

Министерство науки и высшего образования РФ
Федеральное государственное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

ЭЛЕКТРОТЕХНИКА, ЭЛЕКТРОНИКА И СХЕМОТЕХНИКА

сборник учебно-методических материалов
для направления подготовки 10.03.01
«Информационная безопасность»

Благовещенск 2019

Печатается по решению
Редакционно-издательского совета
энергетического факультета
Амурского государственного университета

Составитель: Теличенко Д.А.

Электротехника, электроника и схемотехника: сборник учебно-методических материалов для направления подготовки 10.03.01 – Информационная безопасность. – Благовещенск: Амурский гос. ун-т, 2019.

©Амурский государственный университет, 2019
©Кафедра автоматизации производственных
процессов и электротехники, 2019
©Теличенко Д.А.

Методические указания по освоению дисциплины

Изучение дисциплины студентами должно начинаться со знакомства с рабочей программой и выдаваемыми материалами: методическими пособиями и литературой в электронном формате, а также учебно-методическим комплексом для студентов (далее УМКД, формируемого на основе сборника учебно-методических материалов по дисциплине), в случае необходимости. УМКД это отдельно сформированный документ в электронном варианте содержащий абсолютно все необходимое, включая рабочую программу, методические рекомендации, пояснения по работе с программным обеспечением и специальными средствами и стендами, задания для самостоятельного выполнения и пр.

Весь материал предварительно размещается на сайте и постоянно доступен, в том числе и в твердой копии на кафедре.

На первом занятии студенты обзорно знакомятся с планом проведения и методикой занятий, узнают конкретные требования к изучению дисциплины, им даются рекомендации, представленные здесь.

Студентам необходимо помнить, что качественная текущая подготовка и проработка материала является залогом успешного освоения предмета.

Студентам рекомендуется за один день до проведения соответствующих занятий познакомиться с планом работ, изучить рассматриваемые вопросы по рекомендуемой литературе и выполнить пункты самостоятельной работы.

После проведения занятий, в этот же день, требуется повторить изученные теоретические положения, выполнить необходимые расчеты и примеры домашних заданий (по факту выдачи). При повторении материала желательно охватывать ранее рассмотренные вопросы; сначала более детально, затем ближе к концу семестра – обзорно.

Такая методика позволяет глубоко проработать все вопросы и не оставляет пробелы в знаниях. В итоге, к окончанию семестра, имеющиеся комплексные знания потребуются лишь освежить в памяти за 2-3 дня до итогового контроля (зачета или экзамена).

Для подготовки к занятиям следует пользоваться литературой, указанной в разделе 10 рабочей программы, в том числе и электронным комплектом материалов. Для общей теоретической подготовки рекомендуется использовать источники п.10.1; при этом вспомогательными источниками п.10.2 необходимо пользоваться по мере необходимости. Для подготовки к практическим, самостоятельным и лабораторным работам рекомендуется использовать пособия, указанные ниже.

Привила проведения аттестации по результатам освоения дисциплины представлены в п.9 рабочей программы. Необходимо помнить при этом, что основой аттестационной оценки является результаты выполнения индивидуальных работ: лабораторных, домашних заданий, РГР – самостоятельно и с полным осознанием выполненных процедур и их результатов.

Методические указания к лабораторным работам

Лабораторные работы выполняются по методическим пособиям к имеющимся стендам, см. п.2 и п.3 перечня учебно-методического обеспечения раздела 7 рабочей программы):

а) Бородянко В.Н., Непопалов В.Н., Шулдяков В.В. Электрические цепи: Методические указания к проведению лабораторных работ на минимодульном стенде «Электрические цепи и основы электроники». – Челябинск: Учтех-Профи, 2016;

а') Бородянко В.Н., Гельман М.В., Непопалов В.Н., Шулдяков В.В. Электрические цепи и основы электроники: Методические указания к проведению лабораторных работ на минимодульной части стенда «Электротехника и основы электроники». – Челябинск: Учтех-Профи, 2016;

б) Гельман М.В., Шулдяков В.В., Верцюх А.С. Основы электроники: Методические указания к проведению лабораторных работ по электронике на минимодульном стенде «Электрические цепи и основы электроники». – Челябинск: Учтех-Профи, 2016;

б') Гельман М.В., Шулдяков В.В., Кучурин А.А., Деев А.С. Электрические цепи и основы электроники: Методические указания к проведению лабораторных работ на минимодульной части стенда «Электротехника и основы электроники». – Челябинск: Учтех-Профи, 2016.

в) Теличенко Д.А., Бушманов А.В. Схемотехника. Лабораторный практикум. Благовещенск: Амурский гос. ун-т, 2006, 93 с.

При этом пособия «а» и «б» являются основными (на них делается ссылка ниже, в таблице распределения работ), а отмеченные «'» – вспомогательные (используемые при проведении компьютеризированных занятий).

Преподавателем до проведения занятий формулируются конкретные требования и порядок выполнения и защиты работ.

Возможно несколько вариантов проведения и защиты работ (конкретный выбор закрепляется за преподавателем, ведущим лабораторные работы и доводится до студентов не позднее второго занятия).

Вне зависимости от варианта проведения лабораторных работ студенты делятся на группы (количество которых определяется по доступными на текущий момент стендам и в соответствии с тем чтобы максимальное число студентов в группе не мешало выполнению работы; рекомендуются группы до 5 человек; рекомендуется группы формировать так что бы уровень подготовки студентов в одной группе был одинаков – определяется, например, по результатам входного тестирования).

Вариант А (базовый). В соответствии с п.6.3 рабочей программы и таблицей, представленной ниже на каждую тему, выделяется отведенное количество часов. Здесь на каждом занятии студенты как выполняют саму работу, так и защищают ее. Данный вариант предполагает: предварительную подготовку заготовок отчетов и выполнение всех заданий непосредственно на самом занятии, одновременно с защитой самой работы. В этом случае вначале занятий преподавателем проверяется подготовка группы к выполнению работы (опрос по теории, просмотр необходимых заготовок; в случае отсутствия подготовки группа или отдельные лица отстраняются от выполнения работы). Затем осуществляется допуск к работе и выполнение работы группой и заполнение заготовки отчетов. В конце занятия, осуществляется защита работы группой или отдельными лицами. Защита предусматривает обсуждение соответствующей теории, ответы на контрольные вопросы, проверку отчетов и выполненных заданий, проверку выводов и глубины усвоения материала в конкретной теме. Предпочтительнее осуществлять индивидуальный допуск и защиту работ.

Вариант Б (усовершенствованный). На всех занятиях отдельно выделяются чередующиеся циклы снятия (выполнения) и защиты работ. При этом в рамках одного занятия (пары – академических 2 часа) проводится только один цикл – снятие или защита работы. В этом случае предусматривается объединение в одно занятие нескольких тем. Порядок допуска и защиты идентичен варианту А.

Вариант В (модифицированный). В данном случае аналогично варианту Б имеются циклы выполнения работ (снятие и защита). Однако здесь в рамках одного занятия отдельные группы снимают работу, другие – защищают. Порядок групп и вида выполняемых ими работ определяется преподавателем.

Примечание: выполнение работ так же предполагает, что для определенной группы студентов возможна выдача дополнительных тем – отмеченных *; либо наоборот – сокращения объема проводимых исследований для определенного контингента обучающихся.

Вне зависимости от выбранного варианта проведения лабораторных занятий студентам предварительно предлагается самостоятельно ознакомиться с краткой теорией к каж-

дой выполняемой работе и выполнять задания самостоятельно. Это даст необходимую теоретическую основу и облегчит выполнение работ, позволив на занятии уделить большее внимание вопросам, обычно вызывающим наибольшее затруднение. Аналогично так же для каждой темы предполагается проведение трех этапов – допуска, выполнения и защиты.

Допуск к работе и ее выполнение:

- преподавателем осуществляется допуск к работе, на котором проверяется: знание студентов краткой теории по выполняемой работе; наличие заготовки отчета;
- выясняются вопросы, вызвавшие у студентов затруднения, даются необходимые пояснения по ним;
- даются комментарии по методике проведения экспериментов;
- контролируется выполнение работы каждой бригады и всеми студентами в целом.

Работа считается снятой, если: студенты одной бригады, и каждый в отдельности, выполнили все задания работы, согласно вариантам; зафиксировали снятые данные в заготовку отчета.

При выполнении работ на стендах так же обязательным этапом является после допуска к работе, до подачи питания – демонстрация собранных схем преподавателю для проверки правильности и выполнения условий безопасности.

Защита работы:

- преподавателем, каждому из студентов, выдается произвольный вариант необходимый для выполнения упражнений (если предусмотрено);
- каждый из студентов лично выполняет упражнения (если предусмотрено), согласно выданному на данной работе варианту (в случае если работа не защищается на одном занятии, варианты на упражнения изменяются);
- преподавателем проверяется личный отчет каждого из студентов, задаются вопросы по ходу выполнения работы; задаются контрольные вопросы (список вопросов приведен в лабораторном практикуме к каждой работе).

Работа считается защищенной, если: правильно выполнен отчет по работе; даны корректные ответы на вопросы преподавателя; правильно выполнены упражнения (если предусмотрены).

Представленный отчет (после успешной защиты работы отчет сдается преподавателю и сохраняется до успешной сдачи студентом экзамена) должен удовлетворять следующим требованиям:

- отчет выполняется на одной стороне белого листа формата А4 в рукописной или печатной форме, в варианте возможном для прочтения (почерк, шрифт, размер, интервал);
- титульный лист должен содержать следующие сведения: название предмета; тему работы, с ее порядковым номером; фамилию студента, выполнившего работу с указанием номера группы и вариантов (личного и на бригаду); фамилию преподавателя, осуществляющего прием работы; дату снятия и защиты (дата защиты заполняется преподавателем лично).
- основная часть работы должна содержать следующие сведения: краткую теорию; цель работы; элементы, приборы и инструменты, используемые в работе; ход работы с необходимыми рисунками, схемами, таблицами, формулами и пояснениями.

В случае если студент не снял или не защитил работу, он может приступить к следующей работе. Ликвидировать возникшую задолженность можно на оставшемся времени после проведения очередной лабораторной работы или на дополнительных занятиях. Если ликвидировать задолженность по лабораторным работам в течение семестра не удастся, студент является на экзамен с отчетами по несданным работам, где ему до ответа на экзаменационные вопросы дается возможность защитить каждую работу.

План проведения занятий с указанием последовательности изучаемых тем, объема часов, представлен в виде таблиц. При этом в последней колонке представлены ссылки на номера работ лабораторных пособий. Проведение работ предполагает гибкое распределе-

ние тем, с учетом успехов студентов при освоении дисциплины, и как было указано выше может гибко меняться как в одну сторону, так и в другую – это не уменьшит объема полученных знаний, так как соответствующие темы отрабатываются на лекциях и практиках.

Методические указания к практическим работам

Предварительно студенты знакомятся со списком всех изучаемых тем, рассматриваемых на практических занятиях. Характер вопросов, прорабатываемых здесь, связан с лекционным курсом и графиком самостоятельной работы. Большая часть времени уделяется решению конкретных задач (аналогичные по тематике задачи использованы в качестве экзаменационного вопроса). Эти же задачи частично входят в домашние задания (РГР).

Задачи решаются студентами самостоятельно. При этом один из студентов вызывается к доске, решает поставленную задачу. Преподавателем контролируется не только правильность решения, но и даются: практические рекомендации по выполнению подобных заданий, применимость рассматриваемых тем к практике, а также предлагается другим студентам предложить альтернативные способы решения. Каждому вышедшему к доске, а также студентам, принявшим участие в обсуждении, выставляется оценка.

План проведения практических занятий, включая темы, объем часов представлен выше.

Методическое обеспечение данного вида работ представлено в приложении Б.

Методические указания по самостоятельной работе

Самостоятельная работа предполагает проработку вопросов теории и практики в соответствии с учебной программой.

Основная часть работы здесь связана с подготовкой к лабораторным и практическим работам – см выше, а также выполнении РГР.

Индивидуальные практические задания (РГР) предусмотрены только для главы 3 «Схемотехника». Сами задания изложены в формируемом отдельно УМКД или базируются на имеющемся в свободном доступе методическом обеспечении (представлены так же в приложении Б). Все задания РГР выполняются письменно персонально каждым студентом и оформляются в соответствии с действующими стандартами АмГУ на листах белой бумаги формата А4 (рукописным или печатным способом) в варианте возможным для прочтения.

В общем случае РГР состоит из следующих разделов: титульный лист (со всеми данными касающимися выполнившего работу студента, его варианта задания и пр.); задания к работе; решение соответствующих заданий (включая все необходимые выкладки и пояснения по их решениям, расчеты, доказательства и построения, графические иллюстрации, таблицы и пр.).

Защита РГР предполагает индивидуальную беседу со студентом, где выясняется уровень освоения материала, аутентичность работы с точки зрения авторства. Защита работы проводится в рамках итоговой аттестации по дисциплине либо на консультациях перед экзаменом, либо на самом экзамене.

Подготовка к защите и выполнение работы предполагает выполнение заданий осуществляемых в рамках практических работ по главе 3 (см. выше) и содержит только те задания, которые студенты выполняют на практиках; аналогичные задания имеются в экзаменационных билетах.

УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ЛАБОРАТОРНЫХ РАБОТ

ЛАБОРАТОРНАЯ РАБОТА № 1

ИЗУЧЕНИЕ ЛОГИЧЕСКИХ СХЕМ И ФУНКЦИЙ

Цель работы.

1. Исследование базовых логических элементов.
2. Реализация логических функций при помощи логических элементов.
3. Синтез логических схем.

Приборы и элементы.

Логический преобразователь (панель «Instruments/Logic Converter»).

Генератор слов (панель «Instruments/Word Generator»).

Вольтметр (панель «Indicators/Voltmeter»).

Логические пробники (панель «Indicators/Red probe»).

Источник напряжения + 5 В (панель «Basic/Pull-Up Resistor»).

Земля (панель «Sources/Ground»).

Двухпозиционные переключатели (панель «Basic/Switch»).

Двухвходовые элементы И, И-НЕ, ИЛИ, ИЛИ-НЕ (панель «Logic Gates/2-Input AND, NAND, OR, NOR Gates»).

Микросхемы различных серий (панель «Logic Gates/»).

Краткие теоретические сведения

Любые цифровые микросхемы современных вычислительных машин строятся на основе простейших логических элементов «И», «ИЛИ», «НЕ» и их комбинаций. В настоящее время используется несколько технологий построения логических элементов:

транзисторно-транзисторная логика (ТТЛ, TTL);

логика на основе комплементарных МОП транзисторов (КМОП, CMOS);

логика на основе сочетания комплементарных МОП и биполярных транзисторов (БиКМОП);

эмиттерно-связанная логика (ЭСЛ) и т.д.

Прежде чем приступить к изучению базовых логических элементов, введем основные понятия булевой алгебры, или алгебры логики, на которой базируется все теоретическое обоснование работы цифровых устройств.

1. Основные определения и аксиомы алгебры логики

Переменные, рассматриваемые в алгебре логики, принимают только два значения – «0» или «1». Чаще всего сами переменные обозначаются латинскими буквами: либо малыми (x, y, z, \dots), либо большими (A, B, C, \dots).

Примечание. Физически самым простым устройством, моделирующим поведение любой булевой переменной, является двухпозиционный переключатель. У него за логический сигнал «1» принимается положение «включено», а за сигнал – «0» положение «выключено». Поэтому самой простой схемой для изучения логических функций и выражений является электрическая, состоящая из исследуемого элемента или элементов, источников питания, заземления и устройств, моделирующих заданные уровни логических сигналов (в простейшем случае ими являются переключатели). Причем в данном случае важно подчеркнуть неразрывную связь между физическим сигналом (ток или напряжение) и логическим сигналом: за логический сигнал «0» и «1» принимаются определенные уровни физических сигналов (чаще всего за логический «0» принимается сигнал по напряжению в 0 В, а за уровень логической «1» +5 В).

Перечислим основные операции и аксиомы алгебры логики.

Основные операции алгебры логики:

отношение эквивалентности, обозначаемое знаком « \Rightarrow »;

операция логического сложения (дизъюнкция), обозначаемая знаком « \vee » или « $+$ »;

операция логического умножения (конъюнкция), обозначаемая знаком « \wedge » или « \bullet »;

операция отрицания (инверсии), обозначаемая надчеркиванием или апострофом « $'$ ».

Основные аксиомы булевой алгебры:

$$\left\{ \begin{array}{l} \bar{0} = 1, \\ \bar{1} = 0. \end{array} \right. \quad \left\{ \begin{array}{l} 0 \vee 0 = 0 + 0 = 0, \\ 1 \vee 0 = 1 + 0 = 1, \\ 1 \vee 1 = 1 + 1 = 1. \end{array} \right. \quad \left\{ \begin{array}{l} 0 \wedge 0 = 0 \cdot 0 = 0, \\ 0 \wedge 1 = 0 \cdot 1 = 0, \\ 1 \wedge 1 = 1 \cdot 1 = 1. \end{array} \right. \quad (1)$$

2. Логические выражения

Из логических переменных с помощью базовых логических операций можно составить логическое выражение. Логические выражения связывают значение логической функции со значениями логических переменных.

Логическое выражение является одним из способов описания цифрового устройства.

Запись логических выражений обычно осуществляют в конъюнктивной или дизъюнктивной нормальных формах. В дизъюнктивной форме логические выражения записываются как логическая сумма логических произведений, в конъюнктивной форме – как логическое произведение логических сумм.

