

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 21.01.2022 08:53:08

Уникальный программный ключ:

0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

## **МИНОБРНАУКИ РОССИИ**

**Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)**

**Кафедра вычислительной техники**

**УТВЕРЖДАЮ**

Проректор по учебной работе

\_\_\_\_\_ О.Г. Локтионова

«\_\_» \_\_\_\_\_ 2017 г.

## **ЭЛЕКТРОНИКА И СХЕМОТЕХНИКА** **Часть 3**

Методические указания к выполнению лабораторных работ для студентов  
направления подготовки 09.03.01 по дисциплине  
"Электротехника, электроника и схемотехника"

Курск 2017

УДК 681.3.049.77

Составители: В.И. Иванов, С.Н.Гвоздева

Рецензент

Кандидат технических наук *Лапина Т.И.*

**Электроника и схемотехника. Часть 3:** методические указания к выполнению лабораторных работ / Юго-Зап.гос.ун-т; сост.: В.И. Иванов, С.Н.Гвоздева. – Курск, 2017. – 28 с.

Содержат материал, необходимый для формирования у студентов знаний основ цифровой схемотехники и принципов построения цифровых узлов, привития навыков разработки и проектирования цифровых устройств на интегральных схемах.

Методические указания соответствуют Федеральному государственному образовательному стандарту высшего образования направления подготовки 09.03.01 Информатика и вычислительная техника, учебному плану направления подготовки 09.03.01 Информатика и вычислительная техника, одобренному Ученым советом университета (протокол № 7 «29» февраля 2016 г.).

Предназначены для студентов направления подготовки 09.03.01 очной и заочной формы обучения

Текст печатается в авторской редакции

Подписано в печать . Формат 60\*84 1/16.

Усл. печ.л. . Уч.-изд.л. Тираж 100 экз. Заказ Бесплатно.

Юго-Западный государственный университет  
305040 Курск, ул. 50 лет Октября, 94.

## Оглавление

СИНТЕЗ И ИССЛЕДОВАНИЕ СХЕМ ПРЕОБРАЗОВАНИЯ КОДОВ.....	4
ИССЛЕДОВАНИЕ МУЛЬТИПЛЕКСОРОВ И ДЕШИФРАТОРОВ.....	10
АРИФМЕТИКО-ЛОГИЧЕСКИЕ УСТРОЙСТВА.....	18
ТРИГГЕРЫ С ДИНАМИЧЕСКИМ УПРАВЛЕНИЕМ.....	23

# СИНТЕЗ И ИССЛЕДОВАНИЕ СХЕМ ПРЕОБРАЗОВАНИЯ КОДОВ

## 1. Цель работы

Изучение методов синтеза комбинационных схем, в частности, принципов организации схем преобразования кодов, и приобретение навыков в сборке и экспериментальном исследовании синтезируемых схем.

## 2. Порядок синтеза и исследования преобразователя кода

Схемы, работа которых не зависит от состояния входных переменных в предыдущие моменты времени и однозначно определяется лишь комбинацией входных сигналов в текущий момент времени, называют комбинационными. Каждая комбинационная схема описывается системой уравнений.

Задача синтеза комбинационной схемы заключается в построении реальной схемы на основе заданных законов ее функционирования. Причем при синтезе схем чаще всего необходимо построить схему с использованием минимального числа элементов. На пути от исходного описания до готовой схемы можно выделить несколько основных этапов:

1. Составление таблиц или карт функционирования схем.
2. Минимизация.
3. Выбор функционального состава логических элементов и переход к соответствующему базису в полученных в п.2 выражениях.

Минимизация может производиться либо аналитически, путем преобразования математических выражений, либо графически на основе специальных карт. Удобны для использования карты Карно (диаграммы Вейча).

Полученная минимальная функция выражена обычно в дизъюнктивной или конъюнктивной нормальной форме, то есть через логический базис И, ИЛИ, НЕ. Вместе с тем практически используемые серии элементов представляют проектировщику иной логический базис (очень часто это элементы, реализующие операции И-НЕ либо ИЛИ-НЕ). Следовательно, нужно перейти к тому логическому базису, который отвечает выбранным элементам. К базису И-НЕ можно перейти от дизъюнктивной нормальной формы взятием от функции двойной инверсии с последующим применением правила де-Моргана. Аналогично можно перейти к базису ИЛИ-НЕ от конъюнктивной нормальной формы.

В данной работе в качестве устройств комбинационного типа исследуются преобразователи кодов. Преобразователем кодов называется логическое устройство, предназначенное для изменения кодов информации, передаваемой и обрабатываемой цифровыми устройствами.

Существует множество кодов, предназначенных для обеспечения удобства перевода чисел из десятичной системы в двоичную, простоты выполнения арифметических операций и контроля результатов вычислений. Выбор кода влияет также на аппаратные затраты реализуемых схем цифровых устройств и на надежность выполнения ими заданных алгоритмов функционирования.

Входным кодом является четырехразрядный двоичный код десятичной цифры типа "8-4-2-1". Значения кода определены на десяти комбинациях входных переменных  $\{x_3, x_2, x_1, x_0\}$ , начиная от набора 0000 и заканчивая набором 1001. Остальные шесть комбинаций от 1010 до 1111 в работе не принимают участия и при синтезе преобразователя могут использоваться произвольно. Выходной код десятичной цифры представляет собой четырех- или пятиразрядную комбинацию  $\{y_3, y_2, y_1, y_0\}$  или  $\{y_4, y_3, y_2, y_1, y_0\}$  соответственно.

Преобразователь реализуется как система булевых функций  $y_j$  группы аргументов  $x_i$ . Простейшим способом построения схемы, реализующей систему функции с  $n$  выходами, является синтез обычными методами  $n$  независимых одновыходных функций.

Однако экономичное решение обычно получается при подходе к системе функций с учетом их взаимосвязанности. Тогда часто удаётся выявить общие логические фрагменты, входящие в формулы нескольких выходов. Эти фрагменты достаточно реализовать схемно один раз.

На рис. 1 приведена схема, которую необходимо собрать на рабочем поле программы электронного моделирования. Комбинации входного кода генерирует десятичный счетчик (модель 7490), который выбирается из магазина **Digital** в секции **CNT**. На вход СКА счетчика подаются тактовые сигналы от генератора прямоугольных импульсов. Выходы счетчика QA, QB, QC, QD служат источником двоичного кода:  $x_0, x_1, x_2, x_3$  соответственно. Если установить частоту тактовых импульсов 1 кГц, то длительность одного такта составляет 1 мс, а полный цикл – 10 мс.

Схема преобразователя кода, который необходимо синтезировать согласно заданному в табл. 1 варианту, на рис. 1 представлена стандартным условным графическим обозначением.

Исследование работы преобразователя кода производится с помощью логического анализатора **Logic Analyzer** из магазина инструментов **Instruments**, который функционирует как шестнадцатиканальный осциллограф. Анализатор разворачивает на своем экране временные диаграммы прямоугольных логических сигналов: входных переменных  $x_0, x_1, x_2, x_3$  и выходных сигналов преобразователя  $y_0, y_1, y_2, y_3, y_4$ .

### 3. Программа исследований

#### 1. Разработка схемы преобразователя кода

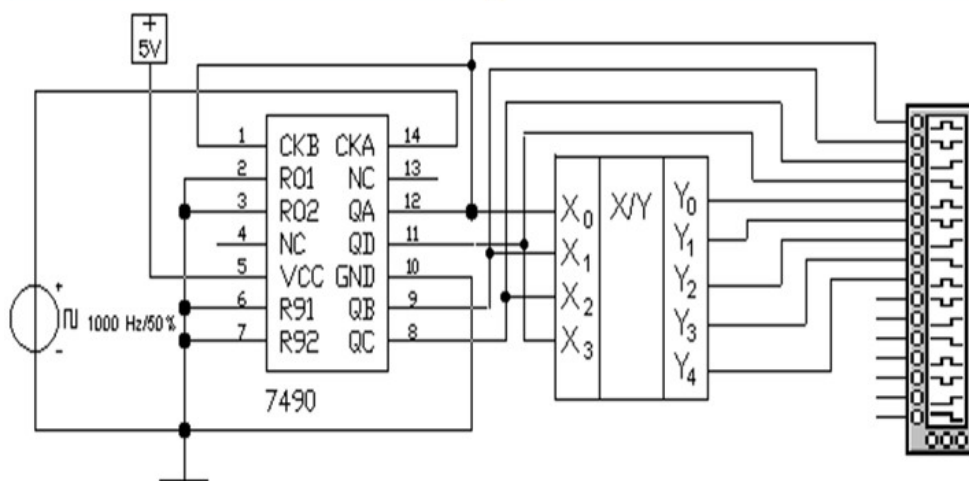


Рис. 1. Схема установки для исследования преобразователя кода

Произвести синтез многovyходной комбинационной схемы для четырех или пяти (в зависимости от варианта задания) переключательных функций четырех переменных (табл. 1.). Привести минимизированные функции  $y_0, y_1, y_2, y_3, y_4$  к элементному базису И–НЕ либо ИЛИ–НЕ. Выбор элементной базы определяется критерием минимальных аппаратных затрат в расчете на количество корпусов микросхем малого уровня интеграции (МИС). Считать, что в одном корпусе ИС размещается:

Варианты заданий

Таблица 1

№ кодовой комбинации	Входной код				1. Код “2-4-2-1”				2. Код “3-3-2-1”			
	$X_3$	$X_2$	$X_1$	$X_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1	1	0	0	0
4	0	1	0	0	0	1	0	0	1	0	0	1

5	0	1	0	1	1	0	1	1	1	0	1	0
6	0	1	1	0	1	1	0	0	1	1	0	0
7	0	1	1	1	1	1	0	1	1	1	0	1
8	1	0	0	0	1	1	1	0	1	1	1	0
9	1	0	0	1	1	1	1	1	1	1	1	1

