_1 + x_2)} x_3 + x_1 x_2}$;
8. $f_8(x_1, x_2, x_3) = \overline{\overline{(x_1 + x_2)} x_3 + x_1 x_2}$;
9. $f_9(x_1, x_2, x_3) = \overline{\overline{\overline{(x_1 + x_2 + x_2 x_3)}}} * \overline{\overline{x_1}} * \overline{\overline{x_1 x_2}}$;

$$10. f_{10}(x_1, x_2, x_3) = \overline{(x_1 + x_2)} * x_3 + x_1 x_2 x_3 (x_1 + x_3);$$

2. Построить таблицу истинности (ТИ) упрощенной логической функции.

3. Используя программу схемотехнического моделирования Electronics Workbench получить таблицу истинности исходной логической функции и сравнить ее с ТИ упрощенной логической функции.

4. Представить логическую функцию из табл. 4 в виде СДНФ и СКНФ (по варианту).

Таблица 4

x ₂	x ₁	x ₀	F ₀	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	F ₇	F ₈	F ₉
0	0	0	1	1	0	0	1	1	0	1	1	1
0	0	1	0	1	1	0	1	1	1	1	0	0
0	1	0	0	0	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	1	1	1	1
1	0	0	0	0	0	0	1	1	0	0	0	0
1	0	1	1	0	1	0	1	0	1	0	1	0
1	1	0	0	1	0	1	0	1	0	1	1	0
1	1	1	1	0	1	0	1	1	1	0	1	1

5. С помощью программы схемотехнического моделирования Electronics Workbench (EWB) построить логические схемы логических функций, заданных в СДНФ и СКНФ, используя логические элементы И, ИЛИ, НЕ, и проверить закон их функционирования с помощью логического конвертера программы EWB.

Лабораторная работа №2

СИНТЕЗ И МОДЕЛИРОВАНИЕ ПРЕОБРАЗОВАТЕЛЯ ДВОИЧНО-ДЕСЯТИЧНОГО КОДА В СЕМИСЕГМЕНТНЫЙ КОД

ЦЕЛЬ: закрепление теоретических положений теории автоматов и получение практических навыков по формальному

синтезу преобразователя двоично-десятичного кода в семи-
сегментный код.

ЗАДАНИЕ: требуется синтезировать преобразователь двоично-десятичного кода в семисегментный код, на вход которого поступают входные сигналы x_0, x_1, x_2, x_3 и который формирует семь выходных сигналов А, В, С, D, E, F, G (рис. 2), которые управляют свечением семисегментного индикатора (рис. 3):

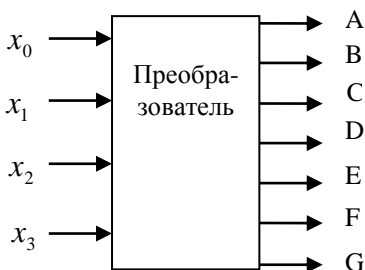


Рис. 2. Структурная схема преобразователя двоично-десятичного кода в семисегментный код на уровне черного ящика

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Семисегментные индикаторы относятся к выходным устройствам для индикации цифровой информации. Семь сегментов индикатора обозначены на рис. 3 а буквами А, В, С, D, E, F, G. Каждый из сегментов может светиться или не светиться. В зависимости от того, какая группа сегментов светится, можно получить изображение десятичных цифр 0, 1, 2, ..., 9 (рис. 3 б). Исходными сигналами, представляющими десятичные цифры, являются обычно двоично-десятичные коды. Поэтому для обеспечения функционирования семисегментного индикатора необходимо преобразовать двоично-десятичный код в семисегментный код, представляющий собой семирядное слово, составленное из логических нулей и единиц [2].



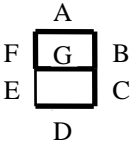
Рис. 3. Обозначение сегментов (а) и цифры, высвечиваемые на семисегментном индикаторе (б)

1. Построение таблицы истинности преобразователя.

Для того, чтобы построить схему преобразователя, необходимо составить таблицу истинности (табл. 5), которая описывает работу данного преобразователя. Для составления таблицы истинности будем использовать рис. 3. Предполагается, что активным уровнем, вызывающим свечение элементов индикатора, является логическая единица.

Таблица 5

i	Входы				Индикаторы	Выходы						
	x_3	x_2	x_1	x_0		A	B	C	D	E	F	G
0	0	0	0	0		1	1	1	1	1	1	0
1	0	0	0	1		0	1	1	0	0	0	0
2	0	0	1	0								

					...							
9	1	0	0	1								

Значения выходных сигналов A, B, C, D, E, F, G заполняются в соответствии со свечением определенных сегментов индикатора, отображающего десятичные цифры 0, 1, 2, ..., 9. Если сегмент индикатора, отображающий ту или иную цифру должен светиться, то выходной сигнал равен 1, если нет – то 0. Например, при отображении цифры 1 светятся только два сегмента B и C, следовательно B=1, C=1, а A, D, E, F, G равны 0.

При составлении ТИ необходимо учитывать, что наборы $i = \overline{10,15}$ никогда не используются и поэтому выходные сигналы на этих наборах могут принимать любые значения.

2. Преобразование части схемы в подсхему.

Electronics Workbench позволяет объединять участки схемы в подсхемы (субблоки). Для этого необходимо выделить участок схемы, который нужно объединить в подсхему. Выделяемая часть схемы должна быть расположена таким образом, чтобы в выделенную область не попали не относящиеся к ней проводники и компоненты. После выделения нужно выбрать пункт Create Subcircuit меню Circuit. В результате выполнения команды открывается диалоговое окно (рис. 4), в строке Name которого вводится имя подсхемы (без пробела), после чего возможны следующие варианты: Copy from Circuit — подсхема копируется с указанным названием в библиотеку Custom без внесения изменений в исходную схему; Move from Circuit — выделенная часть вырезается из общей схемы и в виде подсхемы с присвоенным ей именем копируется в библиотеку Custom; Replace in Circuit — выделенная часть заменяется в

исходной схеме подсхемой с присвоенным ей именем с одновременным копированием ее в библиотеку Custom [3]. Для выполнения лабораторной работы необходимо выбрать пункт Replace in Circuit.

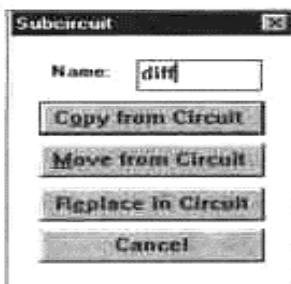


Рис. 4. Диалоговое окно Create Subcircuit

3. Работа с подсхемой.

Для просмотра или редактирования подсхемы нужно дважды щелкнуть мышью по ее значку. Редактирование подсхемы производится по общим правилам редактирования схем. При создании дополнительного вывода подсхемы необходимо из соответствующей точки подсхемы курсором мыши протянуть проводник к краю окна подсхемы до появления не закрашенной прямоугольной контактной площадки, после чего отпустить левую кнопку мыши. Для удаления вывода необходимо курсором мыши ухватиться за его прямоугольную площадку у края окна подсхемы и вынести ее за пределы окна.

4. Проверка правильности функционирования синтезированного преобразователя двоичного кода в семисегментный код.

Для проверки правильности функционирования преобразователя необходимо последовательно подавать на его входы наборы значений входных переменных. Для этого входы схемы необходимо подключить к генератору слов (рис. 5), представленному в разделе Instruments. Самый крайний правый выход генератора слов подключается к логической пере-

менной (x_0), отвечающей за младший разряд входного набора значений логических переменных. Далее в соответствии с рядами подключаются входы x_1, x_2, x_3 .



Рис. 5. Уменьшенное изображение генератора слов

Щелкнув два раза левой кнопкой мыши по изображению генератора слов в расширенном изображении генератора слов найти вкладку РАТТЕМ, выбрать пункт UP COUNTER (счетчик), затем нажать кнопку АССЕРТ. Далее выбрать пошаговый режим работы, нажав кнопку STEP.

Выход преобразователя необходимо подключить к семисегментному индикатору (рис. 6), расположенному в разделе Indicators. Это позволит провести проверку правильности работы синтезированной схемы преобразователя. Каждый из семи выводов индикатора управляет соответствующим сегментом А, В, С, D, Е, F, G. При поступлении на его входы сигналов А, В, С, D, Е, F, G индикатор будет отображать десятичные числа от 0 до 9, в соответствии с табл.5.



Рис. 6. Семисегментный индикатор

После подключения входов и выходов схемы необходимо включить режим моделирования работы преобразователя, нажав на кнопку “1”, расположенную в верхней правой части окна программы ЕWВ.

ХОД ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ:

1. Составить таблицу истинности работы преобразователя двоично-десятичного кода в семисегментный код.
2. Произвести доопределение таблицы истинности преобразователя и минимизировать логические функции А, В, С, D, E, F, G с помощью карт Карно или программы схемотехнического моделирования EWB.
3. Получить структурные формулы, задающие закон функционирования преобразователя в виде дизъюнктивной нормальной формы (ДНФ) или в конъюнктивной нормальной форме (КНФ) (выбрать более выгодную для последующей реализации структурную формулу).
4. По полученным формулам построить логические схемы в пакете EWB, формирующие выходные сигналы А, В, С, D, E, F, G. Каждую из схем оформить в виде подсхемы.
5. Из полученных подсхем, реализующих выходные сигналы А, В, С, D, E, F, G, построить схему логического преобразователя.
6. Проверить правильность работы преобразователя двоично-десятичного кода в семисегментный код с помощью программы схемотехнического моделирования EWB.

Лабораторная работа №3

ПОНЯТИЕ ЭЛЕМЕНТНОГО БАЗИСА И ИХ ОСНОВНЫЕ РАЗНОВИДНОСТИ.

РЕАЛИЗАЦИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ В БАЗИСАХ И-НЕ, ИЛИ-НЕ ПО СКНФ И СДНФ

ЦЕЛЬ: изучение понятий функционально полных базисов и их функциональной полноты, получение практических навыков реализации логических функций в различных функциональных базисах.