3. Логические тождества

При преобразованиях логических выражений используются следующие логические тождества:

$$\begin{aligned} \bar{\bar{x}} &= x; & x \vee 1 &= 1; & x \vee 0 &= x; & x \cdot 1 &= x; & x \cdot 0 &= 0; \\ x \vee x &= x; & x \cdot x &= x; & x \vee x \cdot y &= x; \\ xy \vee x\bar{y} &= x; & (x \vee y)(x \vee \bar{y}) &= x; & x \vee \bar{x}y &= x \vee y; \\ \overline{xy} &= \bar{x} \vee \bar{y}; & \overline{\bar{x} \vee \bar{y}} &= \overline{\bar{x}} \cdot \overline{\bar{y}}. \end{aligned} \quad (2)$$

4. Логические функции

Любое логическое выражение, составленное из n переменных x_n, x_{n-1}, \dots, x_1 , с помощью конечного числа операций алгебры логики, можно рассматривать как некоторую функцию n переменных. Такую функцию называют логической. В соответствии с аксиомами алгебры логики такая функция может принимать (в зависимости от значения логических переменных, входящих в нее)

значение «0» или «1». Функция n логических переменных может быть определена для 2^n значений переменных, соответствующих всем возможным значениям n -разрядных двоичных чисел. Основным интерес представляют следующие базовые логические функции двух переменных x и y :

$$f1(x, y) = x \cdot y \text{ – логическое умножение (конъюнкция),} \quad (3)$$

$$f2(x, y) = x \vee y \text{ – логическое сложение (дизъюнкция),} \quad (4)$$

$$f3(x, y) = \overline{x \cdot y} \text{ – логическое умножение с инверсией,} \quad (5)$$

$$f4(x, y) = \overline{x \vee y} \text{ – логическое сложение с инверсией,} \quad (6)$$

$$f5(x, y) = x \oplus y = xy \vee \bar{x}\bar{y} \text{ – суммирование по модулю 2,} \quad (7)$$

$$f6(x, y) = \overline{x \oplus y} = xy \vee \bar{x}\bar{y} \text{ – равнозначность.} \quad (8)$$

5. Логические схемы

Физическое устройство, реализующее одну из операций алгебры логики или простейшую логическую функцию, называется логическим элементом. Схема, составленная из конечного числа логических элементов по определенным правилам, называется логической схемой. Логическая схема, наряду с логическим выражением, является одним из способов задания цифровых элементов.

Условное обозначение основных логических элементов, принятых в нашей стране и за рубежом, приведено на рис. 1.

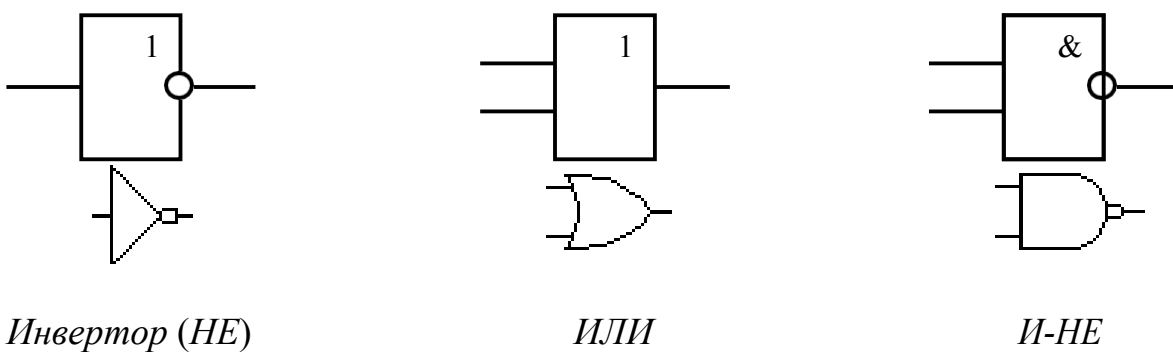


Рис. 1. Обозначение логических элементов.

Примечание. Количество входов в приведенных элементах может быть произвольным; на операцию инвертирования указывает кружок на выходном сигнале.

На рис. 1 в первом ряду приведено обозначение, принятое в нашей стране, во втором ряду, обозначение, принятое за рубежом.

6. Таблица истинности

Поскольку область определения любой логической функции n переменных конечна (2^n значений), то такая функция может быть задана таблицей значений $f(v_i)$, которые она принимает в точках v_i , где $i = 0, 1, \dots, 2^n - 1$. Такие таблицы называют таблицами истинности. В табл. 1 представлены таблицы истинности, задающие указанные выше функции (3) – (8).

Таблица 1.
Произвольная таблица истинности

№	Значения переменных		Функции					
	x	y	f1	f2	f3	f4	f5	f6
1	0	0	0	0	1	1	0	1
2	0	1	0	1	1	0	1	0
3	1	0	0	1	1	0	1	0
4	1	1	1	1	0	0	0	1

В общем случае таблица истинности должна содержать все возможные комбинации логических переменных, входящих в логическое выражение (в случае табл. 1 это комбинации логических переменных x и y), и значения логического выражения, соответствующие каждой комбинации логических переменных

Таблица истинности является третьим способом задания цифровых элементов. Необходимо отметить, что все три способа задания (с помощью логических выражений, логических схем и таблиц истинности) являются однозначными и в равной мере взаимозаменяемыми. Так, по логическому выражению можно составить схему и записать таблицу истинности и наоборот.

7. Получение логических выражений

Целью преобразования сложных логических выражений (а также получения по экспериментальной таблице истинности логического выражения, описывающего данную логическую зависимость) является компактная форма записи, которая в полной мере описывает данную логическую зависимость от любого числа переменных. Операции с логическими выражениями намного удобней, чем с таблицами истинности и со схемами.

Самый простой способ получения логических выражений – это анализ таблицы истинности. Для наглядности рассмотрим пример: пусть требуется найти логическое выражение для функции f_m трех переменных X, Y, Z , описываемых табл. 2.

Таблица 2.
Таблица истинности функции f_m

№	X	Y	Z	f_m
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

Из правил записи логических выражений следует, что искомую зависимость можно получить в двух видах (конъюнктивной или дизъюнктивной нормальных формах – пункт 2). Чаще всего на практике применяется дизъюнктивно нормальная форма записи логической функции.

Дизъюнктивно нормальная форма записи представляет собой дизъюнкцию (логическое сложение) элементарных конъюнкций (произведение всех логических переменных, в нашем случае – это три переменные X, Y, Z).

Для записи логического выражения в дизъюнктивно нормальной форме выберем из таблицы истинности 2 строки, в которых функция f_m принимает значение «1». Это строки 4, 6, 7, 8. Таким образом, элементов нашей дизъюнк-

ции будет 4 (всего слагаемых). Каждое слагаемое – это элементарная конъюнкция. В каждую же элементарную конъюнкцию (произведение), как уже было сказано, должны войти все логические переменные данной строки (в нашем случае X, Y, Z). При этом если соответствующий элемент (X, Y или Z) имеет для данной строки значение «1», то он входит в элементарную конъюнкцию (произведение) в обычном виде, если же он равен «0», то необходимо взять его отрицание. Так, для каждой строки имеем элементарную конъюнкцию (произведение) в следующем виде:

$$\bar{X} \cdot Y \cdot Z \quad \text{для строки 4;}$$

$$X \cdot \bar{Y} \cdot Z \quad \text{для строки 6;}$$

$$X \cdot Y \cdot \bar{Z} \quad \text{для строки 7;}$$

$$X \cdot Y \cdot Z \quad \text{для строки 8.}$$

Искомой формой записи данного логического выражения в дизъюнктивно нормальной форме является дизъюнкция (сложение) полученных выше 4 конъюнкций:

$$f_m = \bar{X}YZ \vee X\bar{Y}Z \vee XY\bar{Z} \vee XYZ. \quad (9)$$

По данной логической функции можно записать таблицу истинности, идентичную табл. 2. Для этого необходимо в логическую функцию подставлять все возможные комбинации логических переменных X, Y, Z (всего таких значений 2^3) и на каждой подстановке, пользуясь аксиомами алгебры логики, вычислять функцию f_m .

Необходимо отметить, что данный способ получения логических выражений не самый рациональный. Полученная логическая функция, хоть и содержит исчерпывающую информацию о выбранной комбинации логических переменных, тем не менее поддается еще упрощению с помощью выражений (2).

Другим способом составления упрощенного выражения по таблице истинности является составление карт Карно. Такой способ в данной работе не рассматривается по двум причинам:

во-первых, наглядность его минимальна при достаточно большой сложности построения самих карт Карно, что вызывает большие трудности у студентов, особенно в начале изучения цифровой техники;

во-вторых, методика построения карт Карно хорошо описана в соответствующей литературе.

Порядок работы

Задание 1. Исследование логической функции «И»

Задание уровней логических сигналов.

Создайте схему, изображенную на рис. 2.

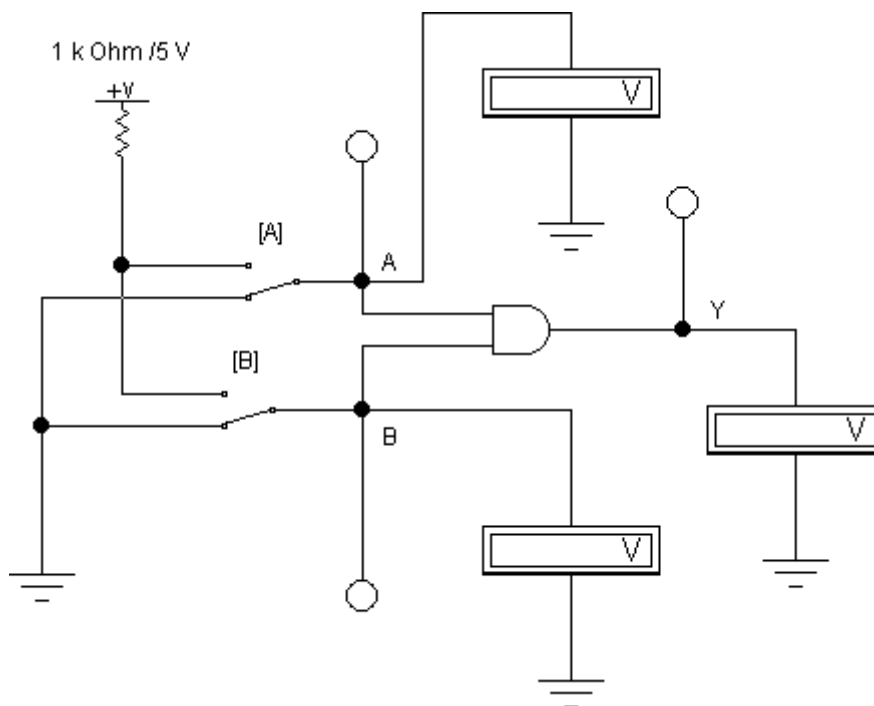


Рис. 2. Логическая функция «И».

В этой схеме два двухпозиционных переключателя «А» и «В» подают на входы логической схемы «И» уровни логических сигналов: «0» (контакт переключателя в нижнем положении) или «1» (контакт переключателя в верхнем положении). В физическом плане это соответствует подаче напряжения от источника (+ 5 В) на логический элемент. Уровень физического сигнала на входе или выходе логического элемента можно измерить с помощью вольтметра (в

данном случае «0» В или «+5» В), а уровень логического сигнала – с помощью логических пробников, которые информируют о наличии на измеряемой линии сигнала вообще (если сигнал есть, то пробник загорается).

Включите схему (тумблером, находящимся в правом верхнем углу окна программы). Установите положения ключей в соответствии с табл. 3. Результаты замеров (логических и физических сигналов) занесите в табл. 3.

Таблица 3.
Задание уровней логических сигналов

Положение переключателей		Сигналы на входах и выходе					
«А»	«В»	Логические (0 или 1)			Физические, В		
		А	В	У	А	В	У
Нижнее	Нижнее						
Нижнее	Верхнее						
Верхнее	Нижнее						
Верхнее	Верхнее						

Примечание. Прежде чем начать работу с переключателями удобно каждому переключателю присвоить букву, при нажатии которой он включается/выключается. Это можно сделать до начала работы схемы, дважды щелкнув на переключатель и на закладке «Value» присвоив ему уникальную кнопку.

Экспериментальное получение таблицы истинности элемента «И».

Подайте на входы схемы (рис. 2) все возможные комбинации уровней сигналов «А» и «В» и для каждой комбинации зафиксируйте уровень выходного сигнала «У». Заполните таблицу истинности исследуемой логической схемы «И» (табл. 4).

Таблица 4.
Таблица истинности логического элемента «И»

Входы		Выход
А	В	У

Получение аналитического выражения для функции.

По табл. 4 составьте аналитическое выражение функции элемента «И» и занесите его себе в отчет. Для этого можно воспользоваться одним из двух способов получения логических выражений по таблице истинности, изложенных в кратких теоретических данных лабораторной работы.

Задание 2. Исследование логической функции «И-НЕ»

Экспериментальное получение таблицы истинности логического элемента «И-НЕ», составленного из элементов «И» и «НЕ».

Соберите схему, изображенную на рис. 3.

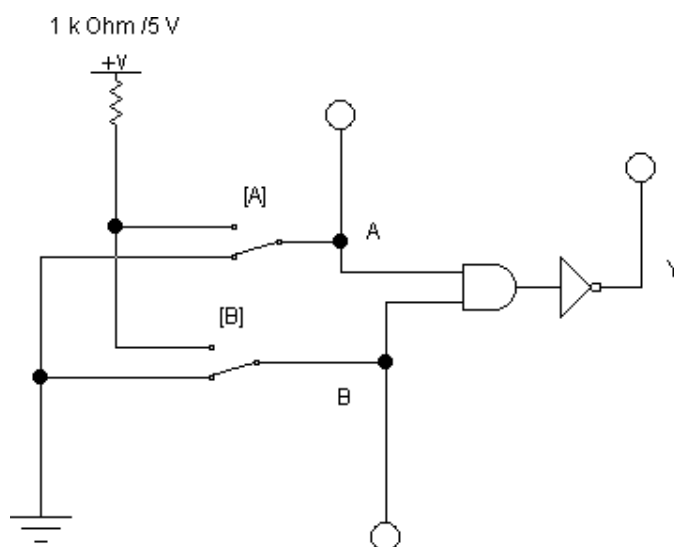


Рис. 3. Логическая функция «И-НЕ».

Включите схему. Подайте на входы схемы все возможные комбинации уровней входных сигналов и, наблюдая уровни сигналов на входах и выходе с помощью логических пробников, заполните таблицу истинности логической схемы «И-НЕ» (табл. 5).

Таблица 5.
Таблица истинности элемента «И-НЕ» (составного)

Входы		Выход
A	B	Y
0	0	
0	1	
1	0	
1	1	

Экспериментальное получение таблицы истинности логического элемента «И-НЕ».

Соберите схему, изображенную на рис. 4.

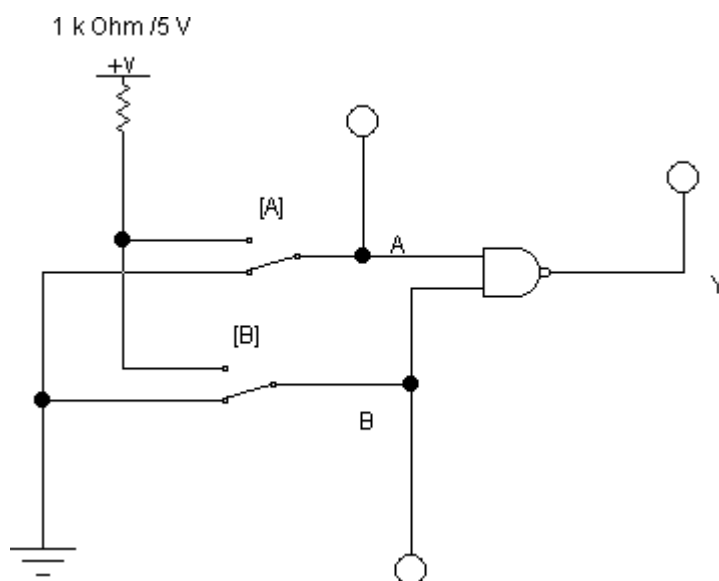


Рис. 4. Логическая функция «И-НЕ».

Включите схему. Подайте на входы схемы все возможные комбинации уровней входных сигналов и, наблюдая уровни сигналов на входах и выходе с помощью логических пробников, заполните таблицу истинности логической схемы «И-НЕ» (табл. 6).

Таблица 6.
Таблица истинности элемента «И-НЕ»

Входы		Выход
A	B	Y
0	0	
0	1	
1	0	
1	1	

Сравните между собой табл. 5 и 6 и сделайте соответствующие выводы.

Задание 3. Исследование логической функции «ИЛИ»

Экспериментальное получение таблицы истинности логического элемента «ИЛИ».

Включите схему. Подайте на входы схемы все возможные комбинации уровней входных сигналов и, наблюдая уровни сигналов на входах и выходе с

помощью логических пробников, заполните таблицу истинности логической схемы «ИЛИ» (табл. 7).

Таблица 7.
Таблица истинности элемента «ИЛИ»

Входы		Выход
A	B	Y
0	0	
0	1	
1	0	
1	1	

Получение аналитического выражения для функции.

По табл. 7 составьте аналитическое выражение функции «ИЛИ» и занесите его в отчет. Для этого можно воспользоваться одним из двух способов получения логических выражений по таблице истинности, изложенных в кратких теоретических данных данной лабораторной работы.

Задание 4. Исследование логической функции «ИЛИ-НЕ»

Экспериментальное получение таблицы истинности логического элемента «ИЛИ-НЕ», составленного из элементов «ИЛИ» и «НЕ».

Соберите схему, изображенную на рис. 6.