Продолжение табл. 1

№ код. комб.	3. Дополнение до 10				4. Дополнение до 16				5. Код "2 из 5"				
	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>4</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	1	0	1	0	0	0	0	0	0	0	0	1	1
1	1	0	0	1	1	1	1	1	0	0	1	0	1
2	1	0	0	0	1	1	1	0	0	0	1	1	0
3	0	1	1	1	1	1	0	1	0	1	0	0	1
4	0	1	1	0	1	1	0	0	0	1	0	1	0
5	0	1	0	1	1	0	1	1	0	1	1	0	0
6	0	1	0	0	1	0	1	0	1	0	0	0	1
7	0	0	1	1	1	0	0	1	1	0	0	1	0
8	0	0	1	0	1	0	0	0	1	0	1	0	0
9	0	0	0	1	0	1	1	1	1	1	0	0	0

Продолжение табл. 1

№ код. комб.	6. Код Джонсона					7. Код Грея 1				8. Код Грея 2			
	Y <sub>4</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	1	0	0	0	1
2	0	0	0	1	1	0	0	1	1	0	0	1	1
3	0	0	1	1	1	0	0	1	0	0	0	1	0
4	0	1	1	1	1	0	1	1	0	0	1	1	0
5	1	1	1	1	1	0	1	1	1	1	1	1	0
6	1	1	1	1	0	0	1	0	1	1	0	1	0
7	1	1	1	0	0	0	1	0	0	1	0	1	1
8	1	1	0	0	0	1	1	0	0	1	0	0	1
9	1	0	0	0	0	1	0	0	0	1	0	0	0

Продолжение табл. 1

№ код. комб.	9. Код "3 из 5"					10. Код Грея 3				11. Код Грея 4			
	Y <sub>4</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	1	1	1	0	0	0	0	0	0	0	0
1	0	1	0	1	1	0	1	0	0	0	0	0	1
2	0	1	1	0	1	1	1	0	0	0	1	0	1
3	0	1	1	1	0	1	0	0	0	1	1	0	1
4	1	0	0	1	1	1	0	0	1	1	0	0	1
5	1	0	1	0	1	1	1	0	1	1	0	1	1
6	1	0	1	1	0	0	1	0	1	1	1	1	1
7	1	1	0	0	1	0	0	0	1	0	1	1	1
8	1	1	0	1	0	0	0	1	1	0	0	1	1
9	1	1	1	0	0	0	1	1	1	0	0	1	0

Продолжение табл. 1

№ код.	12. Код Грея 5				13. Код Грея 6				14. Код Грея 7			
	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>

комб.												
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	1
2	0	0	1	1	0	1	0	1	0	1	0	1
3	0	0	1	0	0	1	1	1	0	1	0	0
4	0	1	1	0	0	1	1	0	1	1	0	0
5	1	1	1	0	1	1	1	0	1	1	0	1
6	1	1	1	1	1	1	1	1	1	0	0	1
7	1	1	0	1	1	1	0	1	1	0	1	1
8	1	1	0	0	1	1	0	0	1	0	1	0
9	1	0	0	0	1	0	0	0	1	1	1	0

Продолжение табл. 1

№ код. комб.	15. Код Грея 8				16. Код Грея 9				17. Код Грея 10			
	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	1	0	0	0	1
2	0	1	0	1	0	0	1	1	0	0	1	1
3	0	0	0	1	0	1	1	1	0	1	1	1
4	0	0	1	1	1	1	1	1	0	1	1	0
5	0	1	1	1	1	1	0	1	1	1	1	0
6	0	1	1	0	1	1	0	0	1	1	1	1
7	1	1	1	0	1	0	0	0	1	1	0	1
8	1	1	1	1	1	0	0	1	0	1	0	1
9	1	1	0	1	1	0	1	1	0	1	0	0

Продолжение табл. 1

№ код. комб.	18. Код Грея 11				19. Код Грея 12				20. Код Грея 13			
	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	1	0	0
2	0	0	1	1	1	1	0	0	1	1	0	0
3	0	1	1	1	1	1	0	1	1	0	0	0
4	0	1	0	1	1	0	0	1	1	0	0	1
5	1	1	0	1	1	0	1	1	1	0	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1
7	1	0	1	1	1	1	1	0	0	1	1	1
8	1	0	1	0	0	1	1	0	0	0	1	1
9	1	1	1	0	0	0	1	0	0	0	0	1

Продолжение табл. 1

№ код. комб.	21. Код с избытком 6				22. Код "4-3-2-1"				23. Код "5-2-1-1"			
	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	1	1	0	0	0	0	0	0	0	0	0
1	0	1	1	1	0	0	0	1	0	0	0	1
2	1	0	0	0	0	0	1	0	0	1	0	0
3	1	0	0	1	0	1	0	0	0	1	1	0
4	1	0	1	0	1	0	0	0	0	1	1	1
5	1	0	1	1	1	0	0	1	1	0	0	0
6	1	1	0	0	1	0	1	0	1	0	1	0

7	1	1	0	1	1	1	0	0	1	1	0	0
8	1	1	1	0	1	1	0	1	1	1	1	0
9	1	1	1	1	1	1	1	0	1	1	1	1

- 6 инверторов;
- 4 двухвходовых элемента;
- 3 трехвходовых элемента;
- 2 четырехвходовых элемента;
- 1 восьмивходовый элемент.

Ниже приведен пример синтеза переключательной функции.

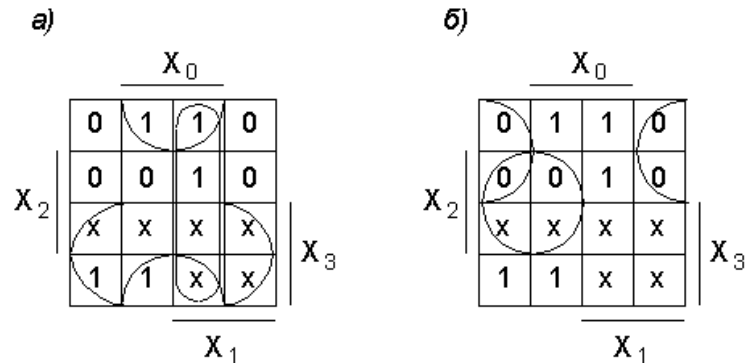


Рис. 2. Диаграммы Вейча функции четырех переменных

На рис. 2, а показано объединение ячеек, содержащих единицы, которое позволяет получить минимальную дизъюнктивную форму синтезируемой функции, приводящую к реализации в базисе И–НЕ:

$$y = x_3 \vee x_1 x_0 \vee \bar{x}_2 x_0 = \overline{\bar{x}_3} \wedge \overline{\bar{x}_1 x_0} \wedge \overline{\bar{x}_2 x_0}$$

Для реализации схемы требуется два инвертора (1 / 3 ИС), два двухвходовых элемента (1 / 2 ИС) и один трехвходовый элемент (1 / 3 ИС), то есть всего 7 / 6 корпуса ИС.

На рис. 2, б показано объединение ячеек, содержащих нули, которое позволяет получить минимальную дизъюнктивную форму для инверсии синтезируемой функции, удобную для реализации в базисе ИЛИ–НЕ:

$$y = \overline{\bar{x}_3 \bar{x}_0 \vee x_2 \bar{x}_1} = \overline{\bar{x}_3} \vee \overline{\bar{x}_0} \vee \overline{x_2} \vee \overline{\bar{x}_1}$$

Для реализации функции в такой форме требуется один инвертор (1 / 6 ИС) и три двухвходовых элемента (3 / 4 ИС), то есть всего 11 / 12 корпуса ИС. В этом примере более экономична реализация заданной функции в базисе ИЛИ–НЕ.

## 2. Построение схемы преобразователя кода

Собрать на рабочем поле схему для исследования спроектированного преобразователя кода согласно рис. 1. Инверторы (**NOT Gate**) и логические элементы типа И–НЕ (**NAND Gate**) либо ИЛИ–НЕ (**NOR Gate**) следует выбирать из магазина компонентов **Logic Gates**. По умолчанию элементы **NAND Gate** и **NOR Gate** имеют по два входа. Требуемое число входов от 2 до 8 можно установить, вызвав двойным кликом мыши по изображению элемента меню **2-Input NAND Gate Properties** либо **2-Input NOR Gate Properties** и в подменю **Number of Inputs** отметить нужное число.

В подменю **Models** выбрать библиотеку **t1l** (ТТЛ) и модель типа **LS** (маломощные ТТЛШ, аналоги отечественной серии К555).

Соединить входы и выходы преобразователя кода с входами логического анализатора.

## 3. Исследование временных диаграмм преобразователя кода

Исследование функционирования преобразователя кода производится в динамическом режиме с помощью логического анализатора.



Выполнить следующие установки в блоке **Clock Set...** на панели анализатора:  
**Clock Mode** – **Internal** (задает внутренний режим запуска от самих исследуемых сигналов);

**Internal clock rate** – 8 кГц (частота тактовых импульсов);

**Threshold voltage (V)** – 1,3 (порог срабатывания анализатора).

Остальные установки оставить по умолчанию и нажать кнопку **Accept**.

Запустить программу моделирования и, когда на экране анализатора заполнятся диаграммы напряжений, остановить процесс моделирования кнопкой **Pause**. Выбрать подходящий масштаб по оси времени с помощью установки **Clocks per division**, так чтобы на развертке укладывался полный цикл работы преобразователя.

Зарисовать для отчета временные диаграммы сигналов на всех входах и выходах схемы. Пример диаграмм показан на рис. 3.

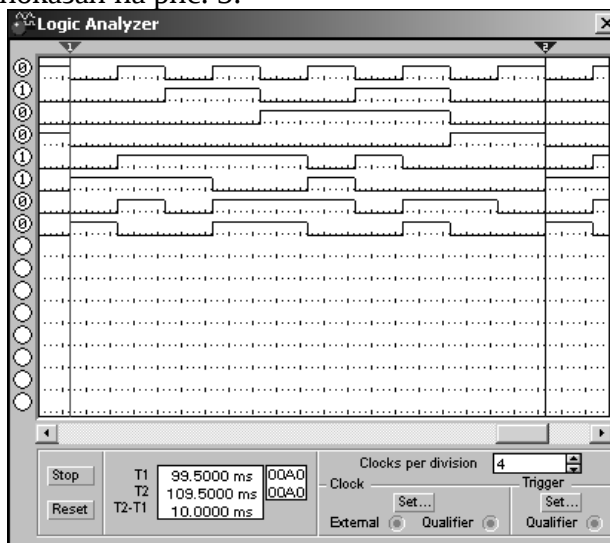


Рис. 3. Временные диаграммы преобразователя кода

#### 4. Контрольные вопросы

1. Каковы основные признаки комбинационных схем?
2. В чем сущность синтеза комбинационных схем?
3. В чем специфика синтеза многовыходных комбинационных схем?
4. Почему задача синтеза неоднозначна?
5. Какие критерии минимизации применяют при синтезе комбинационной схемы?
6. Для чего предназначены схемы преобразователей кодов?
7. Какие существуют методы построения схем преобразователей кодов?