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Любую логическую функцию в алгебре логики можно представить в виде СДНФ и СКНФ, используя три элементарные логические функции: инверсию, конъюнкцию и дизъюнкцию. Данный набор представляет собой полный функциональный набор.

Система функций алгебры логики f_1, f_2, \dots, f_m называется функционально полной, если любая логическая функция от произвольного числа n – аргументов может быть представлена суперпозицией функций f_1, f_2, \dots, f_m .

Система логических функций, обладающая функциональной полнотой, называется функциональным базисом, а набор элементов, реализующих эти функции, называется элементным базисом.

Минимальным базисом называется такой базис из f_1, f_2, \dots, f_n , для которого удаление хотя бы одной из функций f_i , входящих в этот базис, превращает систему функций f_1, f_2, \dots, f_n в неполную.

Если осуществить преобразования над логическими функциями, заданными СДНФ и СКНФ по правилам де Моргана, взяв от них двойное отрицание, то получим функционально полные базисы, состоящие из одной или двух элементарных функций.

Функциональной полнотой обладают следующие элементные базисы:

1. И, ИЛИ, НЕ (основной базис);
2. И, НЕ;
3. ИЛИ, НЕ;
4. И-НЕ;
5. ИЛИ-НЕ;
6. 1, И, \oplus .

Для того, чтобы реализовать логическую функцию, заданную в ДНФ в базисе И-НЕ необходимо взять двойное отрицание от этой функции и по законам де Моргана заменить все дизъюнкции на конъюнкции, т.е.:

$$\overline{\overline{f(x_1, x_2, \dots, x_n)}} = \overline{f(x_1, x_2, \dots, x_n)}.$$

Для того, чтобы реализовать логическую функцию, заданную в ДНФ в базисе ИЛИ-НЕ необходимо взять четверное отрицание от этой функции и по законам де Моргана заменить все дизъюнкции на конъюнкции, т.е.:

$$\overline{\overline{\overline{\overline{f(x_1, x_2, \dots, x_n)}}}} = f(x_1, x_2, \dots, x_n).$$

Для того, чтобы реализовать логическую функцию, заданную в КНФ в базисе ИЛИ-НЕ необходимо взять двойное отрицание от этой функции и по законам де Моргана заменить все конъюнкции на дизъюнкции, т.е.:

$$\overline{\overline{f(x_1, x_2, \dots, x_n)}} = \overline{f(x_1, x_2, \dots, x_n)}.$$

Для того, чтобы реализовать логическую функцию, заданную в КНФ в базисе И-НЕ необходимо взять четверное отрицание от этой функции и по законам де Моргана заменить все конъюнкции на дизъюнкции, т.е.:

$$\overline{\overline{\overline{\overline{f(x_1, x_2, \dots, x_n)}}}} = f(x_1, x_2, \dots, x_n).$$

ХОД ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ:

1. Представить логическую функцию F_i , $i=0, \dots, 15$ (по варианту) из табл. 6 в виде СДНФ и СКНФ.

Таблица 6

x_2	x_1	x_0	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	1	1	0	0	0	1	0	1	1	1	1	0	0	1	1	0
0	0	1	0	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1
0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	0	1
1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0

Продолжение табл. 6

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

2. Реализовать полученные аналитические формы логических функций F_i , $i=0, \dots, 15$ (СДНФ и СКНФ) в элементных базисах И, ИЛИ, НЕ; ИЛИ-НЕ и И-НЕ и построить их логические схемы (всего 6 схем).

3. Используя программу схемотехнического моделирования Electronics Workbench, проверить правильность функционирования полученных схем в соответствии с табл. 6 (проверить работоспособность схемы можно с помощью переключателей, подключенных к каждому входу схемы или с помощью логического конвертера).

Лабораторная работа №4

СИНТЕЗ ОДНОТАКТНОГО АСИНХРОННОГО RS-ТРИГГЕРА

ЦЕЛЬ: закрепление теоретических основ формального синтеза элементарных автоматов с памятью. Получение практических навыков по реализации простейших триггерных схем в различных элементных базисах, а также проверки работоспособности реализованных триггеров средствами пакета схемотехнического моделирования EWB.

ЗАДАНИЕ: синтезировать одноклапный асинхронный RS-триггер в элементных базисах И, ИЛИ, НЕ; И-НЕ; ИЛИ-НЕ. Исследовать работоспособность синтезированных триггеров средствами пакета схемотехнического моделирования EWB.

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Обобщенная характеристика триггеров

Триггер – это устройство, которое может находиться в одном из двух устойчивых состояний и переходить из одного состояния в другое под воздействием входных сигналов. Триггеры – это элементарные автоматы с памятью, которые являются основой для построения сколь угодно сложных автоматов с памятью. Входы триггеров условно подразделяют на информационные и управляющие.

Рассмотрим триггер на уровне чёрного ящика:

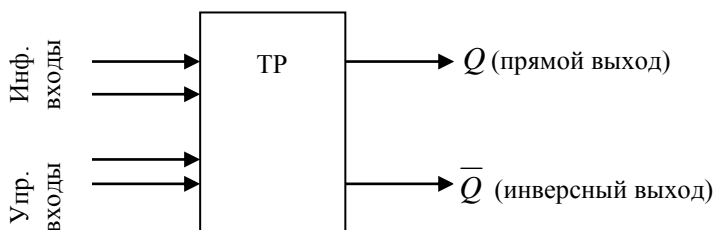


Рис. 7. Представление триггера на уровне черного ящика

где Инф. – информационные входы; Упр. – управляющие входы триггера.

Следует отметить, что на практике часты ситуации, когда информационные входы одновременно являются управляющими, а управляющие - одновременно информационными.

Условились считать, что внутреннее состояние триггера однозначно связано с его выходным сигналом Q (прямой выход). Если на прямом выходе сигнал единичный, то говорят: триггер находится в единице (единичном состоянии). Если на прямом выходе сигнал нулевой, то говорят, что триггер находится в нуле (нулевом состоянии). При этом, если $Q=1$, то $\bar{Q} = 0$ и наоборот.

Информационными входами триггеров являются следующие входы S, R, D, T, J, K .

Самым распространенным управляющим входом является вход C -синхронизация.

Специфической особенностью триггерных схем является то, что они способны запоминать один бит информации (0 или 1) и являются энергозависимыми элементами памяти. Это означает, что при выключении питающего напряжения триггеры «теряют» записанную в них информацию. Поэтому, при включении питающего напряжения состояния триггеров не могут быть однозначно определены.

Все разновидности триггерных схем в своей основе базируются на одноктактных асинхронных RS-триггерах. По этой причине данный тип триггерных схем рассмотрим наиболее подробно.

Синтез одноктактного асинхронного RS-триггера в элементном базисе И, ИЛИ, НЕ

RS-триггеры – триггеры с двумя установочными входами, при подаче активного сигнала на вход R (от англ. Reset – сбрасывать) происходит установка триггера в 0, а на вход S (от англ. Set - устанавливать) – установка триггера в 1.

Для синтеза этого триггера воспользуемся его представлением в виде автомата типа Мура (рис. 8).

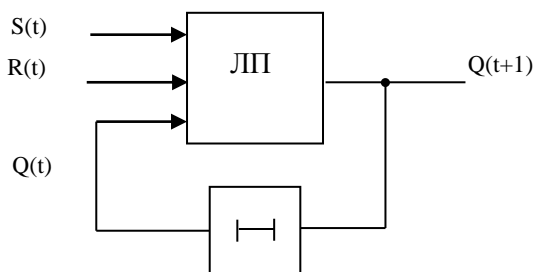


Рис. 8. Представление триггера в виде автомата типа Мура

На рис. 8 показано, что в данном автомате используется тривиальный элемент памяти в виде элемента задержки. Составим таблицу истинности, задающую закон функционирования одноктактного асинхронного RS-триггера (табл. 7).

Таблица 7

R	S	Q(t)	Q(t+1)	
0	0	0	0	}
0	0	1	1	
0	1	0	1	}
0	1	1	1	
1	0	0	0	}
1	0	1	0	
1	1	0	*	}
1	1	1	*	

Режим хранения

Режим записи 1

Режим записи 0

Запрещенные кодовые комбинации

Запрещенными кодовые комбинации называют потому, что при их возникновении триггер становится неуправляемым (состояние триггера не может быть однозначно определено). Поэтому разработчики цифровой аппаратуры не должны допускать возникновения на входах триггера запрещенных кодовых комбинаций. Однако для реализации формального синтеза однотактного асинхронного RS-триггера необходимо доопределить его таблицу истинности (табл. 8). Доопределение таблицы истинности целесообразно провести с использованием карты Карно. Условимся в дальнейшем о следующих обозначениях:

$$S(t)=S; \quad R(t)=R; \quad Q(t)=Q; \quad Q(t+1)=Q^+$$

Таблица 8

Карта Карно для однотактного асинхронного RS-триггера

		RS			
		00	01	11	10
Q	0	0	1	*(1)	0
	1	1	1	*(1)	0

Доопределим исходную таблицу истинности единицами, тогда получим минимизированное уравнение однотактного асинхронного RS-триггера в элементном базисе И, ИЛИ, НЕ:

$$Q^+ = S + \bar{R} * Q \quad (24)$$

На рис. 9 представлена схема электрическая функциональная однотактного асинхронного RS-триггера, реализованного в элементном базисе И, ИЛИ, НЕ в соответствии с табл.7.

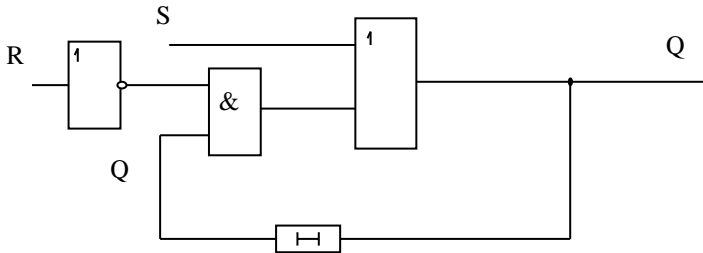




Рис. 9. Схема однотактного асинхронного RS-триггера, реализованного в элементном базисе И, ИЛИ, НЕ



ХОД ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ:

1. Реализовать схему однотактного асинхронного RS-триггера, представленную на рис. 9, средствами пакета схемотехнического моделирования EWB.