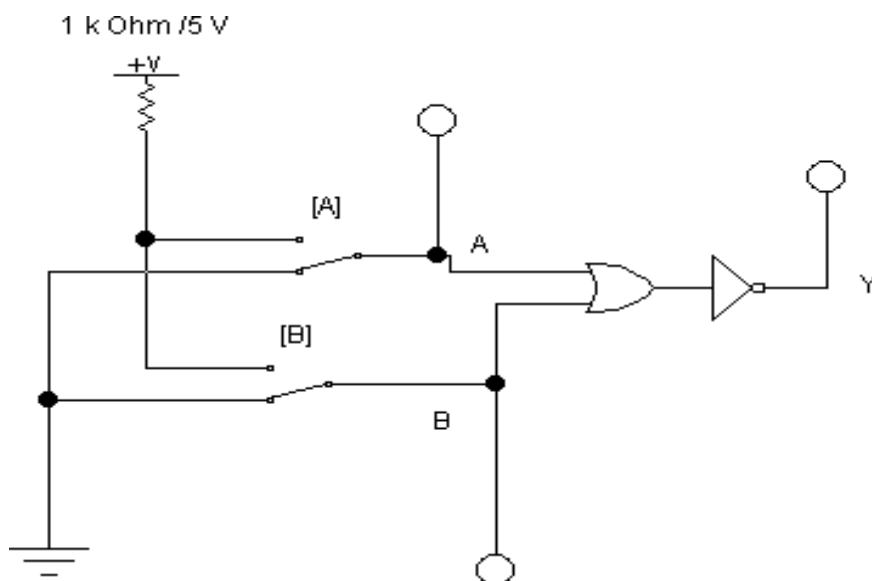


Рис. 6. Логическая функция «ИЛИ-НЕ».

Включите схему. Подайте на входы схемы все возможные комбинации уровней входных сигналов и, наблюдая уровни сигналов на входах и выходе с помощью логических пробников, заполните таблицу истинности логической

схемы «ИЛИ-НЕ» (табл. 8).

Таблица 8.
Таблица истинности элемента «ИЛИ-НЕ» (составного)

Входы		Выход
A	B	Y
0	0	
0	1	
1	0	
1	1	

Экспериментальное получение таблицы истинности логического элемента «ИЛИ-НЕ».

Соберите схему, изображенную на рис. 7.

Включите схему. Подайте на входы схемы все возможные комбинации уровней входных сигналов и, наблюдая уровни сигналов на входах и выходе с помощью логических пробников, заполните таблицу истинности логической схемы «ИЛИ-НЕ» (табл. 9).

Таблица 9.
Таблица истинности элемента «ИЛИ-НЕ»

Входы		Выход
A	B	Y
0	0	
0	1	
1	0	
1	1	

Сравните табл. 8 и 9 между собой и сделайте соответствующие выводы.

Задание 5. Исследование логических схем с помощью генератора слов

Анализ различных логических функций иногда удобно проводить с помощью микросхем, реализующих различные логические элементы. Кроме того, использование микросхем (в отличие от расширенного представления логических функций, рассматриваемого в предыдущих заданиях) бывает необходимо на стадии проектирования сложных систем, когда в руках современного инженера зачастую имеется набор из различных, законченных модулей, реализующих вполне определенные функции.

Отдавая дань доступности и широкому распространению англоязычных микросхем, а также их аналогов, изготовленных по западным стандартам (в том числе и современные российские разработки), в данной работе мы рассматриваем микросхемы как раз западного стандарта.

Помимо того, для дальнейшего рассмотрения цифровой техники удобно вместо коммутирующих выключателей (задающих уровни логических сигналов) в некоторых случаях использовать «генератор слов» (прибор, предназначенный для выработки последовательности логических сигналов на своих выходах, причем данную последовательность пользователь имеет возможность задавать самостоятельно). Использованию модульных микросхем и «генератора слов» посвящено следующее задание.

Исследование модульных микросхем различных типов.

Выбор исследуемой микросхемы производится по номеру варианта, присвоенному студенту преподавателем. Типы микросхем в соответствии с вариантом задания приведены в табл. 10.

Таблица 10.
Варианты задания исследуемой микросхемы

Вариант задания	Тип микросхемы
1	AND/7408
2	AND/7421
3	AND/7411
4	NOR/7402
5	NOR/7427
6	NOR/7428
7	NOR/7433
8	OR/7432
9	XOR/7486
10	XNOR/74266

Примечание. Данные микросхемы в общем случае могут содержать несколько независимых друг от друга логических элементов (но одного типа: например, 3 логических элемента «И» – «AND»), причем у каждого логическо-

го элемента может быть и несколько входов (1, 2, 3 или 4).

После выбора соответствующей микросхемы нужно собрать схему для ее изучения. В общем случае она должна содержать: источник напряжения, заземление (земля), необходимое количество логических пробников и генератор слов.

Пример собранной схемы для исследования микросхемы NOR/7400 приведен на рис. 8.

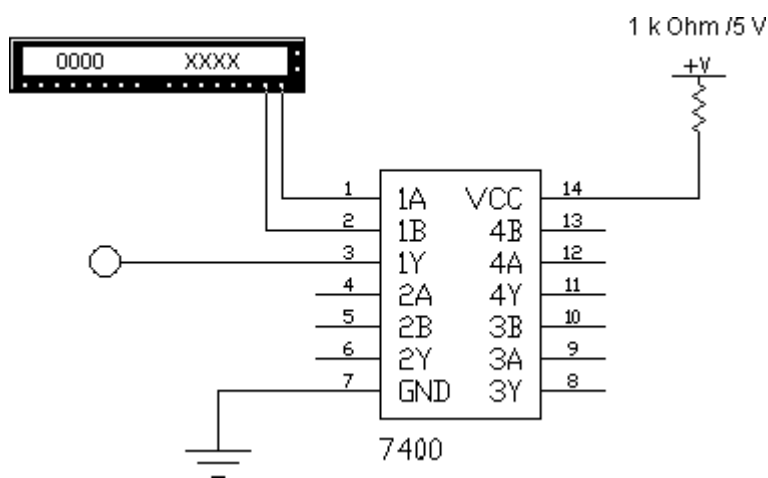


Рис. 8. Исследование микросхемы NOR/7400.

Примечание. Для дальнейшего исследования необходимо принимать во внимание только входы/выходы одного логического элемента на данной микросхеме (на рис. 8 используется первый логический элемент).

В отчете по данному пункту необходимо заполнить табл. 11, дающую исчерпывающую информацию об используемой микросхеме.

Примечание. Исчерпывающую информацию об используемой микросхеме можно получить в справке (выделите интересующий вас элемент и щелкните на знак вопроса).

Экспериментальное получение таблицы истинности микросхемы.

Для дальнейшего изучения микросхемы необходимо использовать «генератор» слов (предполагается, что при выполнении задания 5.2. была собрана схема для изучения, аналогичная рис. 8). Общий вид открытого окна свойств «генератора» показан на рис. 9. Для открытия данного окна необходимо дважды щелкнуть мышью на изображении генератора слов.

Таблица 11.
Информация о микросхеме

Тип микросхемы (полное обозначение по варианту задания)	
Тип базисных элементов (логических функций)	
Число базисных элементов в микросхеме (всего)	
Число исследуемых базисных элементов в микросхеме	
Обозначение выводов микросхемы, используемых для подключения источника питания (номера и название)	
Обозначения выводов микросхемы, используемых для подключения заземления (номера и название)	
Обозначение используемых входов (номера и название)	
Обозначение используемых выводов (номера и название)	

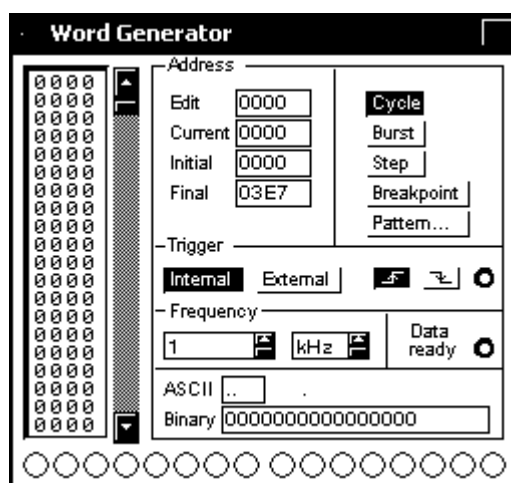


Рис. 9. Генератор слов.

Окно свойств может быть условно разбито на ряд частей.

В левой части, в прокручивающемся окне, отображаются слова (набор логических последовательностей), формируемые «генератором слов» в 16-ричной системе исчисления (данные последовательности могут быть изменены пользователем).

В правой части окна свойств находится ряд областей («Address», «Trigger», «Data ready»), несущих техническую информацию. Из этих областей мы будем пользоваться только кнопками «Cycle» и «Step», переводящие генератор слов в цикловой (заданные пользователем слова прокручиваются автоматиче-

ски) и пошаговый режимы работы (каждая следующая последовательность, вырабатываемая «генератором слов», вызывается нажатием кнопки «Step»).

В правой нижней части окна свойств находятся поля «ASCII», «Binary» и ряд, состоящий из логических пробников. Поле «Binary» (изменяемое) и ряд пробников, (функционирующих только в режиме «Работа»), имеют одинаковую информационную сущность. Они отображают генерируемые последовательности слов в двоичном коде.

Отметим также, что часто бывает удобнее задавать последовательность слов именно в поле «Binary», которое связано (изменение его вызывает изменение другого поля) и с прокручивающимся списком генерируемых слов, находящимся в левой части и рассмотренным ранее.

Для выполнения работы запрограммируйте «генератор слов» так, чтобы на его выходе получил последовательно следующие комбинации: 00, 01, 10, 11 (необходимо помнить: несмотря на то, что в анализируемых микросхемах количество базисных логических элементов и входов на каждом элементе множество, мы исследуем только два входа одного базисного логического элемента). Затем переведите генератор в режим пошаговой работы нажатием кнопки «Step» в окне свойств генератора. Каждое нажатие кнопки «Step» вызывает переход к очередному слову заданной последовательности, которое подается на выход генератора. Последовательно подавая на микросхему слова из заданной последовательности, заполните таблицу истинности элемента вашего базисного элемента (табл. 12).

Таблица 12.
Таблица истинности исследуемой микросхемы

Входы		Выход
первый	второй	Y
0	0	
0	1	
1	0	
1	1	

Указание. Для повторения запрограммированной комбинации слов можно выключить и включить схему заново, тогда последовательности, заданные в «генераторе слов», будут изменяться начиная с первой.

Задание 6. Реализация логической функции 3-х и большего числа переменных

В заключение изучения базовых логических схем и функций приведем задание, иллюстрирующее возможности «Electronic Workbench» в такой непростой и трудно формализуемой задаче как синтез логических схем и функций. Все предыдущие задания опирались в основном на моделирование уже существующих логических функций и изучение их свойств. Выполнив же это задание, студенты смогут автоматически реализовывать любую логическую функцию, заданную булевым выражением в различных базисах, получать таблицы истинности логических выражений и т. п. Все перечисленные процедуры можно реализовать с помощью инструмента «Logic Converter». Общий вид окна свойств «Logic Converter» показан на рис. 10.

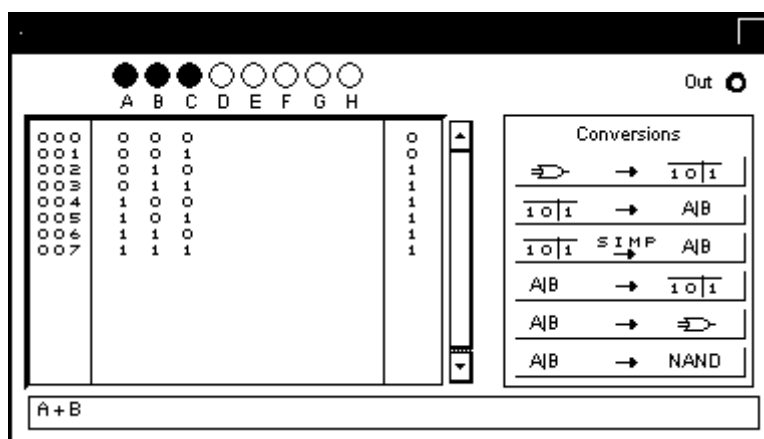


Рис. 10. «Logic Converter».

Напомним, что окно свойств вызывается двойным нажатием на изображении инструмента. Рассмотрим окно свойств подробнее.

Здесь в верхней части под буквами А, В, ..., Н находятся индикаторы, активизируемые путем нажатия на них. На рис. 10 активизированы три индикатора – А, В, С. Подразумевается, что в дальнейшем мы будем оперировать ло-

гической функцией, состоящей из трех переменных – A , B , C . Данный выбор вызывает появление всех возможных комбинаций этих переменных (строки таблицы, изображенной на рис. 10, начиная с нулевой и заканчивая седьмой). В правой части полученной таблицы истинности, в отдельном столбце, по умолчанию присвоены «0» выходной переменной. Пользователь, изменяя значения выхода анализируемой логической функции, тем самым полностью задает таблицу истинности.

В самом нижнем окне находится поле ввода/вывода логической функции (поле является выводящим, если пользователь сам задал таблицу истинности, а затем путем нажатия на соответствующую функциональную кнопку потребовал формирования логической функции; и поле является вводящим, если пользователь предварительно сам задал логическую функцию, а затем путем нажатия на соответствующую функциональную кнопку потребовал формирования таблицы истинности).

В правой части окна свойств находятся соответствующие функциональные кнопки (панель «Conversion»).

1) – получение таблицы истинности для произвольной логической схемы, подсоединенной к входам и выходам «Logic Converter»;

2) $\overline{101} \rightarrow A|B$ – получение по таблице истинности логической функции (не упрощенной);

3) – получение по таблице истинности упрощенной логической функции (аналогичная функция получается путем использования для анализа таблицы истинности карт Карно);

4) – получение по логической функции таблицы истинности;

5) $A|B \rightarrow \text{схема}$ – получение по логической функции схемы, реализованной в произвольном базисе;

6) $A|B \rightarrow \text{NAND}$ – получение по таблице истинности логической функции реализованной в базисе «И-НЕ».

Синтез схемы, реализующей заданную функцию при помощи логического преобразователя.

Для получения схемы, реализующей функцию, описываемую логическим выражением f (задается преподавателем), можно воспользоваться логическим преобразователем. Варианты заданий логических выражений приведены в табл. 13 (здесь апостроф указывает на операцию инвертирования).

Таблица 13.
Варианты задания логического выражения f

Вариант задания	Логическое выражение f
1	$A+B*C$
2	$C+A*B$
3	$A*B*C$
4	$A'+B+C$
5	$A+B'C$
6	$C*B'+A$
7	$A'+B'+C'$
8	$C'+A'*B$
9	$A'+B'+C$
10	$B+C*A'$

Прodelайте с анализируемым логическим выражением следующие действия:

- 1) вызовите логический преобразователь;
- 2) введите в нижнее окно панели преобразователя логическое выражение с клавиатуры (операции ИЛИ соответствует знак «+», инверсия обозначается апострофом, логическая операция умножения не вводится: например, $AB=A*B$);

3) для реализации схемы на элементах «И-НЕ» нажмите соответствующую функциональную кнопку на панели логического преобразователя, занесите полученную схему в отчет, удалите полученную схему;

4) для простой реализации схемы на произвольных элементах нажмите соответствующую функциональную кнопку на панели логического преобразователя, занесите полученную схему в отчет, удалите полученную схему;

5) для получения таблицы истинности заданной логической функции нажмите соответствующую функциональную кнопку на панели логического преобразователя, занесите полученную таблицу истинности в отчет.

Синтез логической функции, реализующей заданную таблицу истинности при помощи логического преобразователя.

Для получения функции, реализующей таблицу истинности, необходимо воспользоваться таблицей истинности, полученной в предыдущем пункте. Для этого сделайте следующее:

1) удалите с нижней строки логического преобразователя логическую функцию;

2) очистите таблицу истинности преобразователя, нажав на его верхнюю функциональную кнопку;

3) выберите в поле таблицы истинности количество используемых логических переменных (в нашем случае их 3) путем нажатия на соответствующие индикаторы;

4) в правой колонке полученной таблицы истинности задайте значение искомой логической функции в соответствии с каждой строкой;

5) для получения неупрощенной логической функции нажмите соответствующую функциональную кнопку на панели логического преобразователя, занесите в отчет полученную функцию;

6) для получения упрощенной логической функции нажмите соответствующую функциональную кнопку на панели логического преобразователя, запишите упрощенную логическую функцию;

7) удостоверьтесь, что упрощенные модели совпадают с заданными в начале моделями;

8) удостоверьтесь, что упрощенная функция может быть получена из неупрощенной (проделайте операции упрощения).

Контрольные вопросы

1. Дайте определение логического сигнала.
2. Дайте определение логической переменной.
3. Дайте определение логической функции.
4. Какие значения могут принимать булевы переменные?
5. Приведите основные логические тождества.
6. Что может быть принято за уровни логических сигналов?
7. Подумайте, почему в технике за уровень логического нуля часто принимают физический сигнал (например, по току), отличный от нуля?
8. Как может быть получена логическая функция?
9. Чем в физическом смысле отличается работа схемы составленной по упрощенной логической функции, от неупрощенной?
10. Сколько различных комбинаций сигналов надо подать на схему, имеющую 4 входа, для составления таблицы истинности?
11. Какой сигнал должен быть подан на неиспользуемые входы элемента «И-НЕ», имеющего 5 входов, если требуется реализовать ту же логическую функцию, но на 3 входа?
12. Какой сигнал должен быть подан на неиспользуемые входы элемента «ИЛИ», имеющего 5 входов, если требуется реализовать ту же логическую функцию, но на 4 входа?
13. Какой сигнал нужно подать на неиспользуемые входы элемента «И», имеющего 2 входа, для реализации на его базе инвертора на один входной сигнал?
14. Какой логической функцией можно описать систему пуска трехфазного двигателя, если двигатель может быть запущен, когда три дат-

чика подтверждают наличие фазных напряжений?