#### 5. Содержание отчета

Отчет должен содержать:

- 1) титульный лист;
- 2) наименование работы и цель исследований;
- 3) результаты синтеза преобразователя кода (карты Карно, обоснование выбора элементной базы, схема преобразователя);
- 4) схема установки для исследования преобразователя кода на рабочем поле программы;
- 5) результаты исследования работы преобразователя кода в динамическом режиме (временные диаграммы входных и выходных сигналов).

#### Библиографический список

1. Угрюмов Е.П. Цифровая схемотехника: Учебное пособие. – СПб.: БХВ-Петербург, 2001. – 528 с.

2. Угрюмов Е.П. Проектирование элементов и узлов ЭВМ: Учеб. пособие для вузов. – М.: Высш. шк., 1987. – 318 с.
3. Иванов В.И. Проектирование цифровых устройств: учебное пособие / В.И. Иванов, В.С. Титов, М.В. Бобырь, А.С. Ястребов; Юго-Зап. гос. ун-т. Курск, 2011. 100 с.

## ИССЛЕДОВАНИЕ МУЛЬТИПЛЕКСОРОВ И ДЕШИФРАТОРОВ

### 1. Цель работы

Изучение принципов организации типовых функциональных узлов комбинационного типа: мультиплексоров и дешифраторов. Изучение методов реализации логических функций на мультиплексорах и дешифраторах. Получение навыков в сборке, наладке и исследовании схем.

### 2. Принципы организации мультиплексоров и дешифраторов. Методика реализации логических функций на мультиплексорах и дешифраторах

Мультиплексором называется комбинационное устройство, предназначенное для коммутации в заданном порядке сигналов с нескольких входных шин на одну выходную шину. С помощью мультиплексора осуществляется временное разделение информации, поступающей по разным каналам к одному приемнику.

Входы мультиплексора делятся на информационные, адресные и разрешающие (стробирующие). На информационные входы подается информация, передаваемая на выход мультиплексора. Адресные входы обеспечивают выбор нужного информационного входа, а на разрешающий вход подается стробирующий сигнал, позволяющий подключить выбранный вход на один общий (реже два – прямой и инверсный) выход. Число информационных и адресных входов взаимосвязано. Если число адресных входов равно  $m$ , то с их помощью можно коммутировать  $2^m$  каналов, т.е. максимальное число информационных входов может быть равно  $2^m$ .

Разрешающий вход, как правило, один, а если их несколько, то сигнал на выходе мультиплексора появится только при наличии разрешающих сигналов на всех стробирующих входах. Наличие разрешающих входов позволяет синхронизировать работу мультиплексора с работой других узлов, а также наращивать его разрядность.

Работа мультиплексора описывается следующим логическим уравнением

$$F = E \cdot \sum_{i=0}^{2^m - 1} D_i S_i \quad (1)$$

где  $D_i$  – входной сигнал на  $i$ -том информационном входе мультиплексора;  $S_i$  –  $i$ -й минтерм, образованный переменными на адресных входах;  $E$  – стробирующий сигнал.

Из уравнения (1) следует, что структура мультиплексора состоит из  $2^m$  конъюнкторов (схем И), каждый из которых имеет  $m$  адресных входов, по одному информационному и стробирующему входу, и одного дизъюнктора (схемы ИЛИ), объединяющего  $2^m$  выходов конъюнкторов. Выход дизъюнктора является выходом мультиплексора.



Рис. 1. Условные графические обозначения мультиплексора; по ГОСТ (а) и в системе DIN (б)

На рис. 1 приведены условные графические обозначения интегральных микросхем (ИС) мультиплексора по ГОСТ (а) и в системе DIN, которая применяется в программе моделирования электронных схем (б). Во втором варианте приняты следующие обозначения сигналов:  $D_0, \dots, D_7$  – информационные входы;  $A, B, C$  – адресные входы;  $G'$  – сигнал разрешения (инверсный);  $Y$  – прямой выход;  $W$  – инверсный выход.

Мультиплексоры, выпускаемые в виде отдельных ИС, имеют ограниченное число информационных входов (не более 16). Для увеличения числа входов мультиплексоры наращивают объединением нескольких мультиплексоров в пирамидальную (древовидную) или параллельную схему.

Пирамидальные схемы из интегральных мультиплексоров строятся по каскадному принципу. Число каскадов определяется требуемым числом входов проектируемого мультиплексора и числом входов реальных микросхем мультиплексоров.

На адресные входы мультиплексоров первого каскада подаются младшие разряды кода адреса. Выходы мультиплексоров первого каскада подаются на информационные входы микросхем второго каскада. На адресные входы ИС второго каскада подключаются следующие по старшинству разряды кода адреса. В последнем каскаде оказывается одна ИС мультиплексора, на информационные входы которой подключаются выходы всех мультиплексоров предыдущего яруса, а на адресные – старшие разряды кода адреса. При наличии стробирующих входов все они объединяются общей шиной. Недостатками пирамидальных структур являются повышенный расход микросхем и снижение быстродействия.

Если ИС мультиплексора имеет разрешающий вход, то за счет его использования как информационного можно нарастить разрядность, включая отдельные микросхемы параллельно. Кроме ИС мультиплексоров, требуется дешифратор. В этом случае разрешающие входы всех мультиплексоров подключаются к выходам дешифратора, на входы которого подаются старшие разряды кода адреса, а выходы всех мультиплексоров объединяются через схему ИЛИ. Быстродействие таких мультиплексоров существенно выше и не зависит от разрядности.

Мультиплексоры могут использоваться не только как коммутаторы информационных каналов, но и для других целей. Одна из перспективных областей их использования – реализация комбинационных логических схем, а в ряде случаев и последовательностных цифровых устройств. На мультиплексоре можно реализовать любую логическую функцию, содержащую до  $m$  переменных, где  $m$  – число адресных входов.

При реализации логической функции  $m$  переменных на мультиплексоре, имеющем  $m$  адресных входов, на информационные входы  $D_i$  подаются значения “0” или “1” согласно значениям функции на данном наборе. На адресные входы поступают входные переменные реализуемой функции в соответствии с уравнением (1):

$$A_0 = x_0, A_1 = x_1, \dots, A_{m-1} = x_{m-1},$$

где  $x_i$  – логические переменные;  $A_i$  – адресные входы мультиплексора.

При реализации на мультиплексоре с  $m$  адресными входами функции, содержащей  $m+1$  переменную, поступают следующим образом.

Пусть  $F(x_0, x_1, \dots, x_{m-1}, x_m)$  – некоторая переключательная функция. В соответствии с теоремой Шеннона [2] разложим функцию  $F$  по переменной  $x_m$ :

$$F(x_0, x_1, \dots, x_{m-1}, x_m) = F(x_0, x_1, \dots, x_{m-1}, 0) \dot{\vee} F(x_0, x_1, \dots, x_{m-1}, 1).$$

Подфункции  $F(x_0, x_1, \dots, x_{m-1}, 0)$  и  $F(x_0, x_1, \dots, x_{m-1}, 1)$  на каждом  $i$ -м наборе переменных  $(x_0, x_1, \dots, x_{m-1})$  либо не зависят от значения переменной  $x_m$ , то есть равны “0” или “1”, либо равны значению  $x_m$ , либо равны ее инверсии  $\bar{x}_m$ .

Если значение заданной функции  $F(x_0, x_1, \dots, x_{m-1}, x_m)$  на  $i$ -м наборе переменных  $(x_0, x_1, \dots, x_{m-1})$  не зависит от аргумента  $x_m$ , то на  $i$ -й информационный вход подается константа “0” или “1”, если же зависит, то на соответствующий вход подается либо сигнал самой переменной  $x_m$ , либо ее инверсии  $\bar{x}_m$ .

В случае реализации функции от большего, чем  $m+1$ , количества переменных прибегают к специальным методам расширения количества входов.

Двоичными дешифраторами называются комбинационные логические устройства, имеющие  $n$  входов  $(X_0, X_1, \dots, X_{n-1})$  и  $N$  выходов  $(F_0, F_1, F_2 \dots, F_{N-1})$ , в которых каждому состоянию выходов однозначно соответствует одна из возможных комбинаций двоичных сигналов на входах. Если на входы дешифратора подается  $n$ -разрядное двоичное число, то на одном из его выходов вырабатывается сигнал единицы (нуля), а на остальных выходах сохраняется сигнал нуля (единицы). Максимальное число выходов дешифратора определяется величиной  $N = 2^n$ .

Дешифратор называется полным, если  $N = 2^n$ , и неполным, если  $N < 2^n$ .

В общем случае схема двоичного дешифратора может быть описана с помощью системы переключательных функций переменных следующего вида:

$$\begin{aligned} F_0 &= \bar{x}_{n-1} \bar{x}_{n-2} \dots \bar{x}_1 \bar{x}_0 ; \\ F_1 &= \bar{x}_{n-1} \bar{x}_{n-2} \dots \bar{x}_1 x_0 ; \\ F_2 &= \bar{x}_{n-1} \bar{x}_{n-2} \dots x_1 \bar{x}_0 ; \\ &\dots \\ F_{2^{n-2}} &= x_{n-1} \bar{x}_{n-2} \dots x_1 \bar{x}_0 ; \\ F_{2^{n-1}} &= x_{n-1} x_{n-2} \dots x_1 x_0 , \end{aligned}$$

где  $x_0, x_1, \dots, x_{n-1}$  – двоичные переменные на соответствующих входах дешифратора,  $F_0, F_1, F_2 \dots, F_{2^n-1}$  – переключательные функции выходов дешифратора.

В зависимости от способа структурной реализации этих функций различают матричные и пирамидальные дешифраторы. Быстродействие дешифратора определяется суммарным временем задержки последовательно включенных логических элементов и зависит от количества ступеней дешифратора.