2. Проверить работоспособность синтезированной схемы в пакете EWB:

а) к каждому из входов триггера R и S подключить светодиод , расположенный в разделе Indicators;

б) к каждому из входов триггера (R или S) подключить перекидной контакт переключателя  (Switch), который находится в разделе Основные компоненты (Basic).

в) к оставшимся контактам переключателя подключить: к одному – логическую 1 ( – источник тока с фиксированным напряжением +5V, который расположен в группе Source), а к другому – логический 0 ( – раздел Basic);

г) к прямому выходу триггерной схемы также следует подключить светодиод;

д) включить режим моделирования работы триггера и проверить правильность его работы в соответствии с табл. 8.

3. По аналогии с синтезом одноклапного асинхронного RS-триггера в элементном базисе И, ИЛИ, НЕ синтезировать одноклапный асинхронный RS-триггер в элементном базисе И-НЕ. Для этого необходимо взять двойное отрицание от уравнения (24). Проверить работоспособность синтезированной схемы в соответствии с п. 2.

4. Используя табл. 8 синтезировать одноклапный асинхронный RS-триггер в элементном базисе ИЛИ-НЕ. Для перехода в базис ИЛИ-НЕ необходимо доопределить карту Карно нулевыми значениями и получить минимизированное уравнение одноклапного асинхронного RS-триггера в виде конъюнктивной нормальной формы (КНФ). Затем необходимо взять двойное отрицание от полученного уравнения. Проверить работоспособность синтезированной схемы в соответствии с п. 2. –

Лабораторная работа №5

ИССЛЕДОВАНИЕ АЛГОРИТМОВ ФУНКЦИОНИРОВАНИЯ ТРИГГЕРНЫХ СХЕМ

ЦЕЛЬ: закрепление теоретических основ формального синтеза элементарных автоматов с памятью. Получение практических навыков по реализации простейших триггерных схем, а также проверки работоспособности реализованных триггеров средствами пакета схемотехнического моделирования EWB.

ЗАДАНИЕ: исследовать работоспособность триггерных схем средствами пакета схемотехнического моделирования

EWB (схемы триггеров представлены на рис. 10, 14, 17, 19, 22, 24, 27, 29).

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Синхронный одноклапный RS-триггер

Недостатком одноклапного асинхронного RS-триггера является то, что на его основе трудно реализовать надежно работающий блок памяти, содержащий большое количество триггеров данного типа, так как между сигналами, поступающими на информационные входы триггеров неизбежно существуют гонки (состязания). Для устранения данного недостатка были разработаны одноклапные синхронные RS-триггеры.

Построим таблицу истинности, задающую закон функционирования одноклапного синхронного RS-триггера (табл. 9).

Таблица 9

C	S	R	Q	Q ⁺
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	*
1	1	1	1	*

}

режим хранения информации

}

режим установки в 0

}

режим установки в 1

}

запрещенные кодовые комбинации

Построим карту Карно для одноклапного синхронного RS- триггера (табл. 10):

Таблица 10

		RQ			
CS		00	01	11	10
00		0	1	1	0
01		0	1	1	0
11		1	1	* (1)	* (1)
10		0	1	0	0

Доопределив исходную таблицу истинности единицами, получаем минимизированное уравнение одноклапного синхронного RS-триггера в элементном базисе И, ИЛИ, НЕ:

$$\begin{aligned}
 Q^+ &= C * S + \bar{R} * Q + \bar{C} * Q = C * S + Q * (\bar{R} + \bar{C}) = \\
 &= C * S + Q * \overline{R * C}
 \end{aligned}
 \tag{25}$$

Преобразуем уравнение (2) к элементному базису И-НЕ. В результате получим соотношение (3):

$$\overline{\overline{C * S + Q * \overline{R * C}}} = \overline{\overline{C * S} * \overline{Q * R * C}}
 \tag{26}$$

На основе соотношения (26) представляется возможность синтезировать одноклапный синхронный RS-триггер в элементном базисе И-НЕ, схема которого представлена на рис.10.

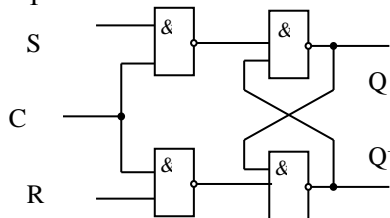


Рис. 10. Схема одноклапного синхронного RS-триггера в элементном базисе И-НЕ

На рис. 11 представлено условное графическое обозначение (УГО) одноклапчатного синхронного RS- триггера:

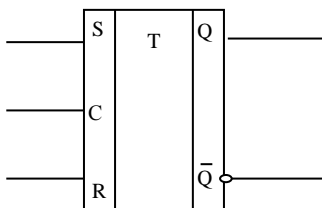


Рис. 11. УГО одноклапчатного синхронного RS- триггера

Синхронный одноклапчатный D-триггер

Одним из способов устранения недостатка RS-триггеров – наличия запрещенных кодовых комбинаций входных сигналов – является построение еще одной разновидности триггерных схем, которая получила название D-триггер. Для управления синхронным одноклапчатым RS-триггером необходимо три входных сигнала (два – информационных; один – управляющий). D-триггеры имеют один информационный D-вход и один управляющий вход – C. В D-триггерах запрещенные комбинации исключаются путем соединения через инвертор S и R входов синхронного RS-триггера так, как показано на рис.12.

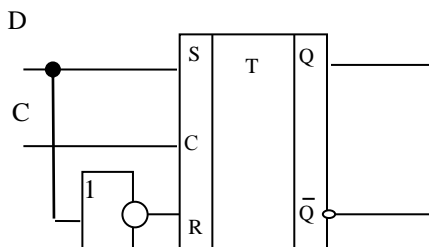


Рис. 12. Схема одноклапчатного синхронного D-триггера с включением инвертора между информационными входами синхронного RS-триггера

При этом на информационные входы RS-триггера никогда не поступают одинаковые сигналы, то есть исключаются не только запрещенные кодовые комбинации ($S=R=1$), но и режим хранения ($S=R=0$), который реализуется путем подачи на вход С нулевого уровня сигнала [2].

Значение сигнала на выходе Q D-триггера в момент времени $(t+1)$ равно значению сигнала на входе D в предыдущий момент времени t . Таким образом, входной сигнал передается на выход с задержкой на один такт, то есть $Q^{t+1} = D^t$. Поэтому D-триггер называют триггером задержки.

Асинхронный D-триггер не применяется, так как его выход просто повторяет входной сигнал после окончания режима состязаний.

Синхронный D-триггер функционирует в соответствии с табл. 11.

Таблица 11

C	D	Q	Q ⁺
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Как видно из этой таблицы, D-триггер способен запоминать по синхросигналу и хранить один бит информации. При отсутствии сигнала синхронизации ($C=0$) состояние триггера остается неизменным (режим хранения).

Во всех рассмотренных ранее схемах синхронных триггеров, триггеры управляются высоким уровнем сигнала синхронизации. Одно из названий таких триггеров, пришедшее из иностранной литературы - триггеры-защёлки. Легче всего объяснить появление этого названия по временной диаграмме, приведенной на рис.13.

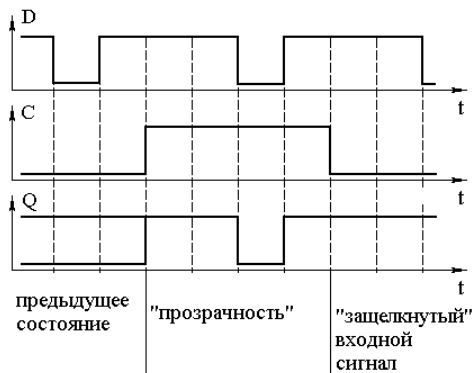


Рис. 13. Временная диаграмма D-триггера (защелки)

По этой временной диаграмме видно, что триггер-защелка хранит данные на выходе только при нулевом уровне на входе синхронизации. Если на входе С синхронного D-триггера постоянно будет присутствовать логическая единица, то принято говорить, что D-триггер становится прозрачным. Этим хотят подчеркнуть то обстоятельство, что сигнал на выходе Q будет повторять значение сигнала на входе D.

Входное напряжение запоминается только в момент изменения уровня напряжения на входе синхронизации С с высокого уровня на низкий уровень. Входные данные как бы "защелкиваются" в этот момент, отсюда и название – триггер-защелка.

Принципиально в этой схеме входной переходной процесс может беспрепятственно проходить на выход триггера. Поэтому там, где это важно, необходимо сокращать длительность импульса синхронизации до минимума. Чтобы преодолеть такое ограничение были разработаны триггеры, работающие по фронту или срезу сигнала синхронизации.

Для синтеза синхронного D-триггера воспользуемся асинхронным RS-триггером. Как видно из табл. 11, сигнал на входе D должен воздействовать на триггер только в том случае, когда имеется единичный уровень сигнала С.

Следовательно, как и в случае синхронного RS-триггера, на входе D-триггера должны быть двухвходовые элементы И, пропускающие сигнал с входа D при C=1.

Рассмотри правую часть табл. 11. Переход триггера в состояние единицы должен происходить, когда на входах воздействуют сигналы C=1 и D=1. Значит, при таком сочетании сигналов на входе \bar{S} асинхронного RS-триггера должен быть нуль, то есть $\bar{S} = \overline{C * D}$. Переход триггера в состояние нуля должен происходить, когда на входах имеется комбинация C=1, D=0. Значит, на входе \bar{R} асинхронного RS-триггера при таком сочетании сигналов также должен быть нуль, то есть $\bar{R} = \overline{C * \bar{D}}$. Поскольку сигнал на входе \bar{S} формируется в момент, когда C=1, и соответствует инверсному значению D, его можно использовать для формирования сигнала \bar{R} , то есть $\bar{R} = \overline{C * \bar{S}}$. Это же рассуждение можно получить и чисто формально с помощью алгебры логики:

$$\bar{R} = \overline{C * \bar{S}} = \overline{C * \overline{C * D}} = \overline{C * (\bar{C} \vee \bar{D})} = \overline{C \bar{C} \vee C \bar{D}} = \overline{C \bar{D}} = \overline{C * D}$$

Следовательно, D-триггер может быть синтезирован на основе асинхронного RS-триггера [2], так как показано на рис. 14.