Упражнения

1. Получение логических функций

По заданной таблице истинности (табл. 14) получите логическую функцию.

Таблица 14.
Варианты таблицы истинности

Значения логических переменных (для всех вариантов)				Варианты задания									
				1	2	3	4	5	6	7	8	9	10
A	B	C	D	Значения логической функции (для каждого варианта)									
0	0	0	0	1	0	1	1	0	1	0	0	1	0
0	0	1	0	0	0	1	1	1	1	0	0	1	1
0	1	0	0	0	0	1	1	0	1	0	0	1	0
0	1	1	0	0	1	0	0	0	0	1	0	1	0
1	0	0	0	1	1	1	1	1	0	0	0	1	0
1	0	1	0	0	1	1	0	1	1	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	1	0	0
1	1	1	0	0	0	1	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0	1	0	1	0	1	1
0	0	1	1	1	0	1	1	1	0	0	0	1	1
0	1	0	1	1	1	0	1	0	1	0	0	0	1
0	1	1	1	0	0	1	1	1	1	1	1	1	1
1	0	0	1	1	0	1	1	0	0	0	1	0	1
1	0	1	1	0	0	1	1	0	0	0	1	0	1
1	1	0	1	1	1	0	1	0	1	1	1	1	1
1	1	1	1	1	1	0	0	1	1	0	1	1	1

Примечание. Значения логических переменных, входящих в вашу логическую функцию, заданы столбцами таблицы (A, B, C, D), они одинаковы для всех вариантов. Из второй части таблицы необходимо выбрать один столбец, согласно вашему варианту, который содержит значения вашей логической функции.

2. Построение структурных схем

По заданной логической функции необходимо построить структурную схему. Варианты логических функций, приведены в таблице 15.

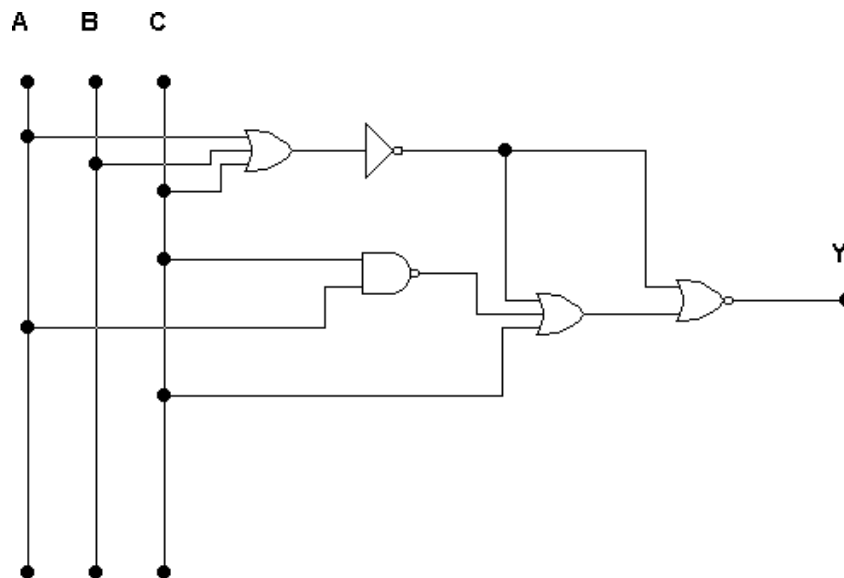
Таблица 15.
Варианты логических функций

Вариант	Логическая функция
1	$f = \overline{a \vee b} \wedge c \wedge d \vee e$
2	$f = \overline{a \wedge c} \vee d \vee e \vee b$
3	$f = c \wedge \overline{a \vee b} \wedge d \vee e$
4	$f = a \vee b \wedge e \wedge d \vee c$
5	$f = a \vee b \wedge c \vee d \vee e$
6	$f = a \wedge c \wedge \overline{d} \vee e \vee \overline{b}$
7	$f = \overline{a \vee b \vee c \vee d} \wedge e$
8	$f = a \wedge b \wedge \overline{d} \vee a \wedge e \vee c$
9	$f = a \wedge a \wedge \overline{c} \wedge d \vee e$
10	$f = e \vee a \wedge \overline{a} \vee d \vee e \wedge b$

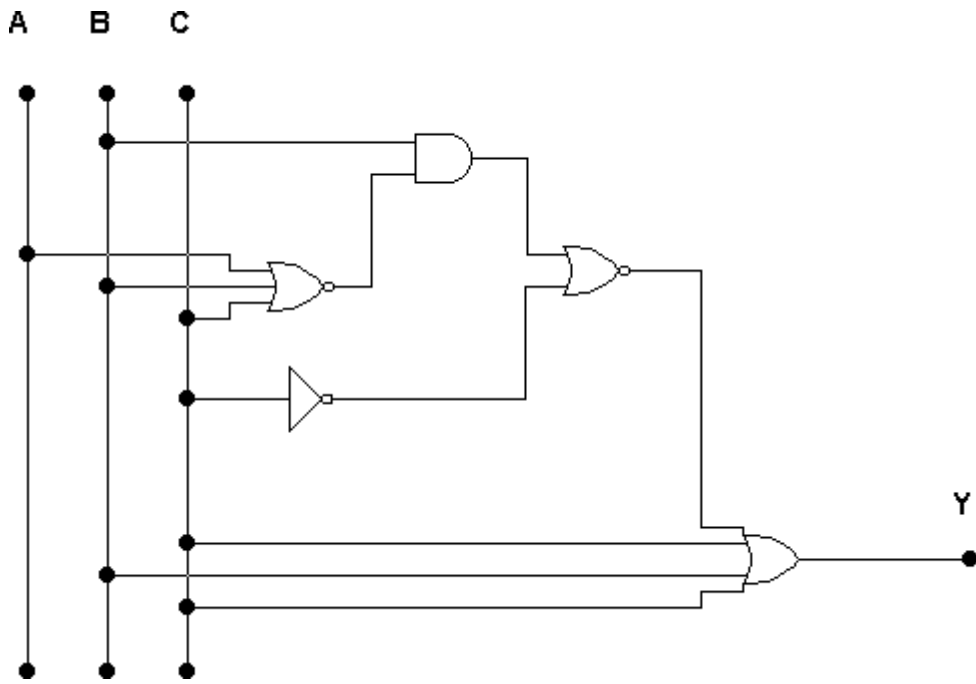
3. Составление логических функций.

По заданной схеме составьте логическую функцию.