Схема матричного дешифратора строится на основании выражений (2), так что каждое уравнение системы реализуется отдельным логическим элементом. При построении неполных матричных дешифраторов возможно упрощение отдельных переключательных функций.

При построении матричного дешифратора на большое число входных переменных могут возникнуть трудности с реализацией схемы из-за ограничения числа входов у логических элементов, количества логических элементов на кристалле внутри корпуса ИС, а также количества внешних выводов ИС.

На рис. 2 приведены условные графические изображения ИС дешифратора по нормам ГОСТ: с прямыми выходами и прямым сигналом разрешения (а) и с инверсными выходными сигналами и инверсным сигналом разрешения (б).

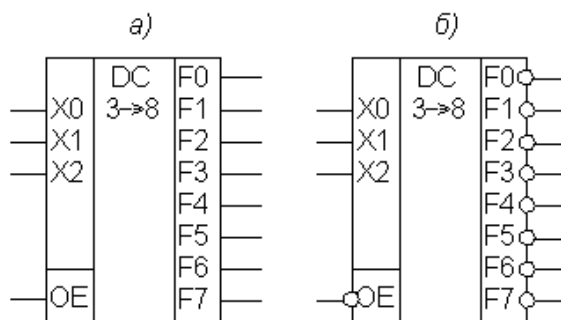


Рис. 2. Условные графические изображения ИС дешифратора по нормам ГОСТ

На рис. 3 приведены условные графические изображения ИС дешифратора в соответствии со стандартом DIN: с прямыми выходами (а) и с инверсными выходными сигналами (б). В схеме рис. 3, а имеется три входа разрешения: один прямой (G1) и два инверсных (G2A' и G2B').

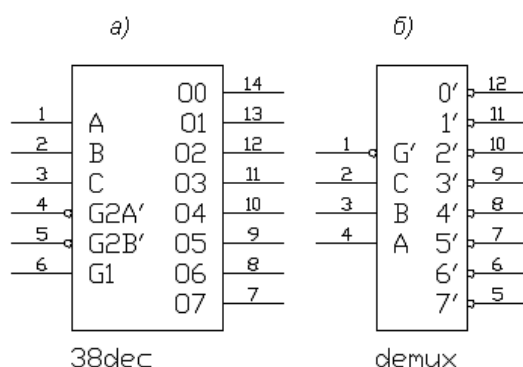


Рис. 3. Условные графические изображения ИС дешифратора по стандарту DIN

Для наращивания разрядности малоразрядных дешифраторов применяются схемы пирамидального типа. Особенностью такого схемотехнического построения является то, что входное многоразрядное слово делится на два “поля”. Поле младших разрядов соответствует числу входов имеющихся дешифраторов. Оставшееся поле разрядов используется для формирования сигналов разрешения работы одного из дешифраторов, декодирующего поле младших разрядов.

Дешифраторы могут использоваться не только как преобразователи двоичного кода в унарный код типа “1 из N”, но и для реализации комбинационных логических схем. На дешифраторе совместно с дополнительными логическими элементами можно реализовать любую логическую функцию, содержащую до  $n$  переменных, где  $n$  – число входов. В самом деле, в соответствии с выражениями (2) схема дешифратора формирует  $N$  всех возможных минтермов от  $n$  входных переменных. Те выходы, которые соответствуют единичным значениям заданной логической функции, необходимо объединить по схеме ИЛИ, и получится реализация функции в СНДФ. При практической реализации устройства следует учитывать, что базовыми логическими элементами существующих серий являются многовходовые элементы типа И–НЕ либо ИЛИ–НЕ. Выбирая дешифраторы с прямыми или инверсными выходами, можно минимизировать аппаратные затраты.

В случае, когда число переменных  $n+1$  превышает на единицу количество входов дешифратора  $n$ , и переменная  $x_n$  влияет на значения заданной функции, можно применить различные схемные решения. Одно из них заключается в использовании двух ИС дешифратора: одна из них стробируется по разрешающему входу сигналом  $\bar{x}_n$ , а вторая – сигналом  $x_n$ .

### 3. Порядок выполнения работы

1. Разработка схемы и исследование логического устройства на базе мультиплексора

Разработать устройство на базе мультиплексора “8×1” для реализации логической функции, истинной на наборах четырех переменных  $x_3 x_2 x_1 x_0$ , десятичные номера которых для различных вариантов указаны в таблице 1, собрать его и проверить функционирование.

На рис. 4 приведен пример схемы, которую необходимо собрать на рабочем поле программы электронного моделирования. Комбинации входного кода генерирует двоичный счетчик (ИС общего назначения серии ТТЛШ), который выбирается из магазина Digital в секции CNT. В свойствах счетчика установить библиотеку ttl и модель LS (ТТЛШ).

На вход СКА счетчика подаются тактовые сигналы от генератора прямоугольных импульсов. Выходы счетчика А, В, С, D служат источником двоичного кода:  $x_0, x_1, x_2, x_3$  соответственно. Если установить частоту тактовых импульсов 1 кГц, то длительность одного такта составляет 1 мс, а полный цикл – 16 мс.

В качестве мультиплексора выбрать в магазине Digital в секции MUX ИС общего назначения Generic 1-of-8 MUX и в ее свойствах выбрать библиотеку ttl, модель LS, то есть ТТЛШ.

Таблица 1

Вариант	Десятичные номера наборов $x_3 x_2 x_1 x_0$	Вариант	Десятичные номера наборов $x_3 x_2 x_1 x_0$
1	0,1,2,3,12,13,14,15	16	0,1,3,4,5,7,8,12
2	0,1,2,3,8,9,10,11	17	0,1,3,4,5,7,9,13
3	0,1,2,3,4,8,12	18	0,1,3,4,5,7,11,15
4	0,1,2,3,5,9,13	19	0,1,2,3,4,8,9,10,11,12
5	0,1,2,3,6,10,14	20	5,7,8,9,10,11,13,15
6	0,1,2,3,7,11,15	21	1,3,4,5,12,14
7	0,4,5,6,7,8,12	22	6,7,8,9,13,15
8	1,4,5,6,7,9,13	23	1,4,5,9,12,13
9	3,4,5,6,7,11,15	24	1,5,7,15
10	2,4,5,6,7,10,14	25	0,1,2,3,4,5,6,10,14
11	0,4,8,12,13,14,15	26	2,3,5,6,7,8,9,13,15
12	1,5,9,12,13,14,15	27	5,7,9,10,11,13,15
13	3,7,11,12,13,14,15	28	1,3,5,7,9,11,12,13,15
14	0,1,4,5,7,10,11,13,14,15	29	1,3,4,5,6,7,9,11
15	0,1,2,4,5,6,8,12	30	1,3,8,9,10,11,13,15

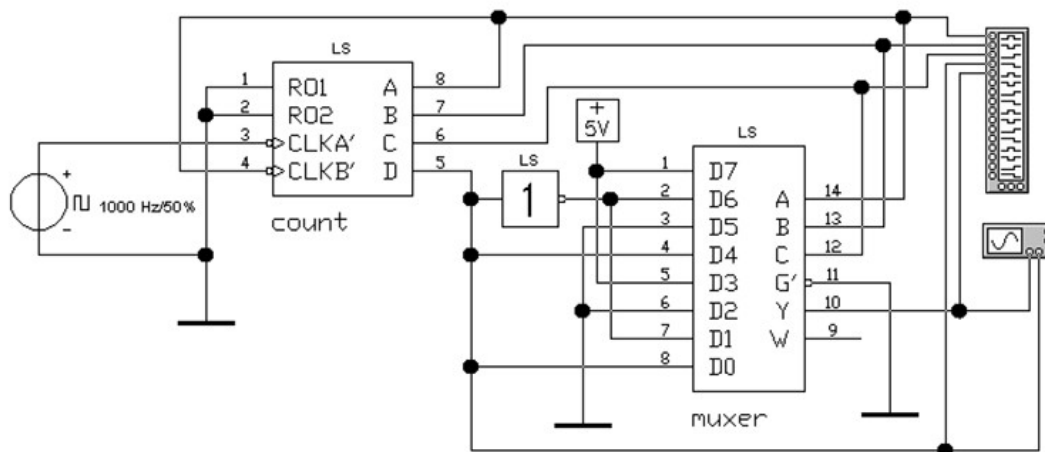


Рис. 4. Схема установки для исследования логического устройства на базе мультиплексора

Исследование работы логического устройства производится с помощью логического анализатора Logic Analyzer и осциллографа Oscilloscope из магазина инструментов Instruments.

Логический анализатор функционирует как многоканальный осциллограф. Он развертывает на своем экране временные диаграммы прямоугольных логических сигналов: входных переменных  $x_0, x_1, x_2, x_3$  и выходного сигнала устройства F.

Перед началом моделирования выполнить следующие установки в блоке Clock Set... на панели анализатора:

Clock Mode – Internal (задает внутренний режим запуска от самих исследуемых сигналов);

Internal clock rate – 8 кГц (частота тактовых импульсов);

Threshold voltage (V) – 1,3 (порог срабатывания анализатора).

Остальные установки оставить по умолчанию и нажать кнопку Ассерт .

Запустить программу моделирования и, когда на экране анализатора заполнятся диаграммы напряжений, остановить процесс моделирования кнопкой Pause. Выбрать подходящий масштаб по оси времени с помощью установки Clocks per division, так чтобы на развертке укладывался полный цикл работы устройства.

Зарисовать для отчета временные диаграммы сигналов на всех входах и выходе устройства.

Просмотреть на экране осциллографа осциллограммы напряжений старшей переменной (для временной привязки) и выходного сигнала устройства. Обратит внимание на появление ложных выбросов в выходном напряжении на границах некоторых тактов – помехи “статического риска”. Зафиксировать полученные осциллограммы в отчете.

## 2. Разработка и исследование схемы логического устройства на базе дешифратора

Разработать устройство на базе трехразрядного дешифратора “3 $\times$ 8” для реализации логической функции, истинной на наборах четырех переменных  $x_3 x_2 x_1 x_0$ , десятичные номера которых для различных вариантов указаны в таблице 2, собрать его и проверить функционирование.