Уравнение одноклапного асинхронного RS-триггера в элементном базисе И-НЕ имеет вид: $Q^+ = \bar{S} * \bar{\bar{R}} * Q$

Если в этом уравнении произвести соответствующие нашим рассуждениям замены переменных, то есть $\bar{S} = \overline{C * D}$, $\bar{\bar{R}} = \overline{C * \bar{D}}$, то получим уравнение для одноклапного синхронного D-триггера (27).

$$Q^+ = \overline{\overline{C * D} * \overline{C * \bar{D}}} * Q \quad (27)$$

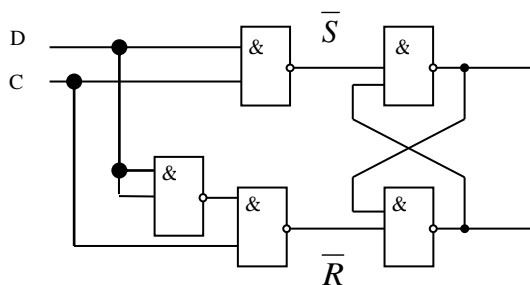


Рис. 14. Синтезированный на основе RS-триггера одноктактный синхронный D-триггер.

На рис. 15 представлено УГО одноктактного синхронного D-триггера.

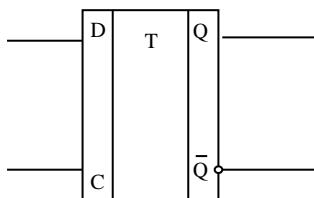


Рис. 15. УГО одноктактного синхронного D-триггера

Принцип построения двухтактного триггера

Рассмотренные разновидности одноктактных триггерных схем усложняют реализацию так называемого режима счета (деления входной тактовой частотой). Но именно данный режим работы триггерных схем является очень важным при построении различных вычислительных устройств. Именно данное обстоятельство потребовало построения двухтактных триггерных схем.

В [4] рассмотрен принцип построения двухтактного триггера, который называют также триггером типа MS (от англ. master, slave, что переводят обычно как «ведущий», «ведомый»). Его упрощенная структурная схема приведена на рис. 16. В схеме имеются два одноступенчатых триггера (ведущий M, ведомый S) и два электронных ключа (Кл1 и Кл2).

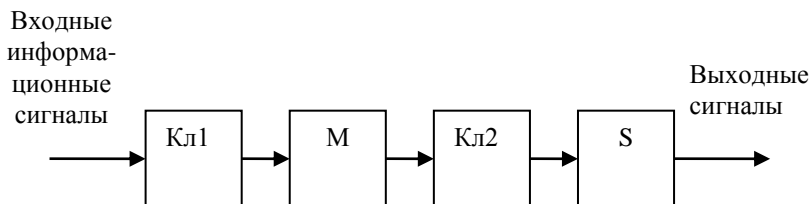


Рис. 16. Структурная схема двухтактного триггера

Двухтактные RS-триггеры

На рис.17 приведен вариант схемы, наиболее доступный для понимания, принципа работы двухтактного синхронного RS-триггера, на рис.18 представлено его УГО.

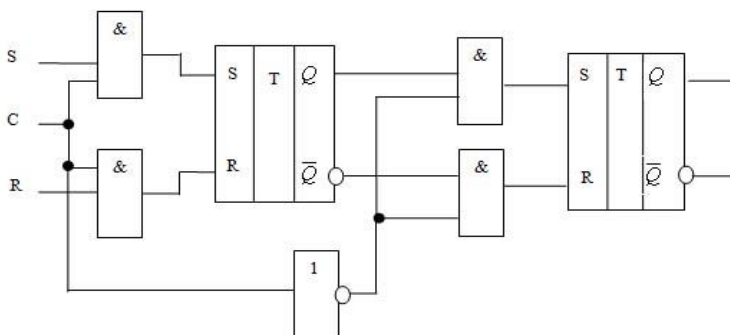


Рис. 17. Синхронный двухтактный RS-триггер

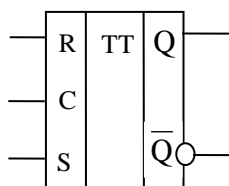


Рис. 18. УГО синхронного двухтактного RS-триггера

В схеме на рис. 17 использованы два асинхронных RS-триггера и инвертор. При $C=1$ производится запись информации в первый триггер, при этом второй триггер заблокирован и находится в режиме хранения, так как на его C-вход через инвертор подается нулевой сигнал. При $C=0$ блокируется первый

триггер, на вход С второго триггера поступает единичный сигнал и информация с выходов первого триггера переписывается во второй триггер.

Двухтактные Т-триггеры

Т-триггер – это счетный триггер. У асинхронного Т-триггера имеется только один вход - Т. После поступления на этот вход импульса, состояние Т-триггера меняется на прямо противоположное. Счётным он называется потому, что он как бы подсчитывает количество импульсов, поступивших на его вход. Жаль только, что считать один такой триггер умеет только до одного. При поступлении второго импульса Т-триггер снова сбрасывается в исходное нулевое состояние.

Булеву функцию, отражающую работу Т-триггера, содержательно можно представить так: триггер инвертирует свое предыдущее состояние при подаче на его вход Т единичного сигнала, то есть $Q^{t+1} = \overline{Q^t}$ при Т=1. Соответствующая таблица переключений Т-триггера представлена в табл. 12.

Таблица 12

T	Q	Q ⁺
0	0	0
0	1	1
1	0	1
1	1	0

} режим хранения

} режим счета

Т-триггеры строятся только на базе двухступенчатых триггеров, подобных рассмотренному ранее RS-триггеру. Использование двух триггеров позволяет избежать самовозбуждения схемы, так как счетные триггеры строятся при помощи схем с обратной связью. Т-триггер можно синтезировать из любого двухступенчатого триггера. Основой для построения двухтактных Т-триггеров, представленных на рис. 19, могут служить двухтактные синхронные RS-триггеры, в которых для изменения состояния триггера на инверсное используются собственные входные сигналы.

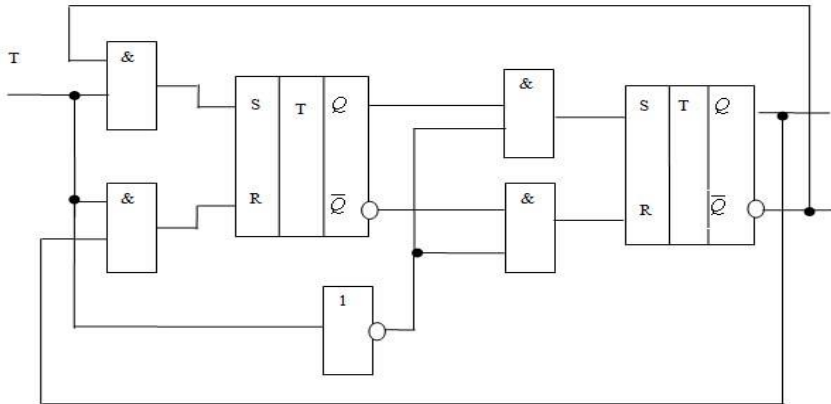


Рис. 19. Синтез двухтактных асинхронных Т-триггеров на основе одноктактных асинхронных RS-триггеров

На рис. 20 представлено УГО двухтактного асинхронного Т-триггера.

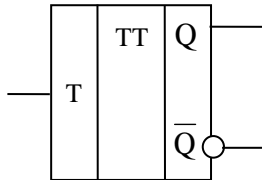


Рис. 20. УГО двухтактного асинхронного Т-триггера

Работу двухтактного асинхронного Т-триггера можно проиллюстрировать следующей временной диаграммой:

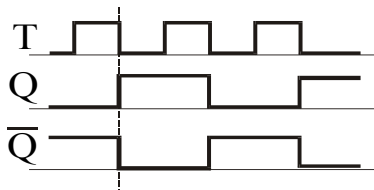


Рис. 21. Временная диаграмма работы асинхронного Т-триггера

На рис. 22 представлена логическая схема двухтактного синхронного Т-триггера на основе одноклапных асинхронных RS-триггеров, а на рис. 23 – УГО.

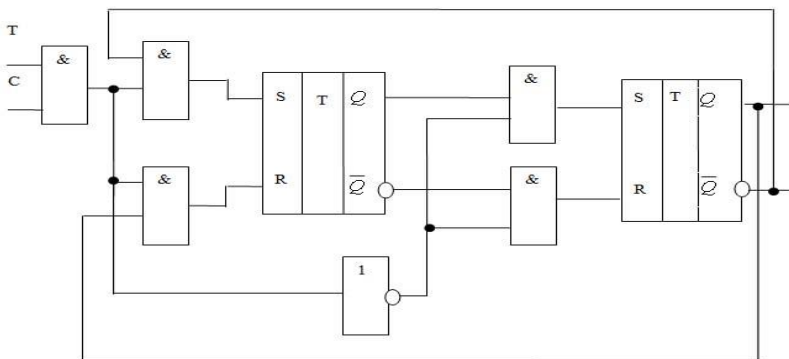


Рис. 22. Схема двухтактного синхронного Т-триггера

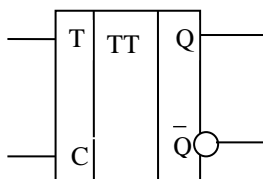


Рис.23. УГО двухтактного синхронного Т-триггера

Двухтактные D-триггеры

На рис. 24 представлена схема двухтактного синхронного D-триггера, а его УГО приведено на рис.26. Рассмотрим принцип работы схемы триггера (рис.24) с помощью временных диаграмм, изображенных на рис. 25.