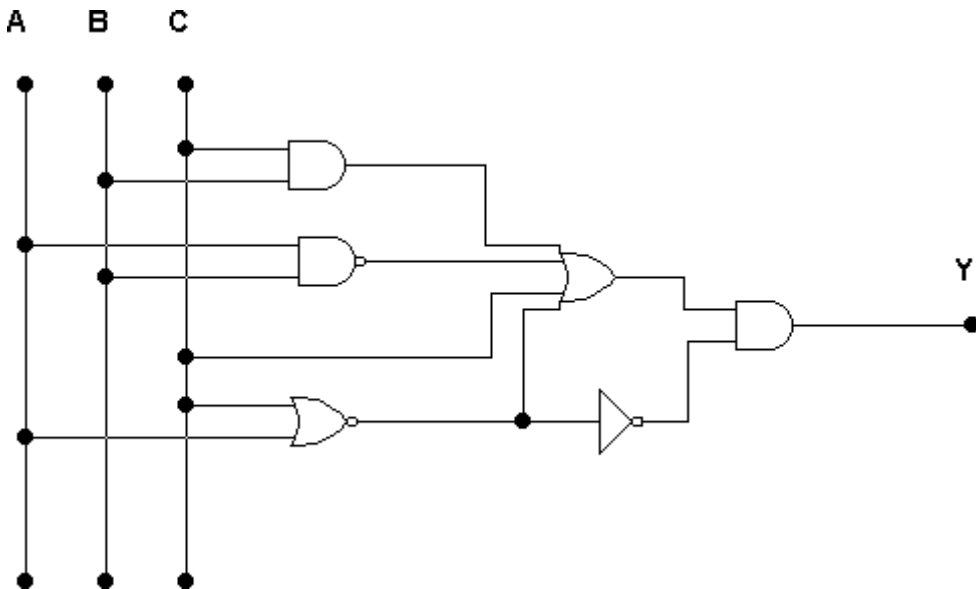
Вариант 1



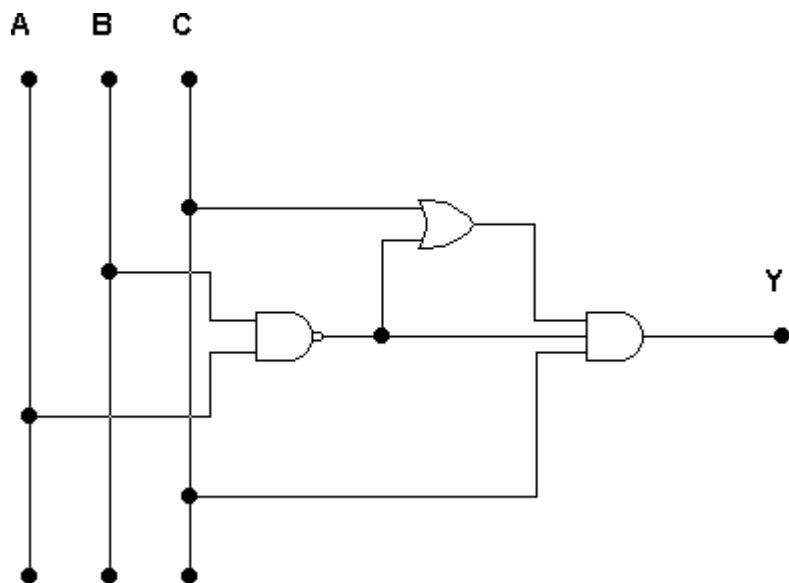
Вариант 2



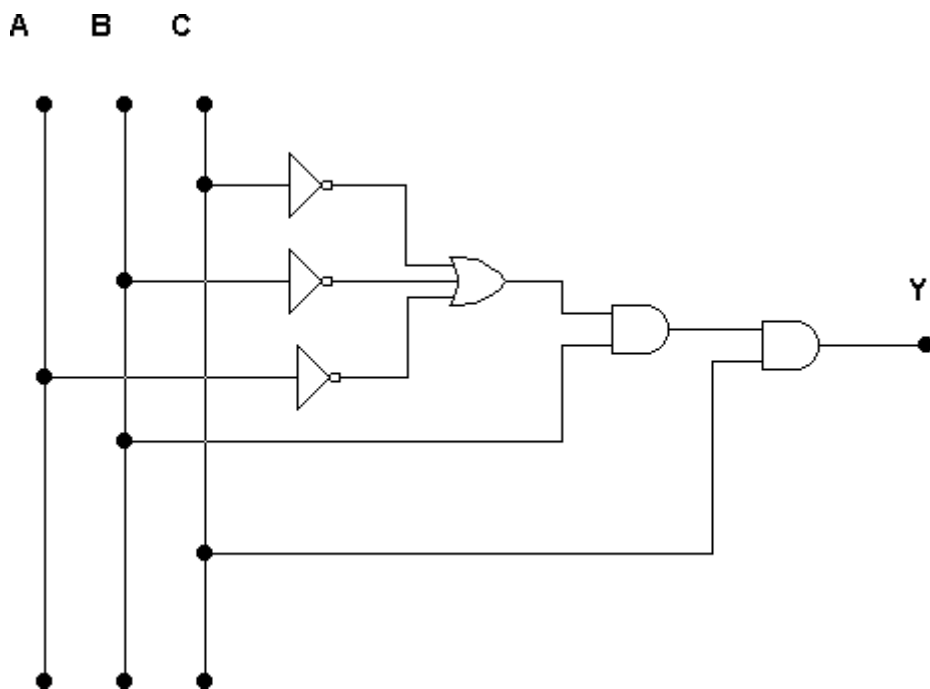
Вариант 3



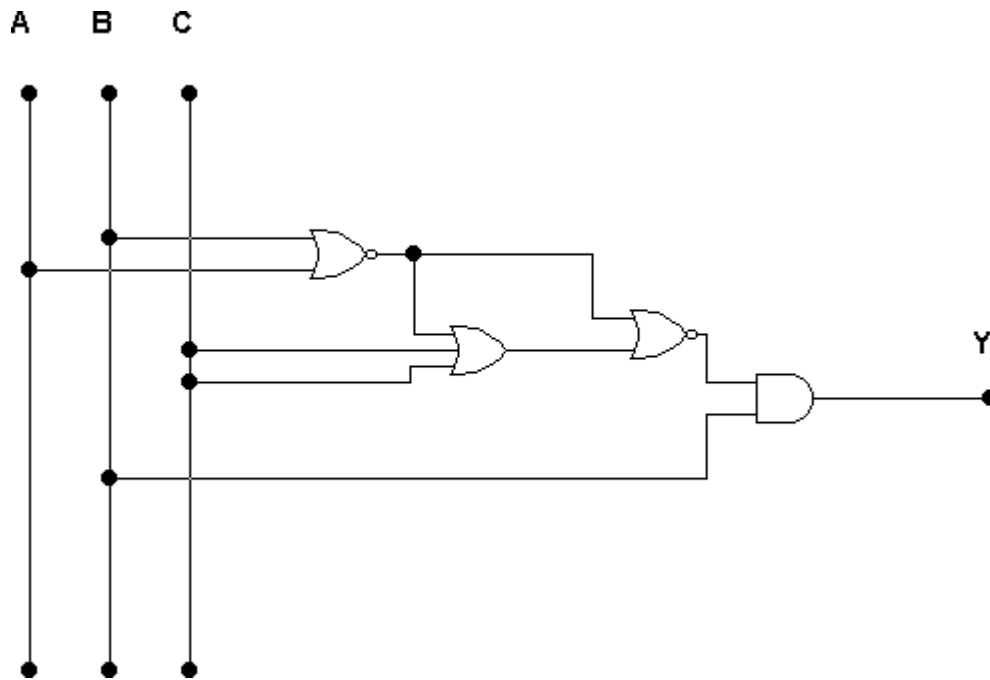
Вариант 4



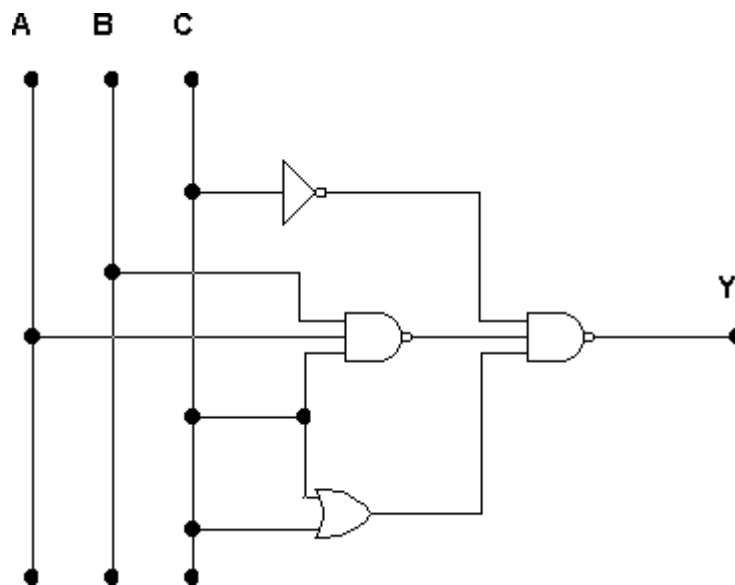
Вариант 5



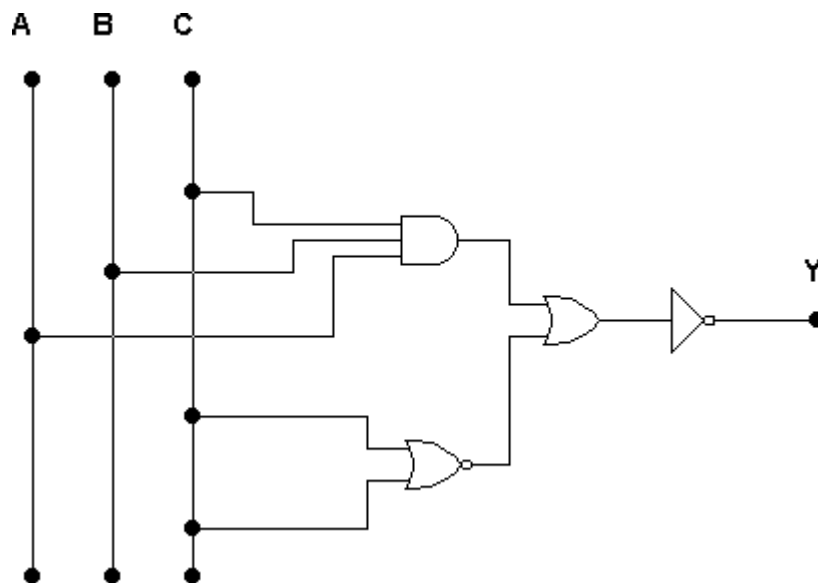
Вариант 6



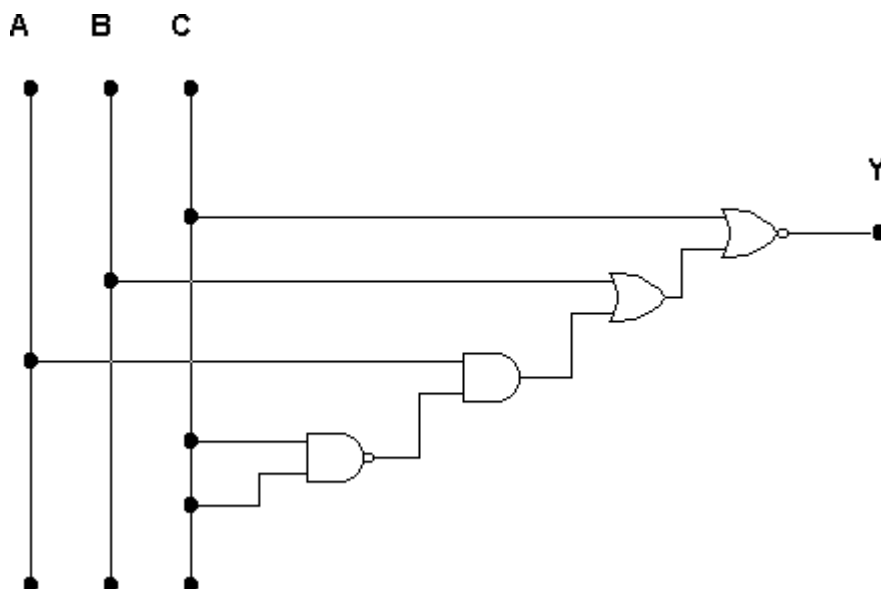
Вариант 7



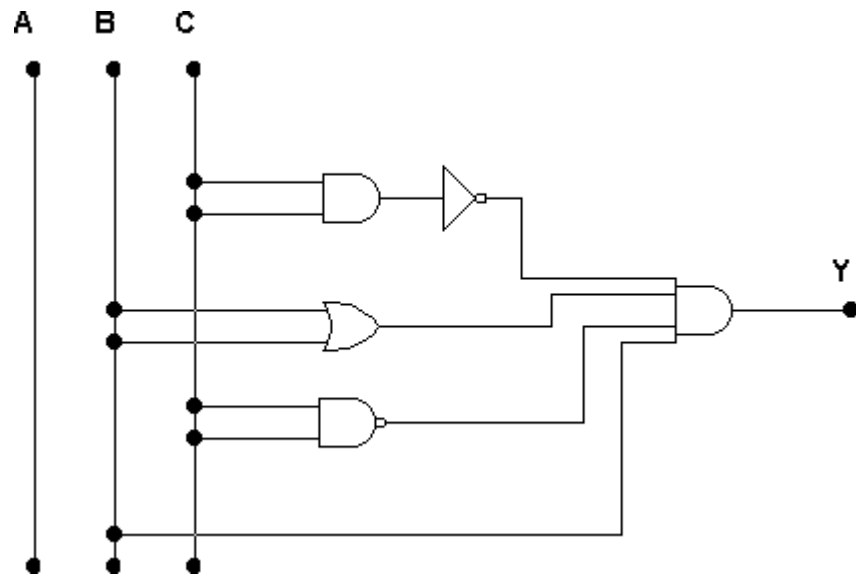
Вариант 8



Вариант 9



Вариант 10



ЛАБОРАТОРНАЯ РАБОТА № 2

ИЗУЧЕНИЕ РАБОТЫ ШИФРАТОРОВ, ДЕШИФРАТОРОВ И МУЛЬТИПЛЕКСОРОВ

Цель работы.

1. Изучение принципов работы шифраторов, дешифраторов и мультиплексоров.
2. Реализация логических функций с помощью мультиплексоров.
3. Изучение способов применения дешифраторов.

Приборы и элементы.

Генератор слов (панель «Instruments/Word Generator»).

Логический анализатор (панель «Instruments/Logic Analyzer»).

Логические пробники (панель «Indicators/Red probe»).

Источник напряжения + 5 В (панель «Basic/Pull-Up Resistor»).

Земля (панель «Sources/Ground»).

Двухвходовые элементы И, И-НЕ, ИЛИ, ИЛИ-НЕ (панель «Logic Gates/
2-Input AND, NAND, OR, NOR Gates»).

Двухпозиционные переключатели (панель «Basic/Switch»).

Дешифратор (панель «Digital/DEC/Generic 8-to-1 DEMUX»).

Мультиплексор (панель «Digital/MUX/ Generic 1-of 8 MUX»).

Краткие теоретические сведения

1. Комбинационные схемы

Комбинационной схемой называется логическая схема, реализующая однозначное соответствие между значениями входных и выходных сигналов. Для реализации комбинационных схем используются логические элементы, выпускаемые в виде интегральных схем. В этот класс входят интегральные схемы дешифраторов, шифраторов, мультиплексоров, демультиплексоров и сумматоров.

2. Шифраторы

Шифратор – логическая комбинационная схема, которая имеет 2^n входов (где n – число информационных выходов). Часто $n=3$, тогда $2^n=8$. Подаче на один из входов активного сигнала будет соответствовать двоичное число, которое можно сформировать из его n выходов, эквивалентное номеру входа, на котором появился активный уровень.

Примечание. Формирование двоичного числа на выходе шифратора означает следующее: каждый из выходов шифратора считается определенным разрядом искомого двоичного числа. Принцип работы шифратора противоположен принципу работы дешифратора, который подробно рассмотрен ниже.

Простейшим случаем применения шифратора может служить, например, схема отслеживания нажатия одной кнопки. Каждая кнопка представляет собой элементарный переключатель. Пусть таких кнопок всего 8 и они пронумерованы начиная с «0» и заканчивая «7». При нажатии определенной кнопки (например, «3») формируется сигнал на входе шифратора (каждый вход – это вполне определенная кнопка), в итоге на выходе шифратора можно получить двоичный сигнал, равный номеру нажатой кнопки. В нашем случае число выходов равно 3 ($2^3=8$) и на каждом из выходов получится следующая комбинация на 3-м = 0, на 2-м = 1, на 1-м = 1. Полученное двоичное число «011», которое в десятичном коде равно 3.

3. Дешифраторы

Дешифратор – логическая комбинационная схема, которая имеет n информационных входов и 2^n выходов. Каждой комбинации логических уровней на входах будет соответствовать активный уровень на одном из 2^n выходов. Обычно n равно 2, 3 или 4. В отечественной литературе входы дешифратора принято обозначать 1, 2, 4, 8, ..., в англоязычной A, B, C, \dots . Работа дешифратора может быть проиллюстрирована в соответствии с таблицей истинности.

Таблица 1.
Таблица истинности дешифратора

Входы				Выходы									
8	4	2	1	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

На рис. 1 представлен дешифратор, имеющий таблицу истинности табл. 1. Условное обозначение базовых элементов соответствует обозначению, принятому в отечественной литературе.

В дальнейшем для детального рассмотрения дешифратора мы будем пользоваться обозначениями (микросхем и базовых элементов), принятыми в EWB. На рис. 2 изображен дешифратор (блок микросхемы, использующийся для моделирования дешифратора в EWB) с $n = 3$.

Активным уровнем сигнала данной микросхемы является уровень логического нуля, т.е. в отличие от таблицы истинности 1 у данного дешифратора на выходах по диагонали расположены нули, а остальные элементы равны единице. На входы C, B, A можно подать следующие комбинации логических уровней: «000», «001», «010», ..., «111», всего 8 комбинаций. Схема имеет 8 выходов, на одном из которых формируется низкий потенциал, на остальных – высокий. Номер этого единственного выхода, на котором формируется активный (нулевой) уровень, соответствует числу N , определяемому состоянием входов C, B, A следующим образом:

$$N = C \cdot 2^2 + B \cdot 2^1 + A \cdot 2^0 \quad (1)$$

Например, если на входы подана комбинация логических уровней «011», то из 8 выходов микросхемы на выходе с номером $N=3$ (в двоичном

исчислении $3 = 011$) установится нулевой уровень сигнала ($Y_3=0$), а все остальные выходы будут иметь уровень логической единицы. Этот принцип формирования выходного сигнала можно описать следующим образом:

$$Y_i = \begin{cases} 0, & \text{если } i = k, \\ 1, & \text{иначе} \end{cases} \quad (2)$$

$$i \begin{cases} 1, \text{ если } i \\ k = 2^2 \cdot C + 2^1 \cdot B + 2^0 \cdot A. \end{cases}$$

Помимо информационных входов A, B, C , дешифраторы обычно имеют дополнительные входы управления G . Сигналы на этих входах, например, разрешают функционирование дешифратора или переводят его в пассивное состояние, при котором (независимо от сигналов на информационных входах) на всех выходах установится уровень логической единицы. Можно сказать, что существует некоторая функция разрешения, значение которой определяется состояниями управляющих входов.

Разрешающий вход дешифратора может быть прямым или инверсным. У дешифраторов с прямым разрешающим входом активным уровнем является уровень логической единицы, у дешифраторов с инверсным входом – уровень логического нуля. На рис. 2 представлен дешифратор с одним инверсным входом управления. Принцип формирования выходного сигнала в этом дешифраторе с учетом сигнала управления описывается следующим образом:

$$Y_i = \begin{cases} \overline{G}, & \text{если } i = k, \\ 1, & \text{если } i \neq k, \\ k = 2^2 \cdot C + 2^1 \cdot B + 2^0 \cdot A. \end{cases} \quad (3)$$

У дешифратора с несколькими входами управления функция разрешения, как правило, представляет собой логическое произведение всех разрешающих сигналов управления. Например, для дешифратора серии 74138 с одним прямым входом управления $G1$ и двумя инверсными $G2A$ и $G2B$ (рис. 3) функции выхода Y_i и разрешения G имеют вид:

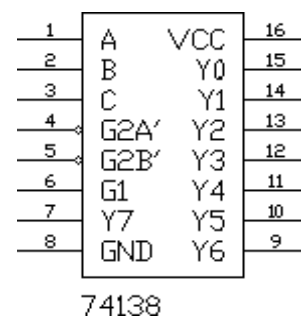


Рис. 3. Дешифратор серии 74138.

$$Y_i = \begin{cases} \overline{G}, & \text{если } i = k, \\ 1, & \text{если } i \neq k, \\ k = 2^2 \cdot C + 2^1 \cdot B + 2^0 \cdot A. \end{cases} \quad (4)$$

$$G = G1 \cdot \overline{G2A} \cdot \overline{G2B} \quad (5)$$

Обычно входы управления используются для каскадирования (увеличения разрядности) дешифраторов или при параллельной работе нескольких схем на общие выходные линии.

Дешифратор может быть использован как *демультиплексор* – логический коммутатор, подключающий входной сигнал к одному из выходов. В этом случае функцию информационного входа выполняет один из входов разрешения, а состояние входов *C*, *B* и *A* задает номер выхода, на который передается сигнал с входа разрешения.

4. Мультиплексоры

Мультиплексор – комбинационная логическая схема, представляющая собой управляемый переключатель, который подключает к выходу один из входов данных. Номер подключаемого входа равен числу (адресу), определяемому комбинацией логических уровней на адресных входах.

Демультиплексорами называются устройства, которые позволяют подключать один вход к нескольким выходам.

В простейшем случае переключения (коммутацию) можно осуществить при помощи ключей, как это показано на рис. 4.

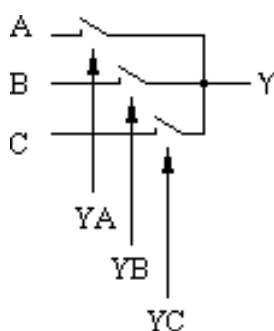


Рис. 4. Мультиплексор на ключах.

В цифровых схемах управление ключами осуществляется при помощи логических сигналов. Сами ключи при этом заменяются соответствующими логическим элементами.

Рассмотрим пример простейшей схемы мультиплексора. Для этого воспользуемся базовым логическим элементом «И» с таблицей истинности 2.

Теперь один из входов элемента будем рассматривать как информацион-

ный вход электронного ключа, а другой – как адресный. По таблице истинности отчетливо видно, что пока на адресный вход Y подан логический уровень «0» сигнал с входа X на выход Out не проходит.

Таблица 2.
Таблица истинности элемента «И»

Входы		Выход
X	Y	Out
0	0	0
0	1	0
1	0	0
1	1	1

При подаче на адресный вход Y логической «1» сигнал, поступающий на вход X , поступает на выход Out , т.е. логический элемент «И» можно использовать в качестве электронного ключа. При этом неважно, какой из входов элемента будет использоваться в качестве адресного входа, а какой – в качестве информационного. Остается только объединить выходы элементов «И» на один выход. Это делается при помощи элемента «ИЛИ». Условное обозначение такой схемы приведено на рис. 5.

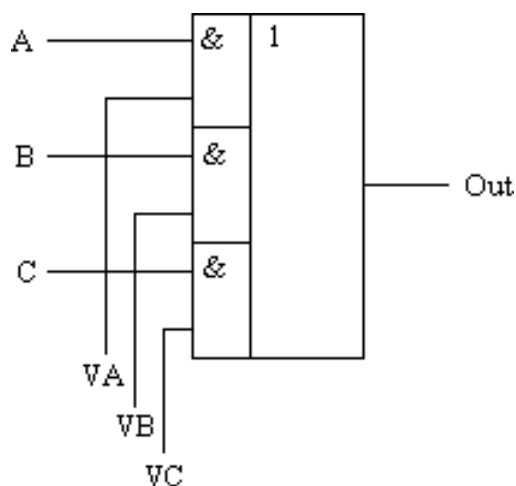


Рис. 5. Принципиальная схема мультиплексора.

Чаще всего для управления требуется много входов, поэтому в схему мультиплексора включают дешифратор. Это позволяет управлять переключением входов микросхемы на выход при помощи двоичных кодов.

Для дальнейшего рассмотрения мультиплексоров мы будем как и прежде пользоваться условными обозначениями логических элементов и микросхем, принятых в EWB.

Кроме информационных и адресных входов, схемы мультиплексоров содержат вход разрешения, при подаче на который активного уровня мультиплексор переходит в рабочее состояние. При подаче на вход разрешения пассивного уровня мультиплексор перейдет в нерабочее состояние, в котором сигнал на выходе сохраняет постоянное значение независимо от значений информационных и управляющих сигналов. Число информационных входов у мультиплексоров обычно равно 2, 4, 8 или 16.

На рис. 7 представлен мультиплексор (блок микросхемы, использующийся для моделирования дешифратора в EWB) 8x1 с инверсным входом разрешения G , прямым Y и инверсным W -выходами ($W = \bar{Y}$)

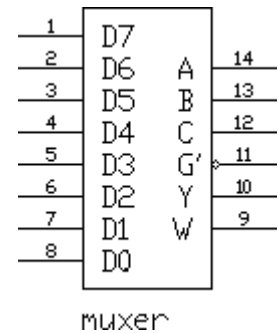


Рис. 7.

Мультиплексор.

5. Реализация логических функций

Функционирование мультиплексора, представленного на рис. 7, описывается характеристическим уравнением, связывающим сигнал на выходе (Y) с разрешающим (G), входными информационными ($D0...D7$) и адресными (A, B, C) сигналами:

$$Y = \left(C \cdot B \cdot A \cdot D0 + C \cdot B \cdot A \cdot D1 + C \cdot B \cdot A \cdot D2 + C \cdot B \cdot A \cdot D3 \right) \cdot G \quad (6)$$

$$\left(+C \cdot B \cdot A \cdot D4 + C \cdot B \cdot A \cdot D5 + C \cdot B \cdot A \cdot D6 + C \cdot B \cdot A \cdot D7 \right)$$

Как видно из уравнения, на мультиплексоре можно реализовать логические функции, для чего нужно определить, какие сигналы и логические константы следует подавать на входы мультиплексора.

Логическая функция n переменных определена для 2^n комбинаций значений переменных. Это позволяет реализовать функцию n переменных на мультиплексоре, имеющем n адресных и 2^n информационных входов. В этом случае каждой комбинации значений аргументов соответствует единственный информационный вход мультиплексора, на который подается значение функции.

Например, требуется реализовать функцию:

$$F_1 = \bar{c} \cdot \bar{b} \cdot \bar{a} \vee c \cdot b \cdot a \vee c \cdot b \cdot \bar{a} \vee \bar{c} \cdot b \cdot a. \quad (7)$$

Эта функция определена только для 8 комбинаций значений переменных, поэтому для ее реализации можно использовать мультиплексор 8x1 с тремя адресными входами. Составим таблицу истинности функции (табл. 3).

Таблица 3.
Таблица истинности логической функции F_1

Вход мультиплексора	Логические переменные			Выход функции
	c	b	a	
N				F_1
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Из табл. 3 видно, что для реализации функции на мультиплексоре необходимо подать на информационный вход мультиплексора с номером N сигнал, значение которого равно соответствующему значению функции F_1 , т.е. на входы с номерами 1, 2, 4, 5 следует подать уровень логического нуля, а на остальные – уровень логической единицы. Таким образом, при подаче комби-

нации логических уровней (a , b , c) на адресные входы мультиплексора к его выходу подключится вход, значение сигнала на котором равно соответствующему значению функции. Схемная реализация приведена на рис. 8.

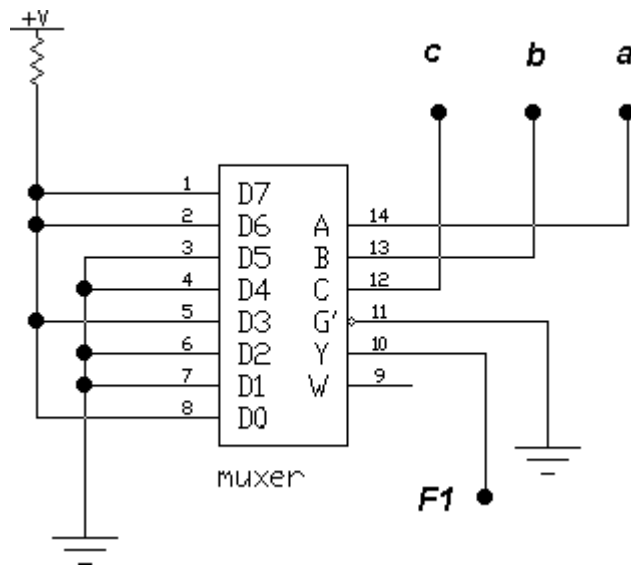


Рис. 8. Реализация логической функции.