Таблица 2

Вариант	Десятичные номера наборов $x_3 x_2 x_1 x_0$	Вариант	Десятичные номера наборов $x_3 x_2 x_1 x_0$
1	1,3,4,5,12,14	16	0,1,2,3,7,11,15
2	6,7,8,9,13,15	17	0,4,5,6,7,8,12

3	1,4,5,9,12,13	18	1,4,5,6,7,9,13
4	1,5,7,15	19	3,4,5,6,7,11,15
5	0,1,2,3,4,5,6,10,14	20	2,4,5,6,7,10,14
6	2,3,5,6,7,8,9,13,15	21	0,4,8,12,13,14,15
7	5,7,9,10,11,13,15	22	1,5,9,12,13,14,15
8	1,3,5,7,9,11,12,13,15	23	3,7,11,12,13,14,15
9	1,3,4,5,6,7,9,11	24	0,1,4,5,7,10,11,13,14,15
10	1,3,8,9,10,11,13,15	25	0,1,2,4,5,6,8,12
11	0,1,2,3,12,13,14,15	26	0,1,3,4,5,7,8,12
12	0,1,2,3,8,9,10,11	27	0,1,3,4,5,7,9,13
13	0,1,2,3,4,8,12	28	0,1,3,4,5,7,11,15
14	0,1,2,3,5,9,13	29	0,1,2,3,4,8,9,10,11,12
15	0,1,2,3,6,10,14	30	5,7,8,9,10,11,13,15

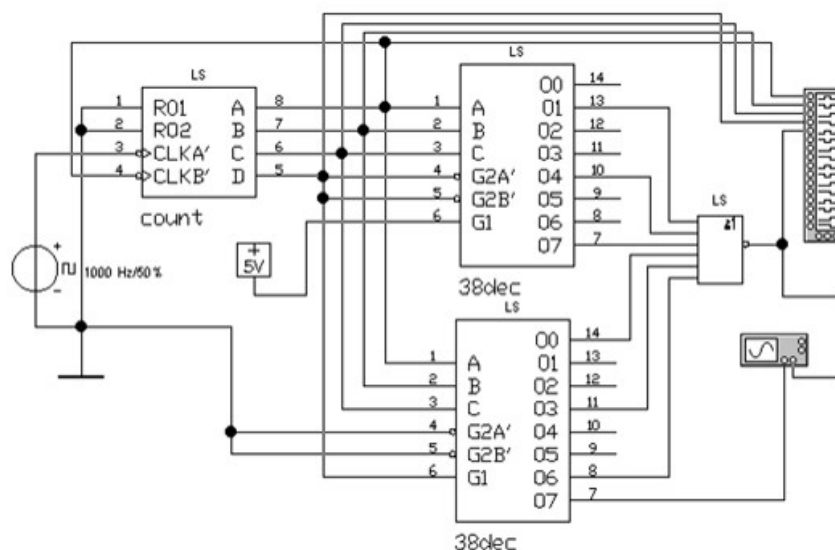


Рис. 5. Схема установки для исследования логического устройства на ИС дешифраторов

На рис. 5 приведена в качестве примера схема логического устройства, реализующего логическую функцию, которая принимает единичное значение на 10 наборах входных переменных: 0, 2, 3, 5, 6, 9, 10, 12, 13, 15. При объединении выходов дешифраторов, соответствующих конstituентам единицы, потребовалось бы использовать десятивыходовый логический элемент, который не содержится в реальных сериях элементов. Более экономным схемным решением является объединение шести выходов дешифраторов, которые соответствуют нулевым значениям функции. Для этого применяются ИС дешифраторов с прямыми выходными сигналами и логический элемент типа ИЛИ-НЕ.

В качестве дешифратора выбрать в магазине Digital в секции DEC ИС общего назначения Generic 8-to-1 DEMUX (дешифратор с инверсными выходами) или Generic 3-to-8 Dec (дешифратор с прямыми выходами) и в ее свойствах выбрать библиотеку ttl, модель LS, то есть ТТЛШ.

Остальные интегральные схемы в установке рис. 5 выбраны также из библиотеки ttl, модель LS.

Исследование работы логического устройства производится в динамическом режиме с помощью логического анализатора Logic Analyzer и осциллографа Oscilloscope



из магазина инструментов Instruments. Перед началом моделирования выполнить установки в блоке Clock Set... на панели анализатора как указано в предыдущем пункте.

Запустить программу моделирования. Выбрать подходящий масштаб по оси времени с помощью установки Clocks per division, так чтобы на развертке укладывался полный цикл работы устройства. Зарисовать для отчета временные диаграммы сигналов на всех входах и выходе устройства.

Просмотреть на экране осциллографа осциллограмму выходного напряжения устройства. Для синхронизации выходного сигнала с началом цикла на другой вход осциллографа нужно подать любой сигнал с известным временным положением, например, в начале или в конце цикла. Обратит внимание на появление ложных выбросов в выходном напряжении на границах некоторых тактов – помехи “статического риска”. Зафиксировать полученные осциллограммы в отчете.

#### **4. Контрольные вопросы**

1. Какие комбинационные устройства называются мультиплексорами и для каких целей они используются?

2. Назовите основные способы наращивания разрядности мультиплексоров и приведите сравнительные характеристики.

3. Какие существуют методы реализации логических функций на основе мультиплексоров?

4. Какие комбинационные устройства называются дешифраторами? Приведите примеры использования дешифраторов.

5. Назовите основные способы наращивания разрядности дешифраторов. Какой тип дешифратора наиболее быстродействующий и почему?

6. Какие существуют методы реализации логических функций на основе дешифраторов?

#### **5. Содержание отчета**

Отчет должен содержать:

6) титульный лист;

7) наименование работы и цель исследований;

8) результаты разработки схемы логического устройства на базе мультиплексора (обоснование выбора элементной базы, схема устройства);

9) результаты исследования работы логического устройства на мультиплексоре в динамическом режиме (временные диаграммы входных и выходных сигналов)

10) результаты разработки схемы логического устройства на базе дешифраторов (обоснование выбора элементной базы, схема устройства);

11) результаты исследования работы логического устройства на дешифраторах в динамическом режиме (временные диаграммы входных и выходных сигналов).

#### **Библиографический список**

1. Угрюмов Е.П. Цифровая схемотехника: Учебное пособие. – СПб.: БХВ-Петербург, 2001. – 528 с.

2. Угрюмов Е.П. Проектирование элементов и узлов ЭВМ: Учеб. пособие для вузов. – М.: Высш. шк., 1987. – 318 с.

## АРИФМЕТИКО-ЛОГИЧЕСКИЕ УСТРОЙСТВА

### 1. Цель работы

Изучение работы интегральной схемы (ИС) арифметико-логического устройства (АЛУ) и проектирования многоразрядного цифрового устройства на основе четырехразрядного АЛУ.

### 2. Основные теоретические положения и описание принципиальной схемы

Центральной частью процессора любой цифровой вычислительной машины от простейших контроллеров до современных персональных компьютеров является арифметико-логическое устройство – АЛУ (см. рисунок 1). Его основой служит многоразрядный сумматор, схема которого дополнена логикой, расширяющей функциональные возможности АЛУ и обеспечивающей перестройку с одной операции на другую.

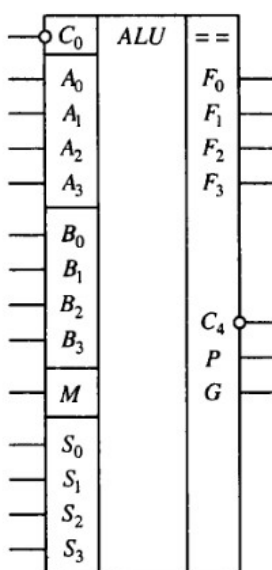


Рисунок 1 – Графическое изображение ИС четырехразрядного АЛУ

Эта перестройка осуществляется путем подачи на АЛУ соответствующего набора управляющих сигналов, называемого *командой*. По одной команде АЛУ может складывать числа, по другой вычитать, по третьей выполнять логические операции над разрядами входных данных, к примеру, инвертировать их и т. п. Если команды и данные подавать в определенной последовательности, то будет производиться соответствующая обработка поступающей цифровой информации. Комбинируя команды, управляющие работой АЛУ, можно реализовывать разные процедуры обработки цифровых сигналов. С этой точки зрения АЛУ является программируемым устройством, т. е. оно способно работать по заранее сформированной последовательности команд, которая называется *программой*.

АЛУ в составе серий цифровых микросхем выпускаются в основном для обработки четырехразрядных операндов. Это связано с числом внешних выводов корпуса, требуемых для подачи входных, управляющих сигналов и получения результата. Одной из таких микросхем является К155ИПЗ, изображаемая на схемах, как это показано на рисунке. У нее имеются входы операндов, а также вход и выход переноса, сигналы на которых воспринимаются и формируются в инверсном виде. Нулевой уровень на входе  $C_0$  соответствует его наличию, а единица — отсутствию. Так как аналогичная ситуация наблюдается и сигналом на выходе  $C_4$ , то для увеличения разрядности обрабатываемых чисел, АЛУ по этим входам и выходам можно соединять непосредственно.

Кроме этого, в АЛУ формируются функции генерации и распространения переноса, что позволяет для увеличения разрядности объединять их группами по четыре через схемы ускоренного переноса. Для выбора операции, выполняемой АЛУ, служат пять управляющих входов. Сигнал на входе  $M$  (*Mode*) переключает вид реализуемой функции (логическая либо арифметическая). Комбинируя сигналы на управляющих входах  $S_0 \div S_3$ , можно при  $M = 1$  выбрать любую из шестнадцати логических функций от двух переменных либо, если  $M = 0$ , шестнадцать вариантов арифметических и арифметико-логических операций, что отражено в таблице 1.