На этих временных диаграммах обозначение Q' соответствует сигналу на выходе первого триггера. Так как на вход синхронизации второго триггера тактовый сигнал поступает через инвертор, то когда первый триггер находится в режиме хранения, второй триггер пропускает сигнал на выход схемы.

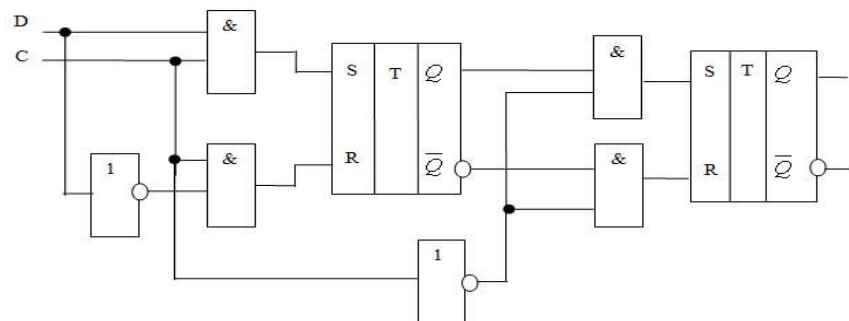


Рис. 24. Синхронный двухтактный D-триггер

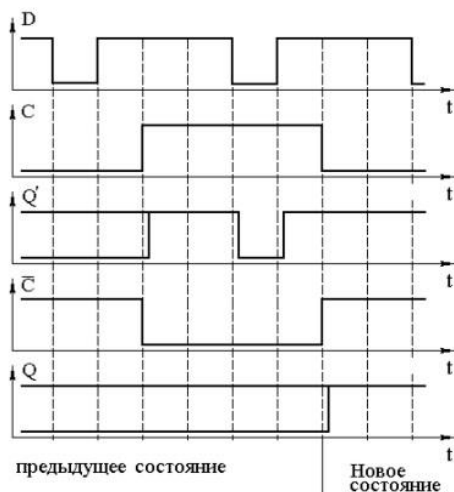


Рис. 25. Временные диаграммы D-триггера

И наоборот, когда первый триггер пропускает сигнал с входа схемы на свой выход, второй триггер находится в режиме хранения [4]. Обратите внимание, что сигнал на выходе всей схемы в целом не зависит от сигнала на входе "D" схемы. Если первый триггер пропускает сигнал данных со своего входа на выход, то второй триггер в это время находится в режиме хранения и поддерживает на выходе предыдущее значение сигнала, то есть сигнал на выходе схемы тоже не может измениться.

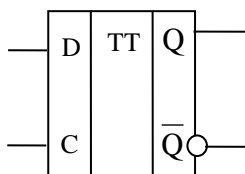


Рис. 26. УГО синхронного двухтактного D-триггера

Двухтактный JK-триггер

Второй способ устранения недостатка RS-триггеров – наличие запрещенной кодовой комбинации входных сигналов – приводит к JK-триггерам, у которых появление на обоих информационных входах логических единиц приводит к изменению состояния триггера. Если в D-триггерах запрещенная для RS-триггеров комбинация не используется, то в JK-триггерах она используется в полезных целях: при этой комбинации входных сигналов JK-триггер работает как T-триггер. В остальном JK-триггер подобен RS-триггеру, причем роль входа S играет вход J, а роль входа R – вход K.

JK-триггеры реализуют в виде триггеров типа MS (двухступенчатых триггеров) или в виде динамических триггеров [2, 4].

Закон функционирования асинхронного JK-триггера можно проиллюстрировать табл. 13.

Таблица 13

J	K	Q	Q ⁺		
0	0	0	0	} режим хранения	} RS - триггер
0	0	1	1		
0	1	0	0	} установка в «0»	
0	1	1	0		
1	0	0	1	} установка в «1»	
1	0	1	1		
1	1	0	1	} $Q^{t+1} = \overline{Q^t}$	} T- триггер
1	1	1	0		

На рис. 27 представлена схема асинхронного двухтактного JK-триггера [2], а на рис. 28 – его условное графическое обозначение.

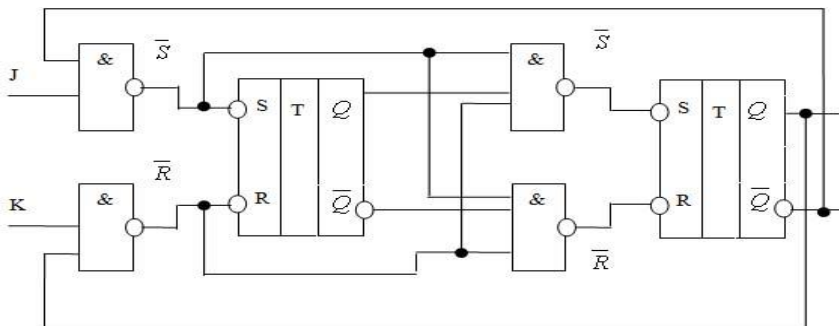


Рис. 27. Схема асинхронного двухтактного JK-триггера

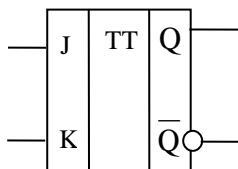


Рис. 28. УГО асинхронного двухтактного JK-триггера

В интегральной схемотехнике распространены синхронные JK-триггеры. Схема такого триггера приведена на рис. 29, а УГО – на рис. 30.

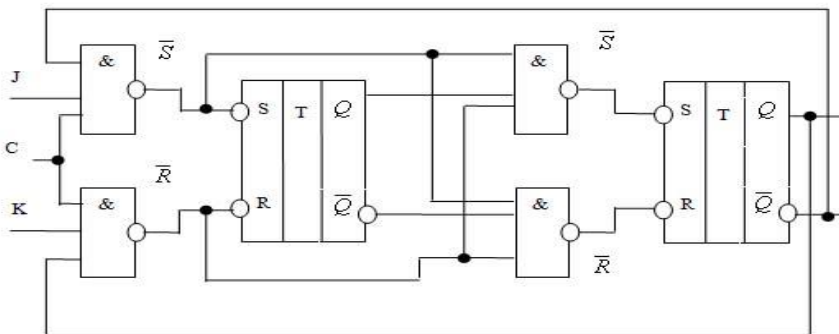


Рис. 29. Схема синхронного двухтактного JK-триггера

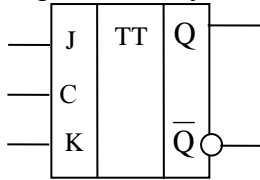


Рис. 30. УГО синхронного двухтактного JK-триггера

Лабораторная работа №6

СИНТЕЗ СИНХРОННОГО АВТОМАТА, ЗАДАННОГО ВРЕМЕННОЙ ДИАГРАММОЙ ЕГО РАБОТЫ

ЦЕЛЬ: закрепление теоретических положений теории автоматов и получение практических навыков по формальному синтезу синхронных автоматов.

ЗАДАНИЕ: требуется синтезировать синхронный автомат (СА), на вход которого поступает сигнал синхронизации S и который формирует четыре выходных сигнала a, b, c, d (рис.31) и реализовать его в элементном базисе И, ИЛИ, НЕ. В качестве элементов памяти использовать комбинированные синхронные двухтактные D – триггеры. Для кодирования состояний автомата использовать тривиальный способ кодирования.

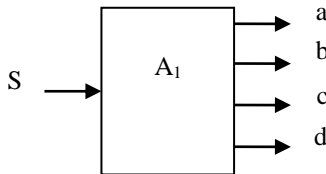


Рис. 31. Структурная схема синхронного автомата на уровне черного ящика.

Для сравнения эффективности способов кодирования состояний автомата синтезировать и реализовать тот же самый синхронный автомат, используя при этом способ соседнего кодирования состояний автомата.

Результатом каждого синтеза должна быть схема электрическая функциональная, работоспособность которой должна быть доказана средствами пакета схемотехнического моделирования EWB. По окончании выполнения лабораторной работы необходимо сформулировать вывод, в котором нужно отразить сравнительный анализ двух полученных электрических функциональных схем.

Временная диаграмма, иллюстрирующая работу автомата, представлена на рис. 27.

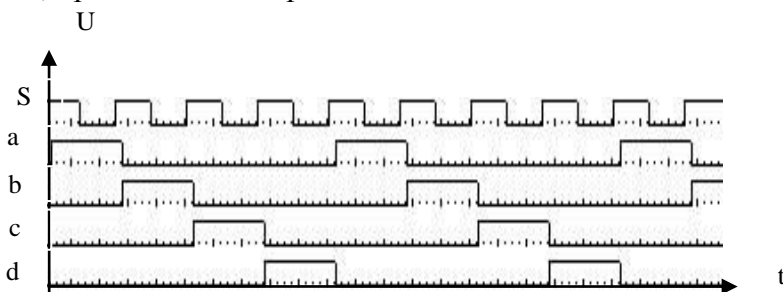


Рис. 32. Временная диаграмма работы автомата

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Работу заданного автомата можно описать следующим образом. По фронту сигнала синхронизации, автомат безусловно переходит из одного состояния в другое. В каждом из состояний автомат формирует один активный сигнал на одном из выходов a,b,c,d, следовательно, автомат имеет четыре состояния, которые циклически повторяются в одном и том же порядке.

Так как выходной сигнал автомата формируется в течение всего времени, пока автомат находится в определенном состоянии, то данный автомат относится к автоматам типа Мура, структура которого представлена на рис. 33, где $g(t)$ - текущее состояние автомата, $g(t+1)$ - следующее состояние автомата.