При реализации логических функций на информационные входы можно подавать не только константы, но и изменяющиеся входные сигналы. Рассмотрим, например, другой способ реализации функции F_1 , определенной выражением (7). Для этого минимизируем выражение функции с помощью известных логических тождеств (см. лабораторную работу № 1) до вида:

$$F_1 = \bar{c} \cdot \bar{b} \cdot \bar{a} \vee b \cdot c \vee b \cdot a. \quad (8)$$

Составим таблицу истинности функции (8) в зависимости от значений переменных a и b (см. табл. 4).

Для составления таблицы в выражение (8) подставляли комбинации a и b и, пользуясь логическими тождествами (см. лабораторную работу № 1), получали значение функции F_1 . Заданную такой таблицей функцию реализуют, как и в предыдущем случае, подав на вход с номером N сигнал, значение которого соответствует значению функции F_1 .

Таблица 4.
Таблица истинности упрощенной логической функции F_1

Вход мультиплексора	Логические переменные		Выход функции
	b	a	
N			F_1

0	0	0	c
1	0	1	0
2	1	0	c
3	1	1	1

В данном случае сигналы c и \bar{c} , соответствующие переменной c , подаются на информационные входы, как указано в таблице истинности. При этом сокращается число управляющих входов.

Схемная реализация такого способа задания функции приведена на рис. 9.

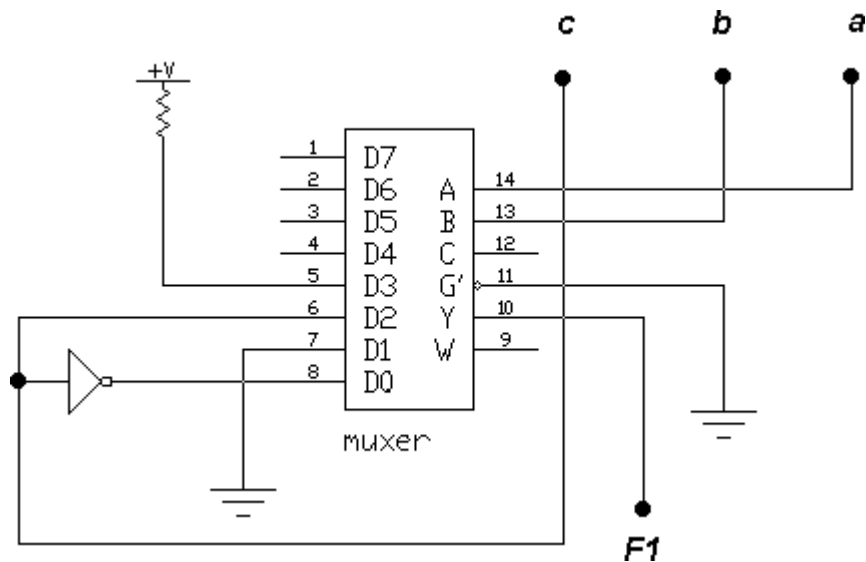


Рис. 9. Реализация упрощенной логической функции.

Так как используются только два адресных входа, вход C остается неподключенным. При этом состояние информационных входов $D4...D7$ безразлично. Схема рис. 9 по существу представляет собой мультиплексор 4×1 с 2 адресными и 4 информационными входами.

Если функцию можно представить в виде произведения одночлена на многочлен, то ее также можно реализовать при помощи мультиплексора. Как следует из уравнения мультиплексора, сигнал, соответствующий одночлену, нужно подать на вход разрешения. Например, требуется реализовать функцию F_2 , описываемую следующим выражением:

$$F_2 = \bar{x} \cdot (d \cdot c \cdot \bar{b} \vee d \cdot \bar{b} \cdot a \vee e \cdot \bar{c} \cdot b \cdot a \vee c \cdot b \cdot a). \quad (9)$$

При реализации данной функции на мультиплексоре сигнал, соответ-

ствующий переменной x , следует подать на его разрешающий вход. Рассмотрим, какие сигналы необходимо подать на управляющие входы мультиплексора. Выражение в скобках можно рассматривать как некоторую функцию f пяти переменных: a, b, c, d, e , из которых наиболее часто используются переменные a, b и c . Поэтому сигналы, соответствующие этим переменным, нужно подать на адресные входы мультиплексора.

Определим, какие сигналы следует подать на информационные входы, чтобы реализовать функцию f . Для этого составим таблицу истинности функции в зависимости от значений переменных a, b и c (табл. 5).

Таблица 5.
Таблица истинности логической функции F_2

Вход мультиплексора	Логические переменные			Выход функции
	c	b	a	
N				f
0	0	0	0	0
1	0	0	1	d
2	0	1	0	0
3	0	1	1	e
4	1	0	0	d
5	1	0	1	d
6	1	1	0	0
7	1	1	1	1

Из табл. 5 видно, что на информационные входы с номерами $N = 0, 2, 6$ нужно подать уровень логического нуля. Сигнал, соответствующий переменной d , нужно подать на входы с номерами $N = 1, 4, 5$, сигнал, соответствующий переменной e , – на вход с номером 3. Схемная реализация такого способа задания функции приведена на рис. 10.

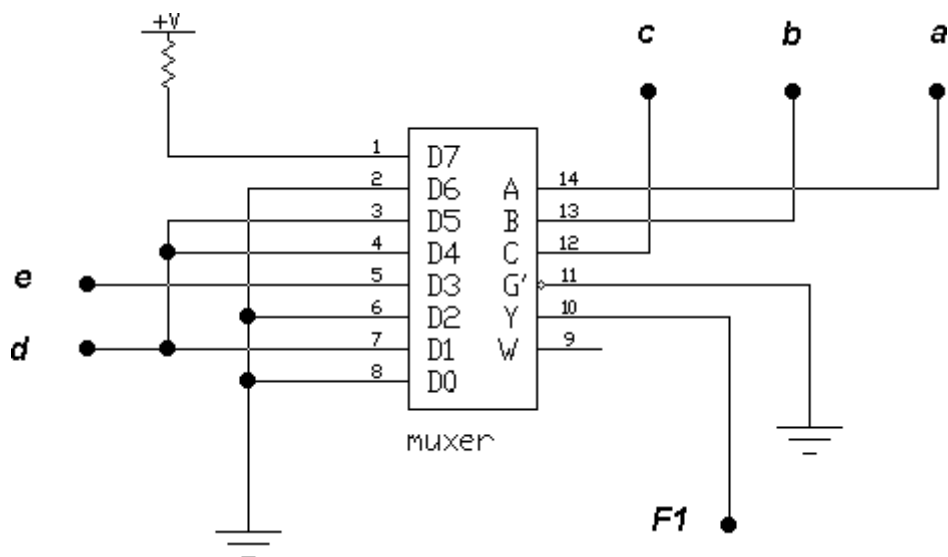


Рис. 10. Реализация логической функции F_2 .

Порядок работы

Задание 1. Исследование работы шифратора

Создайте схему, изображенную на рис. 11. Над генератором слов написаны цифры от 0 до 7 – они обозначают номера входов шифратора, на которые соответственно подаются сигналы управления. Сам шифратор составлен из трех элементов «ИЛИ». Выходы шифратора обозначаются как A , B , C , где A – старший бит, B – средний бит, а C – младший бит двоичного числа, получаемого на выходе. (Это число показывает, на какой из входов подан логический сигнал).

Запрограммируйте генератор слов так, чтобы на его выходах сформировалась двоичная последовательность, эмулирующая поочередную подачу на вход шифратора сигнала логической единицы. Пошагово изменяя значения входов дешифратора (кнопкой «Step» в генераторе слов), заполните таблицу истинности шифратора (табл. 6).

Таблица 6.
Таблица истинности шифратора

Входы шифратора								Выходы шифратора			Десятичное число
								Двоичное число			
0	1	2	3	4	5	6	7	C	B	A	
0	0	0	0	0	0	0	0				

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

Переведите полученное двоичное число, составленное из разрядов CBA , в десятичное. Сделайте вывод о работе шифратора.

Задание 2. Исследование работы дешифраторов

Исследование развернутой схемы дешифратора.

Создайте схему изображенную на рис. 12.

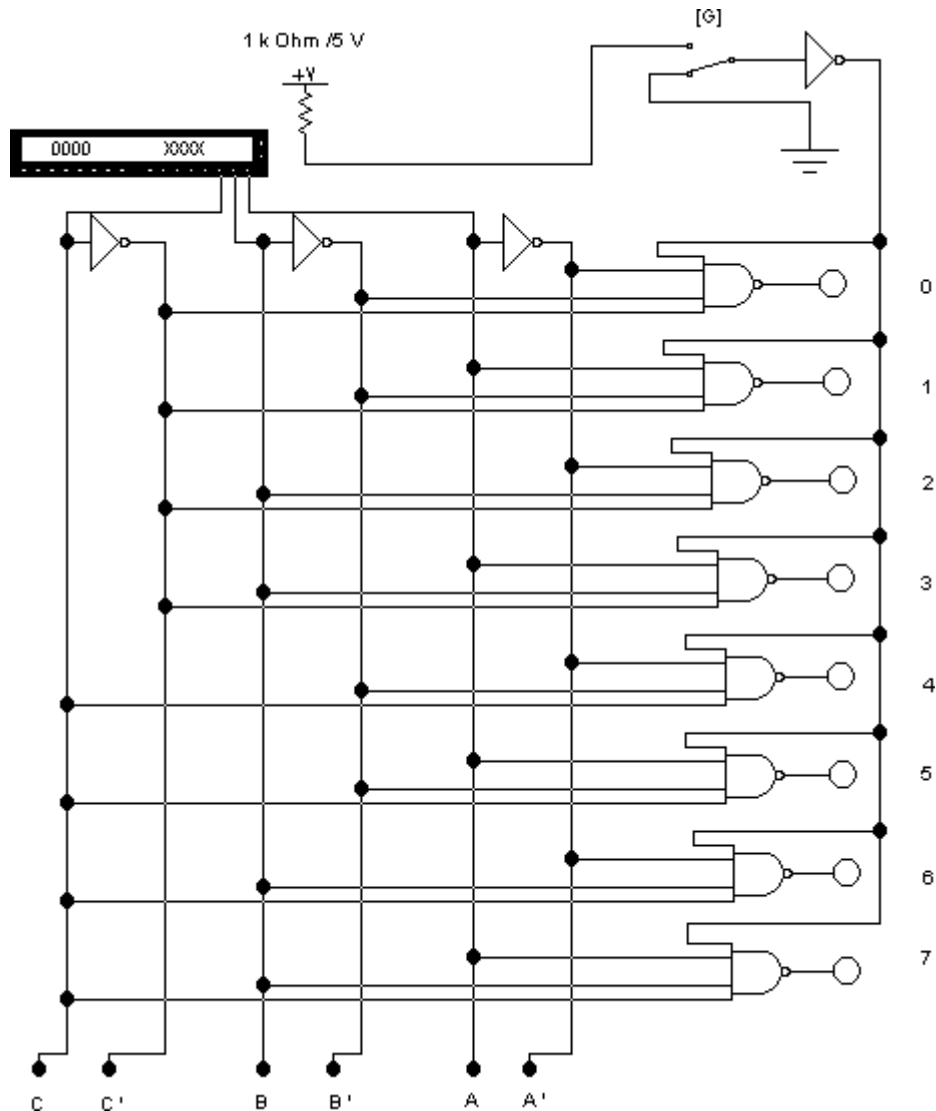


Рис. 12. Развернутая схема дешифратора.

Здесь представлен дешифратор 3×8 (3 входа, 8 выходов). Он составлен из элементов «И-НЕ». C, B, A – входы дешифратора, $0, 1, \dots, 7$ – выходы, G –

вход разрешения.

Запрограммируйте генератор слов так, чтобы на его выходе сформировать все возможные комбинации трехразрядного двоичного числа. Подавая на вход дешифратора различные комбинации двоичного числа C, B, A (кнопкой «Step» в генераторе слов) и разрешения G (ключом G), заполните таблицу истинности дешифратора (табл. 7).

Таблица 7.

Таблица истинности развернутой схемы дешифратора

Входы дешифратора					Выходы дешифратора							
Число	C	B	A	G	0	1	2	3	4	5	6	7
0	0	0	0	0								
1	0	0	1	0								
2	0	1	0	0								
3	0	1	1	0								
4	1	0	0	0								
5	1	0	1	0								
6	1	1	0	0								
7	1	1	1	0								
0	0	0	0	1								
1	0	0	1	1								
2	0	1	0	1								
3	0	1	1	1								
4	1	0	0	1								
5	1	0	1	1								
6	1	1	0	1								
7	1	1	1	1								

Сделайте вывод о работе дешифратора.

*Исследование работы схемы дешифратора 3*8 в основном режиме.*

Создайте схему, изображенную на рис. 13.

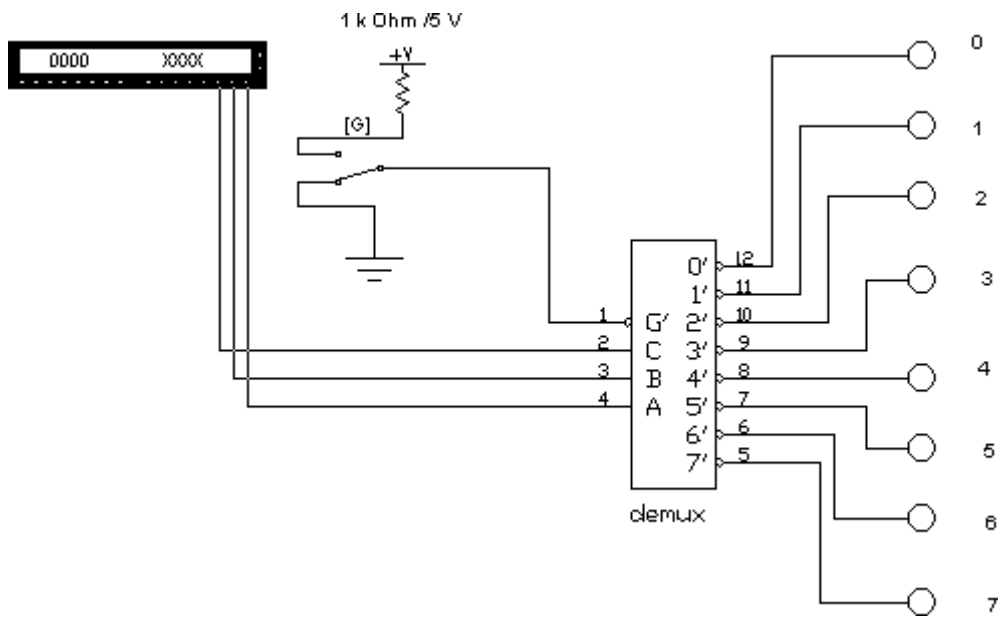


Рис. 13. Схема дешифратора.

Запрограммируйте генератор слов аналогично предыдущему пункту. Подавая на вход дешифратора различные комбинации двоичного числа C, B, A (кнопкой «Step» в генераторе слов) и разрешения G (ключом G), заполните таблицу истинности дешифратора (данная таблица составляется аналогично табл. 7).

Сделайте вывод о работе дешифратора. Сравните полученную в данном пункте таблицу с табл. 7.

*Исследование работы схемы дешифратора 3*8 в режиме 2*4.*

Создайте схему, изображенную на рис. 14.

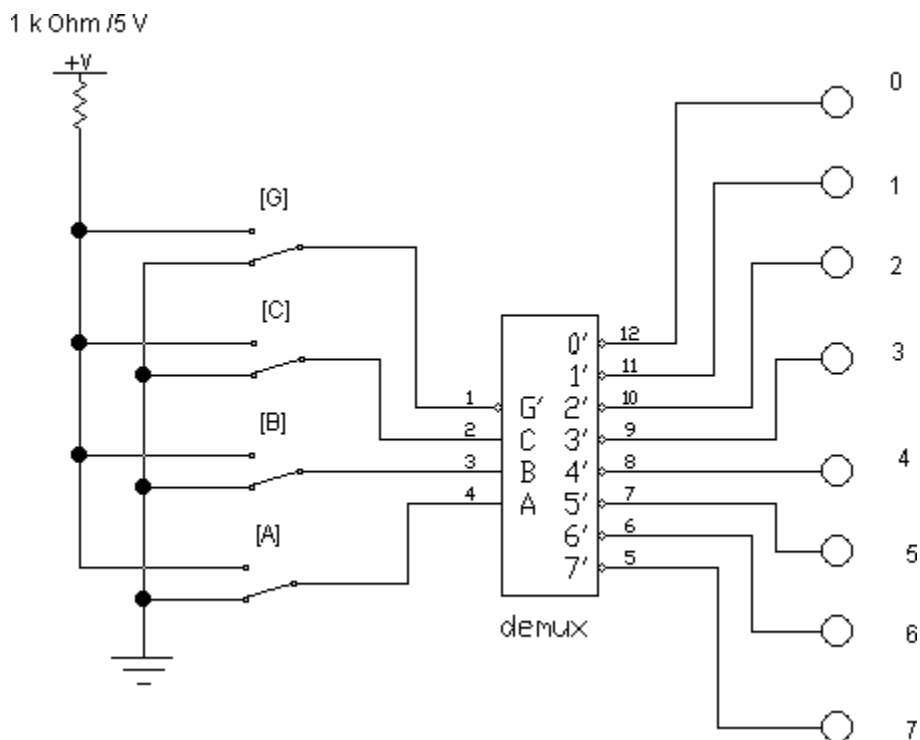


Рис. 14. Дешифратор в режиме 2*4.