Таблица 1 – Перечень логических и арифметико-логических операций

Выбор функции				$M = 1$	$M = 0$	
$S_3$	$S_2$	$S_1$	$S_0$	Логические операции	Арифметические операции	
					$\overline{C_0} = 1$	$\overline{C_0} = 0$
0	0	0	0	$F_i = \overline{A_i}$	$F = A$	$F = A + 1$
0	0	0	1	$F_i = \overline{A_i \vee B_i}$	$F = A \vee B$	$F = (A \vee B) + 1$
0	0	1	0	$F_i = \overline{A_i} \wedge B_i$	$F = A \vee \overline{B}$	$F = (A \vee \overline{B}) + 1$
0	0	1	1	$F_i = 0$	$F = -1$	$F = 0$
0	1	0	0	$F_i = \overline{A_i} \wedge \overline{B_i}$	$F = A + (A \wedge \overline{B})$	$F = A + (A \wedge \overline{B}) + 1$
0	1	0	1	$F_i = \overline{B_i}$	$F = (A \vee B) + (A \vee \overline{B})$	$F = (A \vee B) + (A \vee \overline{B}) + 1$

Выбор функции				$M = 1$	$M = 0$	
$S_3$	$S_2$	$S_1$	$S_0$	Логические операции	Арифметические операции	
					$\overline{C_0} = 1$	$\overline{C_0} = 0$
0	1	1	0	$F_i = A_i \oplus B_i$	$F = A - B - 1$	$F = A - B$
0	1	1	1	$F_i = A_i \wedge \overline{B_i}$	$F = (A \wedge \overline{B}) - 1$	$F = A \wedge \overline{B}$
1	0	0	0	$F_i = \overline{A_i} \vee B_i$	$F = A + (A \wedge B)$	$F = A + (A \wedge B) + 1$
1	0	0	1	$F_i = \overline{A_i} \oplus \overline{B_i}$	$F = A + B$	$F = A + B + 1$
1	0	1	0	$F_i = B_i$	$F = (A \vee \overline{B}) + (A \wedge B)$	$F = (A \vee \overline{B}) + (A \wedge B) + 1$
1	0	1	1	$F_i = A_i \wedge B_i$	$F = (A \wedge B) - 1$	$F = A \wedge B$
1	1	0	0	$F_i = 1$	$F = A + A$	$F = A + A + 1$
1	1	0	1	$F_i = A_i \vee \overline{B_i}$	$F = (A \vee B) + A$	$F = (A \vee B) + A + 1$
1	1	1	0	$F_i = A_i \vee B_i$	$F = (A \vee \overline{B}) + A$	$F = (A \vee \overline{B}) + A + 1$
1	1	1	1	$F_i = A_i$	$F = A - 1$	$F = A$

Логические операции над данными осуществляются поразрядно, т. е. для каждой пары одноименных разрядов операндов. В этом случае четырехразрядные кодовые комбинации, поступающие на входы  $A_i$  и  $B_i$ , воспринимаются как наборы пар независимых логических переменных. Если, к примеру, задать операцию  $A_i \vee B_i$ , то произойдет поразрядное логическое сложение и результаты будут переданы на соответствующие выходы АЛУ. В данной ситуации разряды выходного кода не связаны друг с другом.

При реализации арифметических операций операнды  $A_i$  и  $B_i$  воспринимаются как двоичные коды чисел, и если реализуется сложение, то происходит арифметическое суммирование разрядов с учетом как входного, так и межразрядных переносов. Так как логическая единица на входе  $C_0$  АЛУ воспринимается как отсутствие, а ноль – как наличие переноса, то в зависимости от значения этого сигнала, итог арифметической операции будет отличаться на единицу. При этом положительные результаты формируются в прямом коде, а отрицательные в дополнительном.

Арифметико-логические (смешанные) операции реализуются АЛУ в два этапа. Сначала производится поразрядная логическая обработка операндов, а затем арифметическое действие с учетом переносов между разрядами. К примеру операция  $(A \vee B) + (A \vee \bar{B})$  при  $A = 0110$  и  $B = 0100$  будет выполняться следующим образом. Сначала сформируются функции  $A \vee B = 0110$  и  $A \vee \bar{B} = 1111$ , затем произойдет арифметическое сложение получившихся чисел и, в зависимости от состояния входа переноса, на выходах  $F$ - получится результат  $0101$  либо  $0110$ . Так как сумма в обоих случаях больше пятнадцати, то сигнал на выходе  $C_4$  примет нулевое значение.

В этом АЛУ имеется выход  $A = B$  с открытым коллектором, обозначенный двумя значками равенства. Сигнал логической единицы на нем будет формироваться при подключении внешнего резистора и равенстве чисел, поступающих на входы  $A$  и  $B$  АЛУ, если задана операция их вычитания.

### 3. Программа исследований и методические указания

Работа проводится на персональном компьютере с помощью программы схемотехнического моделирования.

Собрать на рабочем столе программы моделирования схему, приведенную на рисунке 2. Двоичные счетчики предназначены для формирования четырехразрядных операндов  $A_3A_2A_1A_0$  и  $B_3B_2B_1B_0$ . Открывая ключи «1» и «2», можно пропустить на входы каждого счетчика необходимое число импульсов генератора. Коды операндов отображаются шестнадцатиричными цифрами на семисегментных индикаторах. Ключи «3», «4», «5» и «6» служат для установки кода выполняемой функции  $S_3S_2S_1S_0$ . Ключ «7» служит для установки режима, а ключ «8» устанавливает сигнал входного переноса при выполнении арифметической операции: логический «0», если перенос есть, и логическая «1», если перенос отсутствует.

Результат выполненной операции отображается на семисегментном индикаторе, светодиод сигнализирует выходной перенос: «0» – наличие переноса, «1» – отсутствие.

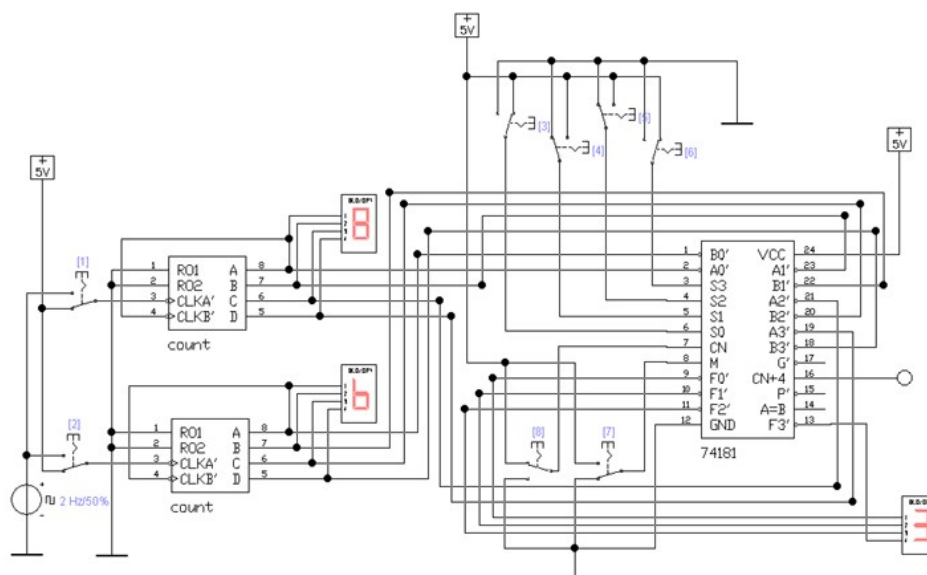


Рисунок 2 – Схема установки для исследования АЛУ

#### 1. Арифметическая операция сложения

Установить режим  $M = 0$ , код операции  $S_3S_2S_1S_0 = 1001$ . Сформировать на выходах первого и второго счетчиков произвольные числа, например,

$$A_3A_2A_1A_0 = 1001 = 9_{(10)}; B_3B_2B_1B_0 = 1101 = 13_{(10)} = d_{(16)}.$$

Ожидаемый результат в отсутствие входного переноса ( $C_N = 1$ ) равен

$$F_3F_2F_1F_0 = 0110 = 6_{(10)}.$$

Кроме того, должен появиться сигнал выходного переноса ( $C_{N+4} = 0$ ).

Скопировать для отчета изображение на экране (см. рис. 2).

Не изменяя операнды, установить входной перенос ( $C_N = 0$ ) и зафиксировать результат операции. Скопировать изображение на экране для отчета.

## **2. Арифметическая операция вычитания**

### **2.1. Вычитаемое меньше уменьшаемого**

Установить в режиме  $M = 0$  код операции  $S_3S_2S_1S_0 = 0110$ . Сформировать на выходах первого и второго счетчиков произвольные числа  $A_3A_2A_1A_0$  и  $B_3B_2B_1B_0$ , но так, чтобы  $A > B$ , например,

$$A_3A_2A_1A_0 = 1110 = 14_{(10)} = E_{(16)}; B_3B_2B_1B_0 = 1011 = 11_{(10)} = B_{(16)}.$$

Ожидаемый результат в прямом коде в отсутствие входного переноса ( $C_N = 1$ ) равен

$$F_3F_2F_1F_0 = 0011 = 3_{(10)}.$$

Сигнал выходного переноса должен отсутствовать ( $C_{N+4} = 1$ ).

Скопировать для отчета изображение на экране (см. рис. 2).

Не изменяя операнды, установить входной перенос ( $C_N = 0$ ) и зафиксировать результат операции. Снять изображение на экране для отчета.

### **2.2. Вычитаемое больше уменьшаемого**

Установить в режиме  $M = 0$  код операции  $S_3S_2S_1S_0 = 0110$ . Сформировать на выходах первого и второго счетчиков произвольные числа  $A_3A_2A_1A_0$  и  $B_3B_2B_1B_0$ , но так, чтобы  $A < B$ , например,

$$A_3A_2A_1A_0 = 0111 = 7_{(10)} = E_{(16)}; B_3B_2B_1B_0 = 1011 = 11_{(10)} = B_{(16)}.$$

Ожидаемый результат в дополнительном коде в отсутствие входного переноса ( $C_N = 1$ ) равен

$$F_3F_2F_1F_0 = \text{минус } 1100 = \text{минус } 12_{(10)}.$$

При этом должен присутствовать сигнал выходного переноса ( $C_{N+4} = 0$ ).

Скопировать для отчета изображение на экране (см. рис. 2).

Не изменяя операнды, установить входной перенос ( $C_N = 0$ ) и зафиксировать результат операции. Снять изображение на экране для отчета.

## **3. Логическая операция**

Установить режим  $M = 1$  и код операции  $S_3S_2S_1S_0$ , который соответствует функции, заданной преподавателем. Сформировать в первом и втором счетчиках произвольные наборы переменных  $A_3A_2A_1A_0$  и  $B_3B_2B_1B_0$ . Выполнить операцию и сравнить результат с ожидаемым выходным словом  $F_3F_2F_1F_0$ . Скопировать для отчета изображение на экране (см. рис. 2).