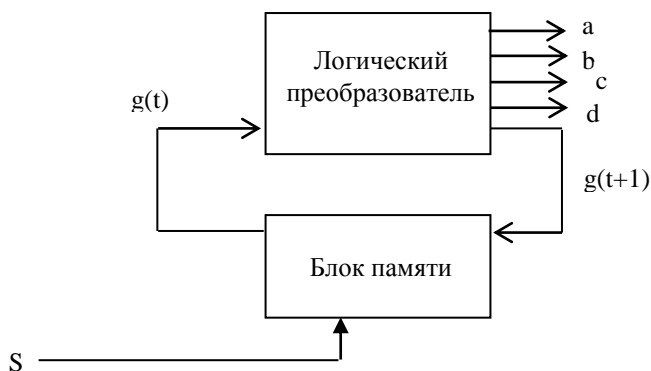


Рис. 33. Структурная модель синхронного автомата типа Мура

ХОД ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ:

Для формального синтеза синхронного автомата может быть использована традиционная методика его синтеза:

1. Работа автомата представляется в виде граф-схемы алгоритма (ГСА).

На рис. 34 представлена граф-схема алгоритма синтезируемого синхронного автомата.

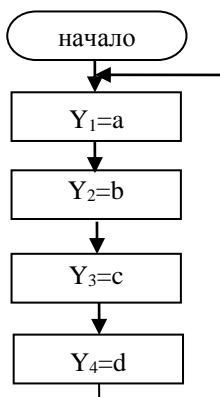


Рис. 34. Граф - схема алгоритма синхронного автомата

2. ГСА размечается для определения числа внутренних состояний автомата.

Переход от алгоритмического описания к автоматному осуществляется путем разметки ГСА в соответствии с выбранной моделью абстрактного автомата.

Правила разметки ГСА при реализации автомата по модели Мура:

- символом начального состояния g_1 отмечаются начальная и конечная операторные вершины;
- все операторные вершины отмечаются различными символами $g_2 \dots g_i \dots g_n$;
- каждая операторная вершина ГСА должна быть отмечена только одним индивидуальным символом g_i .

В результате разметки ГСА по указанным правилам удастся определить множество внутренних состояний автомата: $G = \{ g_1, \dots, g_i, \dots, g_n \}$, а также мощность этого множества, которая равна $|G| = n$.

Для синтеза синхронного автомата необходимо разметить ГСА на рис.34.

3. Структурное кодирование состояний автомата и определение количество триггерных схем, необходимых и достаточных для реализации блока памяти.

В настоящее время самым распространенным способом структурного кодирования является двоичное кодирование. Структурное кодирование проводится в два этапа: определяется количество (β) двоичных разрядов, необходимое и достаточное для двоичного представления некоторого множества абстрактных символов; осуществляется сопоставление каждому отдельному абстрактному символу β - разрядного двоичного кода [5].

В самом простейшем случае величина β находится на основе следующего соотношения:

$$\beta = 1 + \text{int} (\log_2 (\alpha - 1)), \quad (27)$$

где α – мощность множества кодируемых символов абстрактного алфавита G ; $\text{int} (w)$ – целая часть (w).

При тривиальном кодировании исходными данными является мощность символов абстрактного алфавита состояний автомата, полученных в результате разметки ГСА. Соотношением (27), по сути, определяется количество триггерных схем, необходимых для реализации блока памяти автомата. После определения количества двоичных разрядов, необходимых для структурного кодирования, каждому абстрактному символу состояния автомата сопоставляется двоичный код, отличный от всех остальных и без какого – либо регламентирующего правила, вплоть до случайного сопоставления. При соседнем кодировании состояний автомата каждому абстрактному символу состояния автомата сопоставляются двоичные коды, соответствующие кодам ячеек в карте Карно (например, при использовании двухразрядного двоичного кода соответствующие коды будут 00, 01, 11, 10).

Таким образом, по окончании этапа структурного кодирования находятся структурные (двоичные) коды, которые ставятся в соответствие каждому состоянию автомата, что позволяет перейти к этапу детализации блока памяти. Полученные структурные двоичные коды записывают в форме таблицы, например:

Таблица 14

Абстрактные символы		d_2	d_1
	g_1		
	g_2		
	g_3		
	g_4		

где g_1, \dots, g_4 – состояния автомата, а d_1, d_2 – структурный двоичный код, d_1 – младший разряд структурного двоичного кода, а d_2 – старший.

4. Детализация блока памяти.

Конечной целью данного этапа является разработка схемы электрической функциональной блока памяти синтезируемого автомата, который должен быть реализован на заданном типе триггерных схем. По сути, блок памяти представляет собой β триггеров, электрически соединённых определенным образом, или, иначе говоря, представляет одну β – разрядную ячейку памяти. В вычислительной технике такую организацию триггеров принято называть β – разрядным регистром.

Для реализации блока памяти в лабораторной работе должны быть использованы комбинированные D-триггеры. Особенностью комбинированных триггерных схем является то, что наряду с наличием у них синхронно управляемых информационных входов, присутствуют также и входы асинхронной установки S и R триггеров в единичное “1” и нулевое “0” состояния. Входы асинхронной установки необходимы для приведения триггеров в некоторые исходные (начальные) состояния, которые в совокупности соответствуют начальному состоянию синтезируемого синхронного автомата (g_1).

Для того, чтобы детализировать блок памяти синтезируемого автомата в программе EWB необходимо в панели инструментов основного окна программы выбрать компонент DIGITAL. Из меню этого компонента выбрать необходимый тип триггера. Синхронные D-триггеры, которые необходимо использовать при выполнении лабораторной работы, представлены на рис. 35.



Рис. 35. D-триггеры без предустановки а) и со входами предустановки S и R б)

Для соединения элементов блока памяти используется компонент окна BASIC – CONNECTOR, который является точкой соединения. Параметры точки можно задавать с помощью диалогового окна, которое появляется при двойном щелчке правой кнопки мыши по этой точке.

На один из асинхронных входов каждого из триггеров необходимо подать логическую единицу, для этого из компонента BASIC выбирается компонент Pull-Up Resistor –



На другие входы должен быть подан сигнал начальной установки. В результате детализации блока памяти должна быть получена схема, представленная на рис. 36:

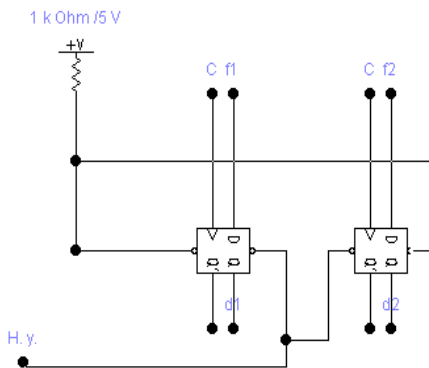


Рис. 36. Детализация блока памяти синтезируемого автомата

5. Построение расширенной таблицы переходов и выходов синтезируемого автомата.

Расширенные структурные таблицы переходов содержат информацию о структурном коде состояния автомата в текущий момент времени $K(g_m)$, о структурном коде автомата в последующий момент времени $K(g_s)$, а также структурный код функции возбуждения блока памяти $F(g_m, g_s)$, который должен формироваться логическим преобразователем для подготовки перехода автомата из состояния g_m в состояние g_s .

При использовании D – триггеров функция возбуждения блока памяти находится на основании следующего уравнения:

$$F(g_m, g_s) = K(g_s). \quad (28)$$

Из уравнения (28) следует система уравнений (29):

$$\begin{aligned} f_1 &= d_1(g_s) \\ f_2 &= d_2(g_s) \\ &\dots \\ f_\beta &= d_\beta(g_s), \end{aligned} \quad (29)$$

где f_i - функция возбуждения i -ого триггера блока памяти, $d_i(g_s)$ – D вход i -ого триггера блока памяти.

Для некоторых гипотетических автоматов типа Мура структурная расширенная таблица переходов и выходов представляется табл. 15. Табл. 15 заполняется с помощью полученных в ходе выполнения лабораторной работы данных.

Поскольку автомат безусловно переходит из одного состояния в другое только по фронту сигнала синхронизации, то входные сигналы $X(g_m, g_s) = 1$.

В каждом из состояний g_m автомат формирует только один активный сигнал на выходах a, b, c, d, что отражено в первой колонке табл. 15.

Таблица 15

$g_m, Y(g_m)$	$K(g_m)$		g_s	$K(g_s)$		$X(g_m, g_s)$	$F(g_m, g_s)$	
	d_2	d_1		d_2	d_1		f_2	f_1
g_1, a			g_2			1		
g_2, b								
g_3, c								
g_4, d								

6. Логические уравнения для функций выходов автомата и функций возбуждения триггеров.

Составление логических уравнений для функций возбуждения блока памяти $F(g_m, g_s)$ сводится к составлению совокупности логических уравнений для каждой отдельной функции возбуждения элементов памяти ($f_1 \dots f_r$).

Логические уравнения записываются как дизъюнкция конъюнкций структурного кода исходного состояния автомата $K(g_m)$ и комбинации входных сигналов $X(g_m, g_s)$ по тем строкам табл. 15, в которых в соответствующем столбце f_i присутствует значение, равное 1.

Составление логических уравнений для выходных сигналов (т.е. функций выходов) производится отдельно для каждого выходного сигнала.

Для автомата типа Мура логические уравнения функций выходов формируется на основе графы $g_m, Y(g_m)$ соответствующей структурной табл. 15. Функции выходов для автомата типа Мура представляют собой дизъюнкции только конъюнкций структурного кода исходного состояния автомата $K(g_m)$ по тем строкам структурной таблицы, в которых присутствует выходной сигнал.

7. Минимизация логических уравнений функций возбуждения блока памяти и выходов синтезируемого автомата.

Целью минимизации логических функций является получение более простых аналитических выражений, что в свою очередь позволит осуществить более простую реализацию соответствующего комбинационного автомата.

8. Схема электрическая функциональная комбинационной части автомата.

Синтезируем схему электрическую функциональную комбинационной части синхронного автомата в элементном базисе И, ИЛИ, НЕ с помощью программы схемотехнического моделирования EWB, то есть реализуем на этих логических элементах полученную в п.7 систему логических уравнений для выходных сигналов автомата - a, b, c, d.