В этой схеме подключите вход C к общему проводу (земле), задав $C=0$. Изменяя сигналы на входах B и A , наблюдайте уровни сигналов на выходах схемы (с помощью логических пробников, установленных на выходах). Укажите выходы, на которых уровень сигнала не меняется.

Подключите вход C к источнику питания (логической единице), задав $C=1$. Аналогично изменяя сигналы на входах B и A , наблюдайте уровни сигналов на выходах схемы (с помощью логических пробников, установленных на выходах). Укажите выходы, на которых уровень сигнала не меняется.

Заземлите вход B ($B=0$), подавая на входы A и C все возможные комбинации логических уровней, сделайте вывод о работе схемы в этом случае.

Сформулируйте принцип, по которому можно использовать дешифраторы на меньшую разрешающую способность, чем та, на которую он рассчитан.

Задание 3. Применение дешифраторов

Применение дешифраторов в цифровой технике весьма различно. Наиболее часто они используются как формирующие элементы, – например, в схемах различных микроконтроллеров для формирования сигнала выбора определенной микросхемы. В таком случае на входы дешифратора подаются сигналы с шины адреса микропроцессора. При этом каждому участку адресного пространства ставится в соответствие определенное назначение (например, для ОЗУ выделяется первых 1024 байт, для ПЗУ – следующие 2048 байт и т.д.). Дешифратор в таком случае помогает сформировать сигнал управления (выбора микросхемы), так как согласно его таблице истинности на каждом выходе активный уровень формируется лишь однажды, при вполне определенной комбинации входных сигналов. Таким образом, не составляет особого труда сначала составить карту памяти (подробная запись всего содержимого адресного пространства, с записью начального и конечного адреса каждого блока), а затем по ней выделить те разряды адреса, которые однозначно определяют обращение к тому или иному блоку адресного пространства. Именно эти разряды и будут входами для дешифраторов. Выход каждого дешифратора соединен с входом разрешения работы той микросхемы, которая в данном случае необходима.

В задании внимание в основном будет акцентировано на другом способе

применения дешифраторов, который в общем является базовым и для всех других. Это использование дешифратора в совокупности с логическими элементами.

Помимо того, будет введено понятие временной диаграммы, которая также очень часто используется в технике для иллюстрации работы цифровых устройств.

Временная диаграмма в общем случае представляет собой график, по оси абсцисс которого откладывается время в тактах, а по оси ординат – необходимая цифровая величина (вход или выход цифровой схемы), принимающая значения «0» или «1». Необходимо отметить, что чаще всего осей ординат несколько и они располагаются друг под другом. Собственно говоря, именно в этом и состоит основное преимущество применения временных диаграмм, работа цифрового устройства наглядно представляется во времени. Простейший пример временной диаграммы приведен на рис. 15.

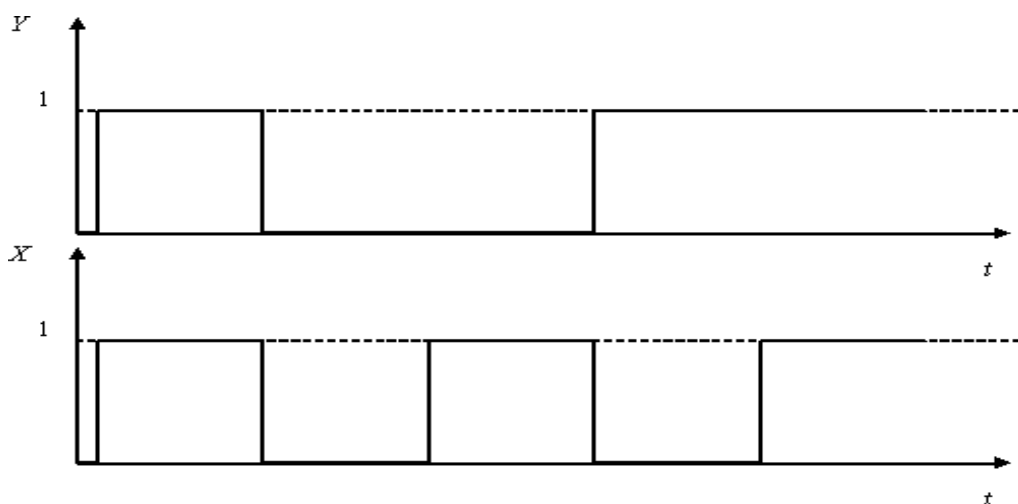


Рис. 15. Временная диаграмма.

По таблице истинности логической функции можно без труда построить временную диаграмму. В *EWB*, кроме инструментов автоматического построения таблицы истинности, присутствуют инструменты и для построения временных диаграмм. Его можно осуществить с помощью «Logic Analyzer». Общий вид окна свойств данного инструмента приведен на рис. 16 (окно свойств вызывается двойным щелчком по изображению).

Построение временных диаграмм начинается автоматически при включении всей схемы. В меню «Clock per division» можно задавать масштаб про-

смотря по оси абсцисс (в данном случае времени). В области «Clock» кнопкой «Set» можно изменить установки внутреннего генератора времени (ось абсцисс), в частности задать частоту анализа («Internal clock rate»). В области «Trigger» можно изменить установки триггера (ось ординат).

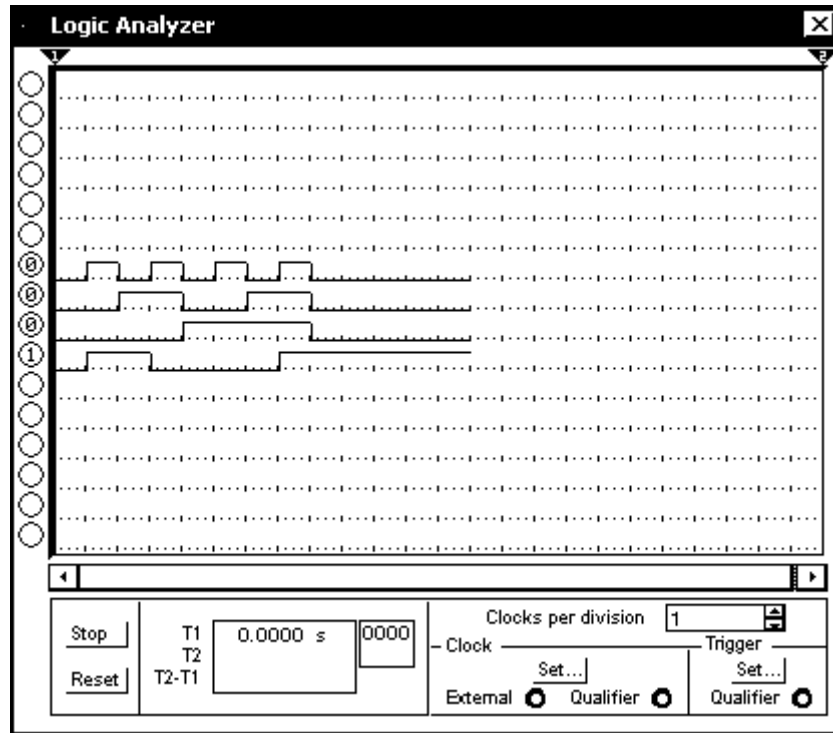


Рис. 16. Окно свойств логического анализатора.

Соберите схему, изображенную на рис. 17.

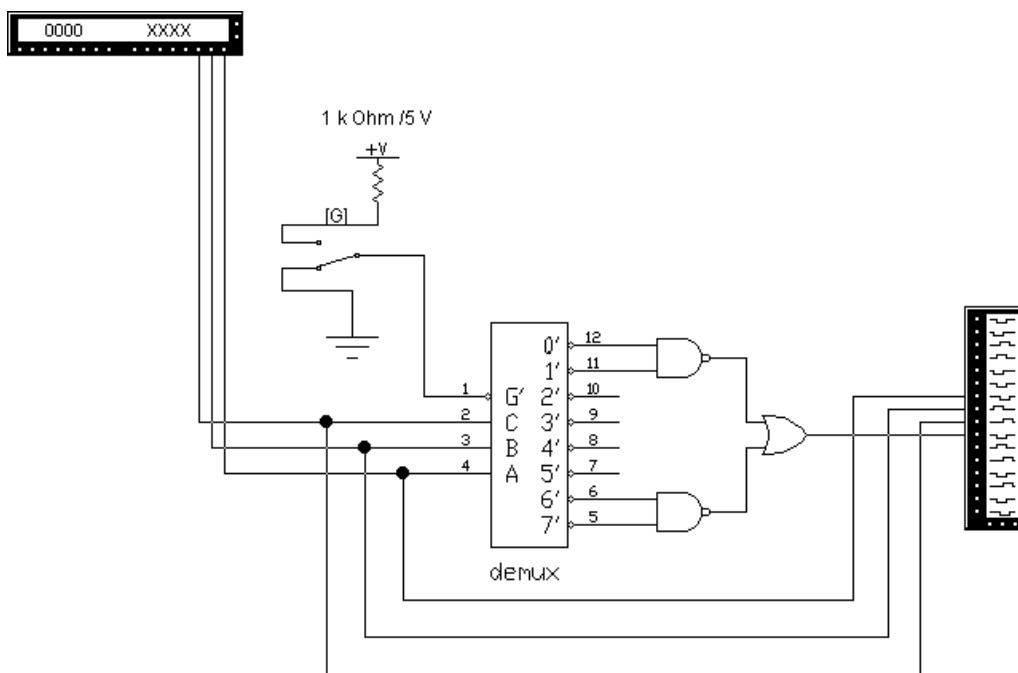


Рис. 17. Дешифратор с логическими элементами.

Запрограммируйте генератор слов так, чтобы на его выходе сформировать все возможные комбинации трехразрядного двоичного числа.

Установите ключ G в разрешающее положение. Подавая на вход дешифратора различные комбинации двоичного числа C, B, A (кнопкой «Step» в генераторе слов), постройте временную диаграмму работы дешифратора с логическими элементами на выходе. Оси диаграммы выберите согласно рис. 18.



Рис. 18. Временная диаграмма.

Получите логическую функцию данного цифрового устройства. Для этого удобнее предварительно построить таблицу истинности, а по ней, аналогично пунктам, изложенным выше, построить саму логическую функцию.

Задание 4. Исследование работы мультиплексора

Исследование развернутой схемы мультиплексора.

Создайте схему, изображенную на рис. 19.

Здесь приняты следующие условные обозначения: G – сигнал разрешения; $d0, d1$ – информационные входы; A – адресный вход.

Исследуйте поведение схемы мультиплексора, задавая различные сочетания логических уровней на входе схемы, заполнив при этом таблицу истинности, приведенную ниже (табл. 8).

Входы		Адрес	Выход
d0	d1	A	Q
0	0	0	
0	1		
1	0		
1	1		
0	0	1	
0	1		
1	0		
1	1		

Задание 5. Реализация логической функции с помощью мультиплексора

По заданной таблице истинности (табл. 11) реализуйте логическую функцию с помощью мультиплексора. Выбор значений выхода логической функции производится согласно вашему варианту. Реализация логической функции может быть осуществлена с помощью стандартного мультиплексора.

Таблица 11.
Варианты таблицы истинности

Значения логических переменных (для всех вариантов)			Варианты задания									
			1	2	3	4	5	6	7	8	9	10
A	B	D	Значения логической функции (для каждого варианта)									
0	0	0	0	0	0	0	1	0	1	0	1	1
0	0	1	1	0	1	1	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	0
0	1	1	0	0	1	1	0	0	0	0	1	0
1	0	0	1	1	0	1	0	1	1	1	1	1
1	0	1	1	1	0	0	1	1	0	1	1	1
1	1	0	1	0	1	1	0	1	0	0	0	1
1	1	1	0	0	1	1	1	1	0	0	0	1

Создайте схему с вашим вариантом реализации логической функции в EWB, занесите полученную схему в отчет. С помощью логического анализатора постройте временные диаграммы работы вашей логической функции. По ним проверьте правильность функционирования схемы.

Контрольные вопросы

1. Дайте определение шифратору, дешифратору.
2. Чем отличается схема шифратора от схемы дешифратора?
3. Как в простейшем случае реализовать на дешифраторе демультиплексор? Постройте полученную схему в случае использования де-

шифратора 3×8 .

4. Как влияет сигнал управления на работу логической схемы?
5. Как из двух дешифраторов 2×4 сделать один 3×8 ?
6. Как изменить расширенную схему дешифратора (рис. 12), чтобы активным уровнем выходного сигнала данной схемы была «1»?
7. Дайте определение мультиплексу.
8. Приведите примеры применения мультиплексов.
9. Как на мультиплексе можно реализовать логическую функцию?
10. Любую ли логическую функцию можно реализовать на мультиплексе?
11. Дайте определение временной диаграммы.
12. Можно ли по произвольной временной диаграмме составить таблицу истинности?
13. Можно ли по таблице истинности составить временную диаграмму?
14. В каком виде записывается логическая функция мультиплексора?

Упражнения

1. Применение дешифраторов

Сформируйте сигнал выбора определенной микросхемы памяти в микроконтроллере с использованием стандартных дешифраторов.

Микропроцессор, входящий в состав микроконтроллера, имеет 8-разрядную шину адреса, т.е. может работать с объемом памяти $2^8=256$ байт (разряды нумеруются $A_0 - A_7$, начиная с младшего). По каждому адресу хранится информация, объем которой определяется разрядностью шины данных микропроцессора и в данном упражнении не важен. В общем случае вся память делится на определенные блоки по характеру их использования (например, для ОЗУ, ПЗУ, портов ввода/вывода). Каждый блок памяти имеет начальный и конечный адрес.

Необходимо внутри объема всей доступной памяти сформировать блок, начальный и конечный адреса которого приведены в табл. 12. Данный блок будет использоваться для адресации ОЗУ.

Таблица 12.

Варианты карты памяти для блока ОЗУ

Вариант	Адреса															
	начальный								конечный							
	A7	A6	A5	A4	A3	A2	A1	A0	A7	A6	A5	A4	A3	A2	A1	A0
1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
2	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1
3	0	0	1	0	0	0	0	0	0	0	1	0	1	1	1	1
4	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1
5	0	1	0	0	0	0	0	0	0	1	0	0	1	1	1	1
6	0	1	0	1	0	0	0	0	0	1	0	1	1	1	1	1
7	0	1	1	0	0	0	0	0	0	1	1	0	1	1	1	1
8	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1
9	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
10	1	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1

Для выполнения задания необходимо проделать следующее:

проанализировать все адреса, входящие в ваш блок, выделить те разряды шины адреса, значения которых одинаковы для всех адресов вашего блока;

из всех полученных комбинаций выбрать достаточное количество разрядов шины адреса, которые однозначно определяют обращение именно к вашему блоку памяти и не повторяются во всем другом объеме доступной памяти; полученные разряды шины адреса и будут входами дешифратора, а его

выход и есть искомый сигнал выбора определенной микросхемы.

После выполнения вышеуказанных процедур необходимо выбрать подходящий дешифратор и занести полученную схему в отчет, отметив номера используемых разрядов шины адреса.

2. Применение мультиплексоров

По заданной логической функции (табл. 13) постройте схему с использованием мультиплексора.

Вариант	Логическая функция
1	$C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot \overline{A} + C \cdot B \cdot A + C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot B \cdot A$
2	$C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot B \cdot A$
3	$C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot A + C \cdot B \cdot A$
4	$C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot B \cdot A$
5	$C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot \overline{A} + C \cdot B \cdot A + C \cdot \overline{B} \cdot A + C \cdot B \cdot A$
6	$C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot A$
7	$C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot \overline{A} + C \cdot B \cdot A + C \cdot \overline{B} \cdot A + C \cdot B \cdot A$
8	$C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot B \cdot A$
9	$C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot A + C \cdot B \cdot A$
10	$C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot A + C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot A$

ЛАБОРАТОРНАЯ РАБОТА № 3

ИЗУЧЕНИЕ РАБОТЫ ТРИГГЕРОВ

Цель работы.

1. Изучение структуры и исследование работы асинхронных и синхронных триггеров.
2. Исследование функций переходов и возбуждения основных типов триггеров.
3. Изучение взаимозаменяемости триггеров различных типов.

Приборы и элементы.

Логические пробники (панель «Indicators/Red probe»).

Источник напряжения + 5 В (панель «Basic/Pull-Up Resistor»).

Земля (панель «Sources/Ground»).

Двухвходовые элементы И, И-НЕ, ИЛИ, ИЛИ-НЕ (панель «Logic Gates/2-Input AND, NAND, OR, NOR Gates»).

Двухпозиционные переключатели (панель «Basic/Switch»).

Базовые триггеры RS, JK, D (панель «Digital/»).

Схемы различных серий (панель «Digital/MUX,DEC»).

Краткие теоретические сведения

Последовательные цифровые устройства часто называют последовательными схемами, последовательными автоматами, дискретными автоматами с памятью, многотактными автоматами. Простейшим примером устройств данного типа являются триггеры.

Триггеры

Триггер – простейшая цифровая схема последовательного типа. У рассмотренных в предыдущих работах комбинационных схем состояние выхода Y в любой момент времени определяется только текущим состоянием входа X . В отличие от них состояние выхода последовательной схемы (цифрового автомата) зависит еще и от внутреннего состояния схемы Q .

Таким образом, цифровой автомат является не только преобразователем, но и хранителем предшествующей и источником текущей информации (состояния). Данное свойство обеспечивается наличием в схемах обратных связей.