## **4. Контрольные вопросы**

1. К какой категории цифровых устройств, – комбинационного типа или автоматам с памятью, – относятся арифметико-логические устройства?

2. Какой функциональный узел является основой схемы АЛУ?

3. В каком порядке выполняются действия сложной арифметико-логической операции?

## **5. Содержание отчета**

Отчет должен содержать:

- 1) титульный лист;
- 2) наименование работы и цель исследований;
- 3) задание на лабораторную работу;
- 4) описание методики выполнения логических и арифметических функций, реализованных в АЛУ;
- 5) копии изображений на экране при выполнении заданных операций;
- 6) комментарии полученных результатов.

## **Библиографический список**

1. Иванов, В.И. Проектирование цифровых устройств [Текст]: Учебное пособие / В.И. Иванов, В.С. Титов, М.В. Бобырь и др.; Юго-Зап. гос. ун-т. Курск, 2011. 100 с.
2. Угрюмов, Е.П. Цифровая схемотехника [Текст]: учебное пособие для вузов / Е.П. Угрюмов. – СПб: БХВ-Петербург, 2004. 528 с.
3. Угрюмов, Е.П. Проектирование элементов и узлов ЭВМ [Текст]: учебное пособие для вузов / Е.П. Угрюмов. – М.: Высш. шк., 1987. 318 с.

# ТРИГГЕРЫ С ДИНАМИЧЕСКИМ УПРАВЛЕНИЕМ

## 1. Цель работы

Изучить логику работы синхронного триггера с динамическим управлением записью; получить навыки в проектировании, сборке, наладке и экспериментальном исследовании заданного типа триггера.

## 2. Теоретические сведения и методика проектирования

На практике для построения синхронных триггеров с динамическим управлением используется устройство внутренней задержки на основе схемы “трех триггеров”, представленной на рис.1.

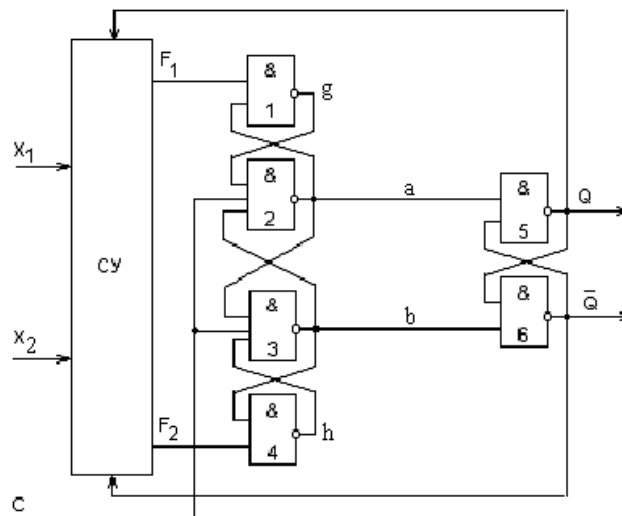


Рис. 1. Схема триггера с динамическим управлением

Рассмотрим функционирование базового триггера, выполненного на вентилях 1... 6. Элементы 1...4 образуют схему управления, а элементы 5 и 6 составляют запоминающий элемент. Переход триггера в новое состояние в этой схеме происходит по положительному фронту тактирующего импульса в соответствии с входными сигналами на информационных входах  $F_1$  и  $F_2$ . Вентили 1 и 4 выполняют функцию инверторов информационных сигналов.

При  $C = 0$  на выходах вентилях 2 и 3 (шины  $a$  и  $b$ ) присутствует сигнал логической 1, и триггер на элементах 5 и 6 не изменяет своего состояния при любых сигналах на входах  $F_1$  и  $F_2$ .

Для записи в триггер состояния  $Q = 1$  необходимо подать на информационные входы комбинацию  $F_1 = 0, F_2 = 1$ . Тогда на выходе вентиля 1 появится сигнал  $g = 1$ , который при  $C = 1$  обеспечивает на выходе вентиля 2 сигнал  $a = 0$ , устанавливающий основной триггер (вентили 5 и 6) в единичное состояние и подтверждающий единичный сигнал на выходе  $g$  вентиля 1.

После этого сигналы  $F_1$  и  $F_2$  могут изменять свое значение, но это не повлияет на состояние основного триггера до тех пор, пока не осуществится очередной переход сигнала  $C$  из 0 в 1.

Для записи в триггер состояния  $Q = 0$  необходимо подать на информационные входы комбинацию  $F_1 = 1, F_2 = 0$ . Тогда на выходе вентиля 4 появится сигнал  $h = 1$ , который при  $C = 1$  обеспечивает на выходе вентиля 3 сигнал  $b = 0$ , устанавливающий основной триггер (вентили 5 и 6) в нулевое состояние и подтверждающий единичный сигнал на выходе  $h$  вентиля 4.

Если на обоих информационных входах установлены сигналы  $F_1 = 1, F_2 = 1$ , то на выходах  $g$  и  $h$  вентилях 1 и 4 будут сигналы нулевого уровня, и при  $C = 1$  не появятся ни

$a = 0$ , ни  $b = 0$ , что обеспечивает режим хранения старого состояния. Комбинация входных сигналов  $F_1 = 0, F_2 = 0$  не допустима, так как произойдет непредсказуемое переключение триггера.

Таким образом, логика работы этой схемы соответствует функционированию триггера RS – типа с инверсными информационными входами.

Особенность и достоинство триггера с динамическим управлением состоит в том, что он допускает изменение информационных сигналов без появления ошибок при обоих уровнях тактового сигнала (нулевом и единичном), а активным, то есть обеспечивающим переключение, тактовый сигнал является лишь в момент перепада (фронта). Таким образом, в триггер записываются данные, поданные на информационные входы  $F_1$  и  $F_2$  в момент активного фронта синхросигнала. Однако вследствие конечного быстродействия элементов схемы интервал времени, в течение которого триггер чувствителен к изменениям информационных сигналов, имеет ненулевую длительность. Для приведенной схемы необходимо условие постоянства информационных сигналов до прихода синхроимпульса на интервале, равном задержке переключения логического элемента 1 (или 4), и интервале, равном задержке переключения логического элемента 2 (или 3), после чего схема зафиксирует входные данные с помощью сигналов  $a$  или  $b$ .

На основе рассмотренной схемы “трех триггеров” можно построить триггер с любой логикой работы. Проектирование произвольной триггерной структуры на основе схемы “трех триггеров” заключается в синтезе схемы управления.

Процедура проектирования схемы управления осуществляется в следующей последовательности:

1. Определение значений  $F_1$  и  $F_2$  для реализации заданной последовательности переходов.
2. Построение таблицы переходов.
3. Синтез и минимизация функций  $F_1$  и  $F_2$ .
4. Приведение функций  $F_1$  и  $F_2$  в заданный логический базис.

На первом этапе определяют, при каких значениях  $F_1$  и  $F_2$  запоминающий элемент (вентили 5 и 6) осуществляет переход из одного состояния в другое. Затем строят полную таблицу переходов заданного триггера с учетом структуры и типа логических элементов. На основании определенного ранее порядка переходов запоминающего элемента заполняют в этой таблице графы для  $F_1$  и  $F_2$ , анализируя переходы из  $Q^t$  в  $Q^{t+1}$  в каждой строке таблицы. С помощью карт Карно (диаграмм Вейча) находят минимальную форму функций  $F_1$  и  $F_2$ , приводят полученные функции к виду, удобному для реализации на элементах И-НЕ. После этого производится построение общей схемы триггера.

### 3. Подготовка к выполнению работы

Разработать комбинационную схему управления, обеспечивающую переключения триггера согласно заданию. Варианты заданий приведены в табл. 1

Таблица 1. Варианты заданий

X1	X2	Значения выхода для соответствующего варианта													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	1	0	1	$\bar{Q}^t$	x	0	$Q^t$	$Q^t$	1	$Q^t$	x	1	$Q^t$
0	1	1	0	$Q^t$	0	x	1	x	1	$\bar{Q}^t$	$Q^t$	1	$Q^t$	0	x
1	0	$\bar{Q}^t$	0	$\bar{Q}^t$	$\bar{Q}^t$	1	0	1	$\bar{Q}^t$	1	$\bar{Q}^t$	x	0	$\bar{Q}^t$	1
1	1	0	$\bar{Q}^t$	0	x	0	$\bar{Q}^t$	$\bar{Q}^t$	0	0	1	$\bar{Q}^t$	$\bar{Q}^t$	1	$\bar{Q}^t$

Продолжение таблицы 1

X1	X2	Значения выхода для соответствующего варианта															
		15	16	17	18	19	20	21	22	23	24	25	26	27	28		



0	0	1	$\bar{Q}^t$	1	0	1	1	0	0	1	0	x	$\bar{Q}^t$	$Q^t$	$\bar{Q}^t$
0	1	$\bar{Q}^t$	0	$\bar{Q}^t$	$\bar{Q}^t$	$\bar{Q}^t$	$\bar{Q}^t$	$\bar{Q}^t$	$\bar{Q}^t$	x	1	0	0	x	1
1	0	$Q^t$	1	$Q^t$	$Q^t$	0	0	$Q^t$	1	0	0	1	1	$\bar{Q}^t$	$Q^t$
1	1	1	x	x	0	$\bar{Q}^t$	0	x	x	$\bar{Q}^t$	$\bar{Q}^t$	$\bar{Q}^t$	x	1	0

Пусть требуется реализовать триггер, работа которого описывается табл.2.

Таблица 2. Таблица переходов триггера

Входные сигналы в момент $t$		Новое состояние триггера в момент $t+1$
$X_1$	$X_2$	$Q^{t+1}$
0	0	0
0	1	$\bar{Q}^t$
1	0	1
1	1	x

В соответствии с описанной ранее методикой проведем синтез схемы управления триггером.

1. Определим, при каких значениях функций  $F_1$  и  $F_2$  запоминающий элемент осуществляет переходы из одного состояния в другое. Так как в схеме в качестве запоминающего элемента используется RS-триггер на элементах И-НЕ, его порядок переходов имеет следующий вид:

из 0 в 0 при  $F_1 = 1; F_2 = *$ ;

из 0 в 1 при  $F_1 = 0; F_2 = 1$ ;

из 1 в 0 при  $F_1 = 1; F_2 = 0$ ;

из 1 в 1 при  $F_1 = *; F_2 = 1$ .