9. Синтез электрической функциональной схемы (логический преобразователь и блок памяти) синтезируемого автомата. Полученная схема моделируется и исследуется средствами пакета схемотехнического моделирования EWB.

В соответствии с рис. 37 в программе EWB синтезируется общая электрическая функциональная схема (логический преобразователь и блок памяти) для синтезируемого автомата.

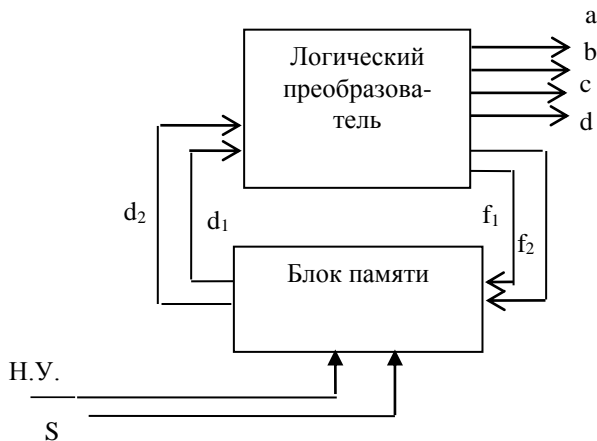


Рис. 37. Структурная модель синхронного автомата типа Мура

Для проверки правильности работы схемы будем использовать генератор слов и логический анализатор [3].

Генератор слов используется для задания цифровых последовательностей, его уменьшенное изображение представлено на рис. 38.



Вход внешней синхронизации
Выход тактового сигнала

Рис. 38. Уменьшенное изображение генератора слов

Вход внешней синхронизации используется для подачи синхронизирующего сигнала от внешнего источника.

Логический анализатор (рис. 39) подключается к исследуемой схеме с помощью внешних выводов. Одновременно могут наблюдаться сигналы в 16 точках схемы.

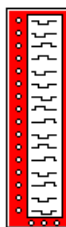


Рис. 39. Уменьшенное изображение логического анализатора

Порядок подключения генератора слов и логического анализатора к синтезируемой схеме:

1. Выход тактового сигнала генератора слов подключается ко входам синхронизации всех триггеров.

2. На один из асинхронных входов каждого из триггеров необходимо подать сигнал (логическая единица) для приведения их в начальные состояния, которые в совокупности соответствуют начальному состоянию синтезируемого синхронного автомата. Недействующие для начальной установки входы триггеров должны быть подключены к дополнительному сигналу. В данной лабораторной работе недействующие входы триггеров также необходимо подключить к логической единице.

3. На входы D триггеров подаются функции возбуждения триггеров f_i при $i=1, 2$.

4. Логический анализатор подключается к исследуемой схеме следующим образом: на первый вывод анализатора подается сигнал синхронизации, на второй вывод – младший разряд структурного двоичного кода d_1 ; на третий вывод – старший разряд структурного кода d_2 (подключить выходы Q D-триггеров); на четвертый, пятый, шестой и седьмой выходы

логического анализатора подаются выходные сигналы a,b,c,d соответственно.

5. Включить режим моделирования работы схемы и проверить правильность ее работы в соответствии с рис. 32.

Лабораторная работа №7

СИНТЕЗ СИНХРОННОГО УПРАВЛЯЮЩЕГО АВТОМАТА

ЦЕЛЬ: закрепление теоретических положений теории автоматов и получение практических навыков по формальному синтезу синхронных автоматов.

ЗАДАНИЕ: в пакете схемотехнического моделирования EWB разработать схему электрическую функциональную синхронного автомата, функционирующего в соответствии с графом, задающим переходы из состояния a_i в состояние a_{i+1} , где $i = 0, \dots, 8$. Варианты графа для выполнения лабораторной работы представлены в табл. 16.

Таблица 16

№ варианта	Граф синтезируемого автомата
	Последовательность выходных сигналов Y_j
1	{ $Y_6, Y_1, Y_3, Y_2, Y_5, Y_4, Y_8, Y_0, Y_7, Y_6$ }
2	{ $Y_1, Y_0, Y_6, Y_2, Y_7, Y_4, Y_8, Y_3, Y_5, Y_1$ }
3	{ $Y_5, Y_6, Y_3, Y_7, Y_2, Y_4, Y_8, Y_1, Y_0, Y_5$ }
4	{ $Y_3, Y_1, Y_0, Y_2, Y_7, Y_4, Y_5, Y_6, Y_8, Y_3$ }
5	{ $Y_4, Y_5, Y_3, Y_2, Y_7, Y_0, Y_8, Y_6, Y_1, Y_4$ }
6	{ $Y_6, Y_1, Y_5, Y_2, Y_7, Y_8, Y_4, Y_0, Y_3, Y_6$ }
7	{ $Y_8, Y_1, Y_3, Y_5, Y_4, Y_7, Y_0, Y_6, Y_2, Y_8$ }
8	{ $Y_2, Y_1, Y_3, Y_0, Y_8, Y_4, Y_7, Y_6, Y_5, Y_2$ }

9	{Y ₇ , Y ₅ , Y ₃ , Y ₄ , Y ₀ , Y ₂ , Y ₈ , Y ₆ , Y ₁ , Y ₇ }
10	{Y ₀ , Y ₅ , Y ₃ , Y ₆ , Y ₇ , Y ₄ , Y ₈ , Y ₂ , Y ₁ , Y ₀ }
11	{Y ₁ , Y ₀ , Y ₅ , Y ₃ , Y ₇ , Y ₄ , Y ₈ , Y ₂ , Y ₆ , Y ₁ }
12	{Y ₅ , Y ₆ , Y ₁ , Y ₇ , Y ₄ , Y ₂ , Y ₈ , Y ₃ , Y ₀ , Y ₅ }
13	{Y ₃ , Y ₈ , Y ₀ , Y ₂ , Y ₄ , Y ₇ , Y ₅ , Y ₆ , Y ₁ , Y ₃ }
14	{Y ₄ , Y ₀ , Y ₃ , Y ₂ , Y ₈ , Y ₅ , Y ₇ , Y ₆ , Y ₁ , Y ₄ }
15	{Y ₆ , Y ₃ , Y ₅ , Y ₂ , Y ₈ , Y ₇ , Y ₄ , Y ₀ , Y ₁ , Y ₆ }
16	{Y ₈ , Y ₂ , Y ₃ , Y ₅ , Y ₄ , Y ₀ , Y ₇ , Y ₆ , Y ₁ , Y ₈ }
17	{Y ₂ , Y ₅ , Y ₃ , Y ₀ , Y ₄ , Y ₈ , Y ₇ , Y ₆ , Y ₁ , Y ₂ }
18	{Y ₇ , Y ₁ , Y ₃ , Y ₂ , Y ₀ , Y ₄ , Y ₆ , Y ₈ , Y ₅ , Y ₇ }
19	{Y ₀ , Y ₅ , Y ₂ , Y ₆ , Y ₄ , Y ₇ , Y ₈ , Y ₃ , Y ₁ , Y ₀ }
20	{Y ₅ , Y ₀ , Y ₁ , Y ₄ , Y ₇ , Y ₂ , Y ₈ , Y ₃ , Y ₆ , Y ₅ }
21	{Y ₁ , Y ₂ , Y ₅ , Y ₃ , Y ₆ , Y ₄ , Y ₈ , Y ₀ , Y ₇ , Y ₁ }
22	{Y ₅ , Y ₆ , Y ₇ , Y ₁ , Y ₄ , Y ₈ , Y ₂ , Y ₃ , Y ₀ , Y ₅ }
23	{Y ₂ , Y ₃ , Y ₅ , Y ₀ , Y ₄ , Y ₆ , Y ₇ , Y ₈ , Y ₁ , Y ₂ }
24	{Y ₈ , Y ₁ , Y ₃ , Y ₅ , Y ₀ , Y ₄ , Y ₇ , Y ₆ , Y ₂ , Y ₈ }

Пример: граф синтезируемого автомата может выглядеть как граф, представленный на рис. 40, где a_i – состояния автомата, $i = 0, \dots, 8$, а стрелками указаны последовательности перехода автомата из одного состояния в другое.

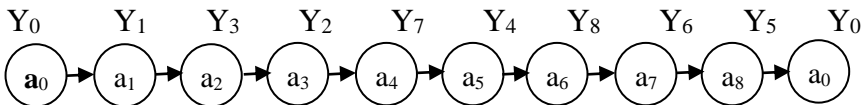


Рис. 40. Граф автомата

Схему электрическую функциональную комбинационной части синхронного автомата необходимо реализовать в выбранном элементном базисе. В качестве элементов памяти

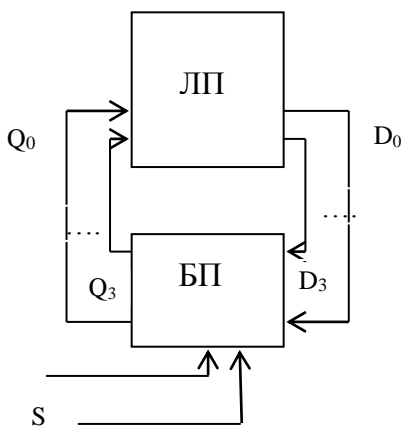
необходимо использовать комбинированные синхронные D-триггеры. Переход автомата из одного состояния в другое осуществляется по фронту сигнала синхронизации.

ХОД ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ:

1. Структурная детализация автомата в соответствии с моделью Мура.

Математическая модель Мура позволяет представить синтезируемый автомат состоящим из двух взаимосвязанных функциональных частей – логического преобразователя (ЛП) и блока памяти (БП), так, как это показано на рис. 41.

ЛП представляет собой комбинационную схему (или комбинационный автомат). БП содержит 4 комбинированных D-триггера.



н. у.

Рис. 41. Первый уровень структурной детализации синтезируемого автомата

2. Составление структурной таблицы кодирования состояний автомата.

Структурное кодирование состояний автомата проводится в два этапа, аналогично структурному кодированию, выполненному в лабораторной работе №6. Сопоставим каждому

отдельному абстрактному символу $a_i \in A$, $i = 0, \dots, 8\beta$ – разрядный двоичный код таким образом, чтобы он соответствовал значению десятичного индекса i каждого состояния автомата. Результаты такого структурного кодирования занесем в табл. 17.

Таблица 17

Структурное кодирование состояний автомата a_i

		Структурный код			
		Q_4	Q_3	Q_2	Q_1
Состояния автомата a_i	a_i				
	a_0	0	0	0	0
	a_1	0	0	0	1
	a_2				
	a_3				
	a_4				
	a_5				
	a_6				
	a_7				
a_8					

Если состояния автомата закодировать в соответствии с табл.17, то для формирования выходных сигналов Y_j потребуется усложнение логического преобразователя, так как он должен будет дополнительно преобразовывать двоичные коды $Q_4 \dots Q_1$ в соответствующие коды $y_{s-1} \dots y_0$. Если же коды $y_{s-1} \dots y_0$ для всех состояний $a_i \in A$ являются отличающимися друг от друга, то их можно непосредственно использовать для структурного кодирования, при этом необходимо и достаточно, чтобы количество триггеров в блоке памяти было равно количеству s выходных сигналов. Для рассматриваемого на рис. 40 случая, максимальное десятичное значение индекса j у выходных двоичных чисел Y_j равно 8. Подставив это значение в соотношение (27) вместо параметра α , получим количество

выходных сигналов $y_0 \dots y_{s-1}$ для данного автомата, то есть $s=4$. В таком случае количество выходных сигналов автомата равно количеству триггеров в блоке памяти, что позволяет существенно упростить логический преобразователь за счёт того, что если изменить структурное кодирование состояний автомата, то выходами автомата будут являться выходы соответствующих триггеров. Тогда структурная схема, представленная на рис. 41, модифицируется в структурную схему, представленную на рис. 42.

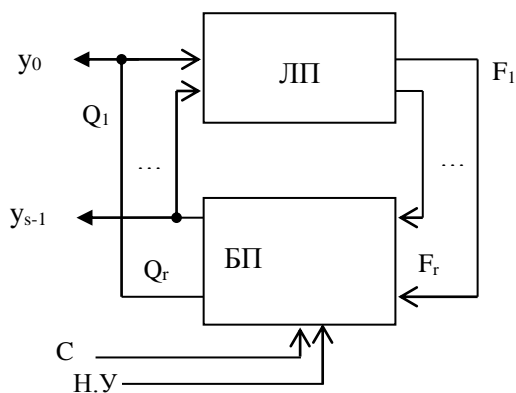


Рис. 42. Модифицированная структурная схема синтезируемого автомата

Для реализации структурной схемы, представленной на рис. 42 необходимо осуществить структурное кодирование состояний автомата, которое представлено в табл. 18 в соответствии со следующим графом автомата (рис. 42).

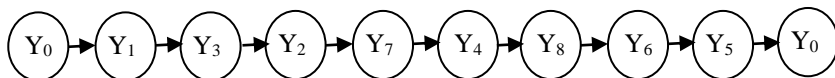


Рис. 42. Модифицированный граф синтезируемого автомата

Модифицированное структурное кодирование состояний автомата Y_j

		Структурные коды			
Индексы состояний автомата Y_j	j	Q ₄	Q ₃	Q ₂	Q ₁
	0	0	0	0	0
	1	0	0	0	1
	3	0	0	1	1
	2	0	0	1	0
	7	0	1	1	1
	4	0	1	0	0
	8	1	0	0	0
	6	0	1	1	0
	5	0	1	0	1

Следует заметить, что модифицированное структурное кодирование возможно и в том случае, если среди выходных двоичных кодов Y_j , формируемых на выходе автомата будут иметься повторяющиеся коды. Для их различения потребуется увеличить разрядность блока памяти.

3. Построение временной диаграммы функционирования синхронного автомата.

В соответствии с табл. 18 построить временную диаграмму синтезируемого синхронного автомата. На временной диаграмме отобразить уровни напряжения сигналов Н.У., С, структурных кодов автомата Q_0 , Q_1 , Q_2 , Q_3 .

4. Детализация блока памяти автомата.

Детализация блока памяти автомата осуществляется по аналогии с детализацией блока памяти, выполненной в лабораторной работе №6. Конечной целью данного этапа является разработка блока памяти синтезируемого автомата, который должен быть реализован на комбинированных D-триггерах.

По сути, блок памяти представляет собой 4 триггера, электрически соединённых определенным образом, или, иначе говоря, представляет одну 4 – разрядную ячейку памяти. Следует отметить, что в разрабатываемом блоке памяти автомата триггеры управляются низким уровнем сигнала.

5. Составление расширенной структурной таблицы переходов автомата.

Исходными данными для составления расширенной структурной таблицы переходов является табл. 18, временная диаграмма функционирования синхронного автомата и граф синтезируемого автомата. Так как для синтеза автомата необходимо использовать комбинированные D – триггеры, то функция возбуждения блока памяти находится на основании соотношения (30):

$$F(a_m, a_s) = K(a_s), \quad (30)$$

где a_m – текущее состояние автомата; a_s – следующее состояние автомата; $K(a_s)$ – структурный код состояния автомата в последующий момент времени; $K(a_m)$ – структурный код состояния автомата в текущий момент времени; $F(a_m, a_s)$ – структурный код функции возбуждения блока памяти.

Расширенная структурная таблица переходов автомата представлена в виде табл. 19.

Таблица 19

a_m	$K(a_m)$				a_s	$K(a_s)$				$F(a_m, a_s)$			
	Q_4	Q_3	Q_2	Q_1		Q_4	Q_3	Q_2	Q_1	F_4	F_3	F_2	F_1
a_0	0	0	0	0	a_1	0	0	0	1	0	0	0	1
a_1	0	0	0	1	a_3	0	0	1	1	0	0	1	1
a_3													
...													

6. Составление логических уравнений для функций возбуждения элементов памяти.

Составление логических уравнений для функций возбуждения блока памяти F_1, \dots, F_4 сводится к получению кратчайших и/или минимальных тупиковых дизъюнктивных нормальных форм, обеспечивающих минимальную структурную сложность при реализации логической функции в элементных базисах И, ИЛИ, НЕ; И-НЕ; ИЛИ-НЕ.

Минимизацию логических функций F_1, \dots, F_4 , значения которых представлены в табл. 19, необходимо осуществить с помощью карт Карно [6, 7].

7. Выбор элементной базы для синтеза логического преобразователя.

Для синтеза ЛП необходимо определить, в каком из элементных базисов выгоднее реализовать функции возбуждения блока памяти F_1, \dots, F_4 . В зависимости от вычислительной сложности логического уравнения, каждое уравнение может быть реализовано в разных элементных базисах.

8. Разработка и моделирование схемы электрической функциональной в пакете Electronics Workbench.

После преобразования логических уравнений F_1, \dots, F_4 в выбранный элементный базис, можно перейти к синтезу схемы электрической функциональной синхронного автомата.

Собрав воедино логические схемы, реализующие логические уравнения F_1, \dots, F_4 и подключив к ним схему БП автомата, в соответствии с рис. 42, получим схему электрическую функциональную (СЭФ) синтезируемого синхронного автомата. Далее необходимо реализовать данную схему средствами пакета схемотехнического моделирования Electronics Workbench (EWB) и проверить правильность ее работы в соответствии с построенной временной диаграммой. Верификация СЭФ должна быть осуществлена средствами пакета схемотехнического моделирования EWB, с помощью логического анализатора (или семисегментного индикатора) и генератора слов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сикорский Р. Булевы алгебры [Текст] / Р. Сикорский. – М.: Мир, 1969. – 376 с.
2. Новожилов О.П. Основы цифровой техники [Текст]: учеб. пособие / О.П. Новожилов. – М.: ИП РадиоСофт, 2004. – 528 с.
3. Кардашев Г.А. Цифровая электроника на персональном компьютере. Electronics Workbench и Micro-Cap [Текст] / Г.А. Кардашев. – М.: Горячая линия-Телеком, 2003. – 311 с.
4. Лачин В.И. Электроника [Текст] : учеб. Пособие / В.И. Лачин, Н.С. Савелов. – Ростов н/Д: Феникс, 2004. – 576 с.
5. Тюрин С.В. Практикум по теории автоматов: синтез синхронного управляющего автомата [Текст]: учеб. пособие / С.В. Тюрин. – Воронеж: ВГТУ, 2004. – 84 с.
6. Аляев Ю.А. Дискретная математика и математическая логика [Текст] / Ю.А. Аляев, С.Ф. Тюрин. – М. : Финансы и статистика, 2006. – 368 с.
7. Закревский А.Д. Логические основы проектирования дискретных устройств [Текст]/ А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. – М.: Физматлит, 2007. – 592 с.

СОДЕРЖАНИЕ

1. Лабораторная работа №1.....	1
2. Лабораторная работа №2.....	8
3. Лабораторная работа №3.....	14
4. Лабораторная работа №4.....	18
5. Лабораторная работа №5.....	22
6. Лабораторная работа №6.....	37
7. Лабораторная работа №7.....	48
Библиографический список.....	56

ПРАКТИЧЕСКИЕ ЗАДАЧИ ТЕОРИИ АВТОМАТОВ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению лабораторных работ № 1-7
по дисциплине «Теория автоматов»
для студентов направления 09.03.01
«Информатика и вычислительная техника»,
профиля «Вычислительные машины, комплексы, системы и
сети» очной и заочной форм обучения

Составители:
Акинина Юлия Сергеевна
Тюрин Сергей Владимирович

В авторской редакции

Подписано в печать .2017.
Формат 60x84/16. Бумага для множительных аппаратов. .
Усл. печ. л. . Уч.-изд.л. . Тираж экз. “С”
Заказ № .

ФГБОУ ВО «Воронежский государственный
технический университет»
394026 Воронеж, Московский просп., 14