Переключение триггера из одного состояния в другое происходит при подаче на вход С положительного фронта синхросигнала.

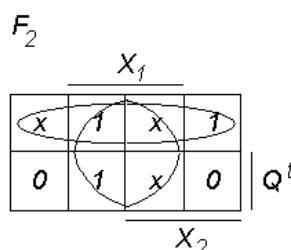
2. Построим полную таблицу переходов триггера (табл.3).

Таблица 3

$t$			$t+1$	$F_1$	$F_2$
$X_1$	$X_2$	$Q^t$	$Q^{t+1}$		
0	0	0	0	1	x
0	0	1	0	1	0
0	1	0	1	0	1
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	1	x	1
1	1	0	x	x	x
1	1	1	x	x	x

Столбцы  $F_1$  и  $F_2$  заполняются на основе полученного ранее порядка переходов триггера из одного состояния в другое.

3. Заполняем диаграммы (рис. 2).



Вейча для функций  $F_1$  и  $F_2$

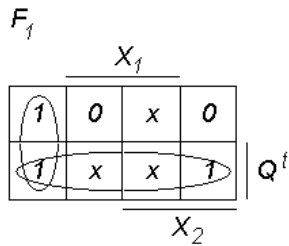


Рис. 2. Диаграммы Вейча для функций  $F_1$  и  $F_2$

3. С помощью диаграмм Вейча находим минимальную форму функций  $F_1$  и  $F_2$ :

$$F_1 = Q^t \vee \bar{X}_1 \bar{X}_2 ; \quad F_2 = \bar{Q}^t \vee X_1 .$$

4. Приводим полученные функции к виду, удобному для реализации на элементах И-НЕ:

$$F_1 = \overline{\bar{Q}^t \wedge \bar{X}_1 \bar{X}_2} ; \quad F_2 = \overline{Q^t \wedge \bar{X}_1} .$$

5. В соответствии с полученными уравнениями для функций  $F_1$  и  $F_2$  строим схему триггера (рис.3).

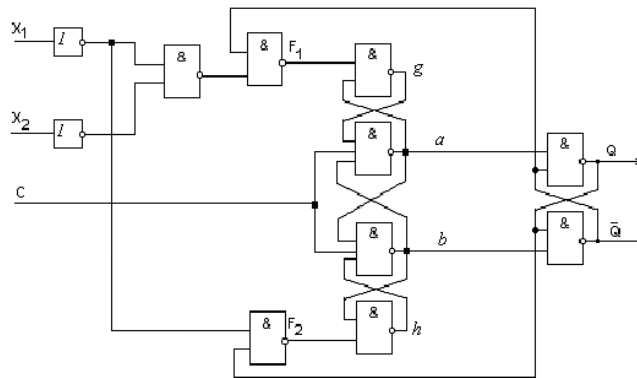


Рис. 3. Схема триггера на элементах И-НЕ

#### 4. Порядок выполнения работы

Собрать на рабочем поле программы электронного моделирования схему для исследования триггера (рис. 4). Комбинации входных сигналов  $X_1$  и  $X_2$  генерирует двухразрядный двоичный счетчик на триггерах типа *JK* с прямыми асинхронными входами и инверсным синхровходом из магазина **Digital**. В свойствах триггера установить библиотеку **t1l** и модель **LS** (ТТЛШ). Установить частоту тактовых импульсов 1 МГц.

Временные диаграммы в исследуемом триггере зависят от порядка чередования комбинаций сигналов  $X_1$  и  $X_2$ . Для определенности выбираем цикл из четырех тактов, в котором переменная  $X_1$  ведет себя как старший разряд, а  $X_2$  – как младший разряд двоичного кода, вырабатываемого счетчиком. Счетчик переключается по отрицательным фронтам тактовых импульсов.

Все логические элементы, используемые в схеме, выбрать из библиотеки **t1l**, модели **LS** (ТТЛШ).

Ключ **Switch** из магазина **Basic**, управляемый клавишей **Space** (Пробел), необходим для принудительного сброса триггера в нулевое состояние перед началом моделирования. Для подключения этого ключа к схеме у некоторых элементов потребовались дополнительные входы.

Исследование функционирования триггера производится в динамическом режиме с помощью логического анализатора.

Выполнить следующие установки в блоке **Clock Set...** на панели анализатора:

**Clock Mode – Internal** (задает внутренний режим запуска от самих исследуемых сигналов);

**Internal clock rate – 10 МГц** (частота тактовых импульсов);

**Threshold voltage (V) – 1,3** (порог срабатывания анализатора).

Остальные установки оставить по умолчанию и нажать кнопку **Accept**.

Запустить программу моделирования и дважды нажать клавишу **Space (Пробел)**, подключив на короткое время входы триггера к общей шине. Когда на экране анализатора заполнятся диаграммы напряжений, остановить процесс моделирования кнопкой **Pause**. Выбрать подходящий масштаб по оси времени с помощью установки **Clocks per division**, так чтобы на развертке укладывался полный цикл работы преобразователя.

Зарисовать для отчета временные диаграммы сигналов во всех точках схемы триггера. Пример диаграмм показан на рис. 5.

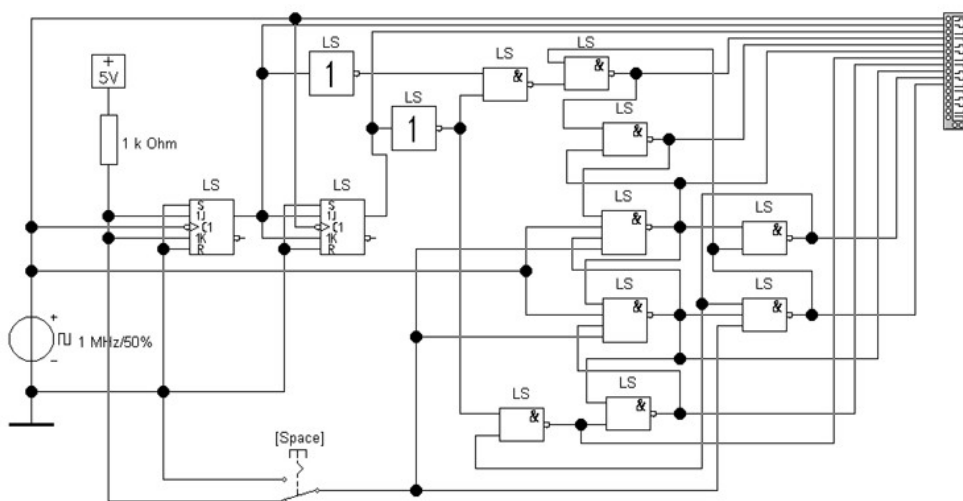


Рис. 4. Схема для исследования триггера с динамическим управлением

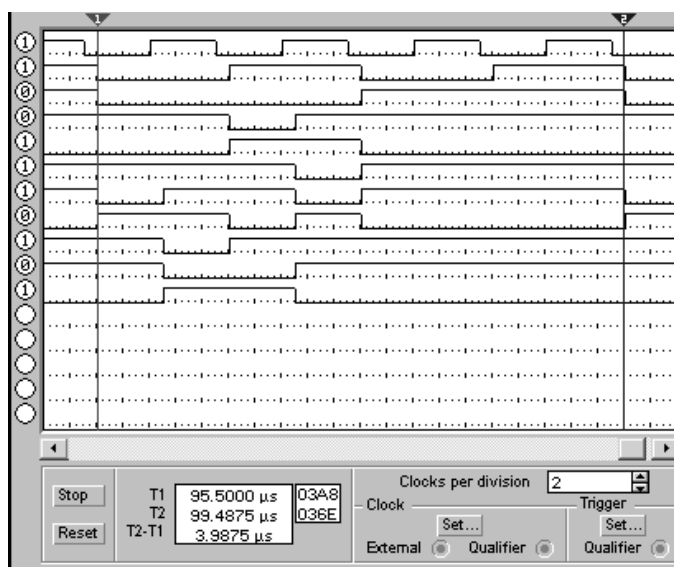


Рис. 5. Временные диаграммы в триггере:

1 – синхроимпульсы; 2 – сигнал  $X_2$ ; 3 – сигнал  $X_1$ ; 4 – функция  $F_1$ ; 5 – выход  $g$ ; 6 – выход  $a$ ; 7 – функция  $F_2$ ; 8 – выход  $h$ ; 9 – выход  $b$ ; 10 – прямой выход  $Q$ ; 11 – инверсный выход  $\bar{Q}$

## 5. Контрольные вопросы

1. В чем заключается принцип динамического управления записью информации в триггер?
2. Каким образом происходит запоминание данных на информационных входах базового триггера на основе схемы “трех триггеров”, поступивших перед приходом активного фронта синхроимпульса?
3. Какую функцию выполняют сигналы  $a$  и  $b$  в базовой схеме “трех триггеров”?
4. Каково назначение дополнительной комбинационной схемы управления базовым триггером?
5. Какими цепями схемы определяется минимальное время предустановки сигналов на информационных входах  $X_1$  и  $X_2$  до прихода активного фронта синхроимпульса?

## **6. Содержание отчета**

Отчет должен содержать:

- 1) титульный лист;
- 2) наименование работы и цель исследований;
- 3) результаты разработки схемы управления триггером (таблица переходов триггера, синтез и минимизация функций  $F_1$  и  $F_2$ , схема триггера в целом);
- 4) результаты исследования работы триггера в динамическом режиме (временные диаграммы входных и выходных сигналов и напряжений в других контрольных точках схемы).

## **Библиографический список**

1. Угрюмов Е.П. Цифровая схемотехника: Учебное пособие. – СПб.: БХВ-Петербург, 2001. – 528 с.
2. Угрюмов Е.П. Проектирование элементов и узлов ЭВМ: Учеб. пособие для вузов. – М.: Высш. шк., 1987. – 318 с.
3. Иванов В.И. Проектирование цифровых устройств: учебное пособие / В.И. Иванов, В.С. Титов, М.В. Бобырь, А.С. Ястребов; Юго-Зап. гос. ун-т. Курск, 2011. 100 с.