Триггер имеет два устойчивых состояния: $Q=1$ и $Q=0$, поэтому его иногда называют бистабильной схемой. В каком из этих состояний окажется триггер, зависит от сигналов на информационном входе триггера и от его предыдущего состояния, т.е. он имеет память. Можно сказать, что триггер является элементарной ячейкой памяти.

Тип триггера определяется алгоритмом его работы. В зависимости от алгоритма триггер может иметь установочные, информационные и управляющие входы. Установочные входы служат для перевода триггера в одно из определенных состояний, независимо от состояния других входов. Входы управления разрешают запись данных, подающихся на информационные входы. Наиболее распространенными являются триггеры RS , JK , D и T типов.

1. Триггер типа RS

RS -триггер – простейший автомат с памятью, который может находиться в двух состояниях. Триггер имеет два информационных входа: установки S (*set* – установка) и сброса R (*reset* – сброс), на которые подаются входные сигналы от внешних источников. При подаче на вход установки активного логи-

ческого уровня триггер устанавливается в «1» ($Q=1, \bar{Q}=0$), при подаче активного уровня на вход сброса – в «0» ($Q=0, \bar{Q}=1$).

Для триггеров этого типа недопустима одновременная подача активного уровня на оба входа установки и сброса, поскольку триггер по определению не может одновременно быть установлен в ноль и единицу. На практике подача активного уровня на оба установочных входа приводит к тому, что это состояние не может быть сохранено и невозможно определить, в каком состоянии будет находиться триггер при последующей подаче на установочные входы сигналов пассивного уровня.

На рис. 1 и 2 показаны два вида *RS-триггера*, выполненных на элементах «ИЛИ-НЕ» и «И-НЕ».



Рис. 1. *RS-триггер* на элементах «ИЛИ-НЕ».

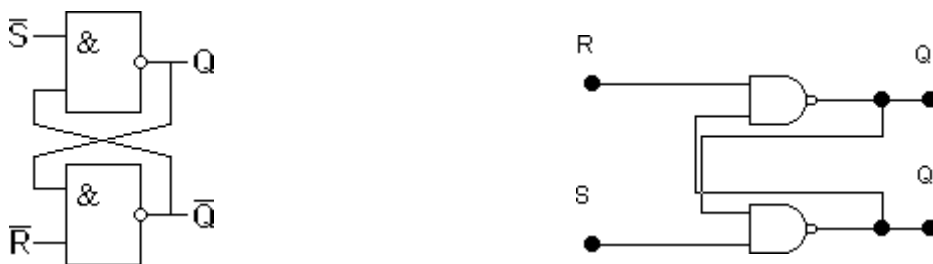


Рис. 2. *RS-триггер* на элементах «И-НЕ» (\overline{RS} -триггер).

Для схемы на рис. 1 активным уровнем является уровень логической единицы, для схемы на рис. 2 – уровень логического нуля. Схема на рис. 2 получила название *RS-триггера* с инверсными входами, или просто – *RS-триггер*.

RS-триггер является основным узлом для построения последовательных схем. Напомним: название схем такого типа «последовательные» означает, то что состояние выхода зависит от того, в какой последовательности на входы подаются сигналы и каково было предшествующее внутреннее состояние. Так,

если в *RS-триггере* (рис. 1) сначала установить комбинацию $R=0, S=1$, а потом перейти к $R=0, S=0$, то состояние выхода будет $Q=1$. Если же сначала установить комбинацию $R=1, S=0$, а потом перейти к $R=0, S=0$, то состояние выхода будет другим – $Q=0$. Таким образом, при одном и том же входном наборе $R=0, S=0$ выход триггера может находиться в разных состояниях.

Условия переходов триггеров из одного состояния в другое (алгоритм работы) можно описать табличным, аналитическим или графическим способами. Табличное описание работы *RS-триггера* (рис. 1) представлено в табл. 1 (таблица переходов и таблица функций возбуждения).

Таблица 1.
Таблица переходов и функций возбуждения RS-триггера

Таблица переходов			Таблица функций возбуждения			
R	S	Q_{t+1}	Q_t	Q_{t+1}	R	S
0	0	Q_t	0	0	-	0
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	1	x	1	1	0	-

В таблице использованы следующие обозначения: « Q_t » – предшествующее состояние выхода; « Q_{t+1} » – новое состояние, устанавливающееся после перехода (возможно $Q_{t+1} = Q_t$); «-» – безразличное значение сигнала: «0» или «1»; «x» – запрещенное состояние.

Аналитическое описание (характеристическое уравнение) можно получить из табл. 1 по правилам алгебры логики

$$Q_{t+1} = \bar{R}S + \bar{R}Q_t = \bar{R}(S + Q_t). \quad (1)$$

Зависимость Q_{t+1} от Q_t характеризует свойство запоминания предшествующего состояния.

Табл. 1 показывает, что схема, которая находилась в состоянии $Q=0$, сохраняет это состояние как при воздействии входного набора $R=0, S=0$, так и при воздействии $R=1, S=0$. Если же на вход схемы, находящейся в состоянии $Q=0$, подействовать набором $R=0, S=1$, то она переходит в состояние $Q=1$ и сохраняет его при входных наборах $R=0, S=1$ либо $R=0, S=0$.

Таблицы переходов и функций возбуждения легче запомнить, держа в уме следующее (для *RS-триггера*): вход S – вход установки, т.е. при подаче на

него логической единицы триггер должен перейти в активное состояние; установить триггер в ноль можно только подав сигнал сброса R ; запрещенным является сигнал одновременного сброса и установки $R=1, S=1$; сигнал $R=0, S=0$ – сохраняет предыдущее состояние.

Схема триггера позволяет запоминать состояние логической схемы, но так как в начальный момент времени может возникать переходный процесс (в цифровых схемах этот процесс называется «опасные гонки»), то запоминать состояния логической схемы нужно только в определенные моменты времени, когда все переходные процессы закончены. Таким образом, цифровые схемы требуют синхросигнала. Схемы с сигналом управления (синхронизации) представлены ниже. Необходимо отметить также, что существуют схемы и синхронных *RS-триггеров*.

2. JK-триггер

Триггер JK-типа имеет более сложную по сравнению с *RS-триггером* структуру и более широкие функциональные возможности. Помимо информационных входов J и K , прямого и инверсного выходов Q и \bar{Q} , *JK-триггер* имеет вход синхронизации C (его также называют управляющим, тактирующим, счетным), а также асинхронные установочные R и S входы (такими же входами могут снабжаться и обычные *RS-триггеры*). Обычно активными уровнями установочных сигналов являются нули, как в схеме \overline{RS} -триггера, изображенной на рис. 2. Установочные входы имеют приоритет над остальными. Активный уровень сигнала на входе S устанавливает триггер в состояние $Q=1$, а активный уровень сигнала на входе R – в состояние $Q=0$, независимо от сигналов на остальных входах.

Если на входы установки одновременно подать пассивный уровень сигнала, то состояние триггера будет изменяться по фронту импульса на счетном входе в зависимости от состояния входов J и K , как показано в табл. 2.

Таблица переходов и возбуждения JK-триггера

Таблица переходов			Таблица функций возбуждений			
J	K	Q_{t+1}	Q_t	Q_{t+1}	K	J
0	0	Q_t	0	0	-	0
0	1	0	0	1	0	1

1	0	1	1	0	1	0
1	1	Q_t	1	1	0	-

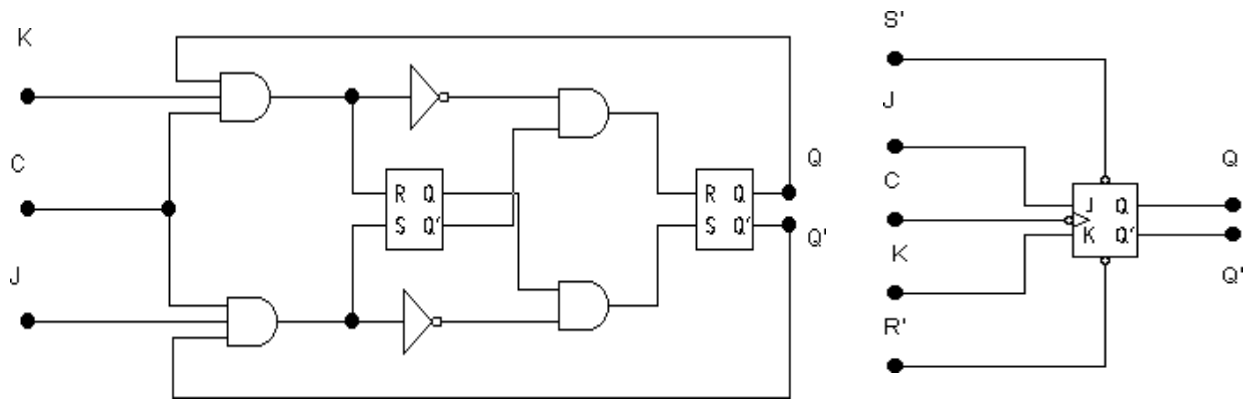


Рис. 3. Схема JK-триггера:

1) расширенная схема,

2) условное обозначение.

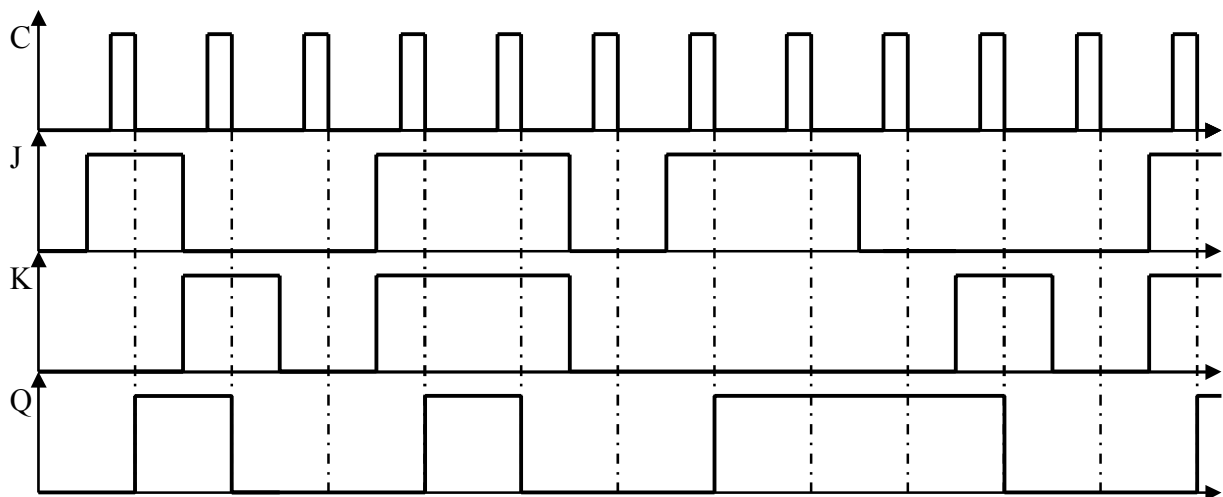


Рис. 4. Временная диаграмма JK-триггера.

В данном случае все изменения выхода происходят только в момент отрицательного перепада тактового сигнала (по заднему фронту импульса C).

3. D-триггер

D-триггер имеет один информационный вход D (*data* – данные). Информация с входа D заносится в триггер по положительному перепаду импульса на входе синхронизации C и сохраняется до следующего положительного перепада. Помимо входов C и D , триггер снабжен асинхронными установочными R и S входами. Установочные входы имеют наивысший приоритет. Они устанавливают триггер независимо от сигналов на входах C и D . Функци-

онирование *D-триггера* описывается табл. 3 и рис. 5.

Таблица 3.
Таблица переходов и возбуждения D-триггера

Таблица переходов		Таблица функций возбуждений		
D	Q _{t+1}	Q _t	Q _{t+1}	D
0	0	0	0	0
		0	1	1
1	1	1	0	0
		1	1	1

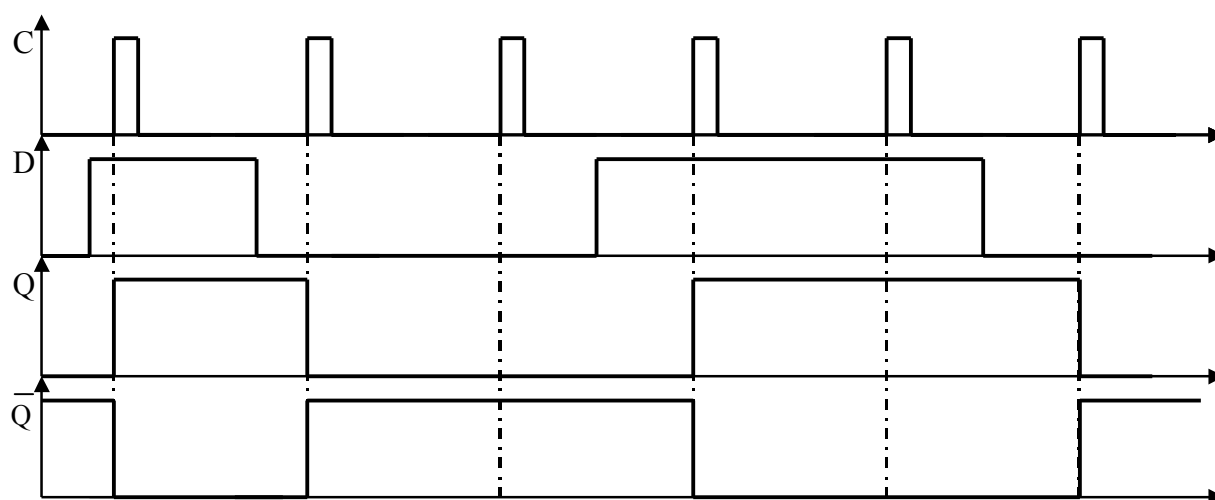


Рис. 5. Временная диаграмма *D-триггера*.

Характеристическое уравнение D-триггера:

$$Q_{t+1} = D_t. \tag{2}$$

Уравнение триггера показывает, что состояние триггера на $t+1$ такте равно входному сигналу в момент, предшествующий тактовому перепаду сигнала C . Условное обозначение *D-триггера* представлено на рис 6.1. Функциональная схема *D-триггера* может быть получена из схемы *JK-триггера* путем подключения входа D к входу J через инвертор (рис. 6.2).

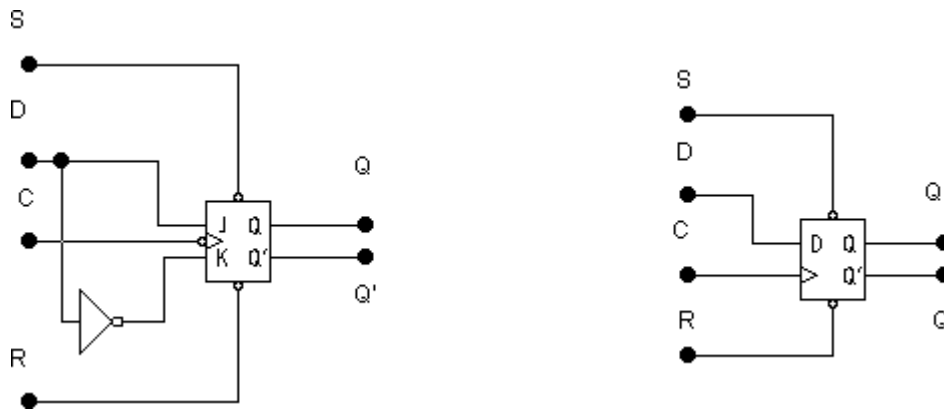


Рис. 6 Схема D-триггера:

1) расширенная схема,

2) условное обозначение.

4. T-триггер (счетный триггер)

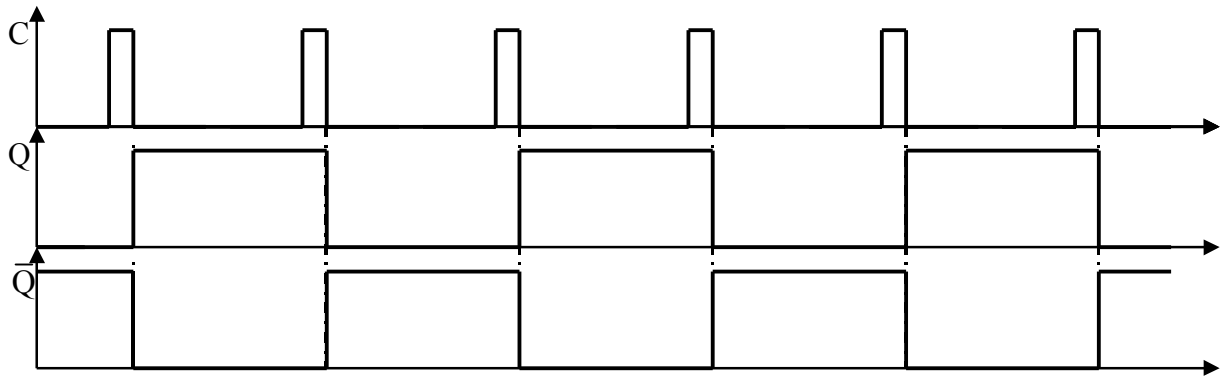
На основе *JK-триггеров* и *D-триггеров* можно построить схемы, осуществляющие так называемый счетный режим. Такие схемы называют *T-триггерами*, или счетными триггерами, что связано со способом их функционирования. На рис. 7 представлены схемы организации *T-триггера* на основе *JK-триггера* (рис. 7.1) и *D-триггера* (рис. 7.2). Счетный режим иллюстрируется временными диаграммами (рис. 8).

В *JK-триггере* с входами установки логическим нулем счетный режим реализуется путем подачи констант $J=K=1$ и $R'=S'=1$ и сигнала T на вход C .

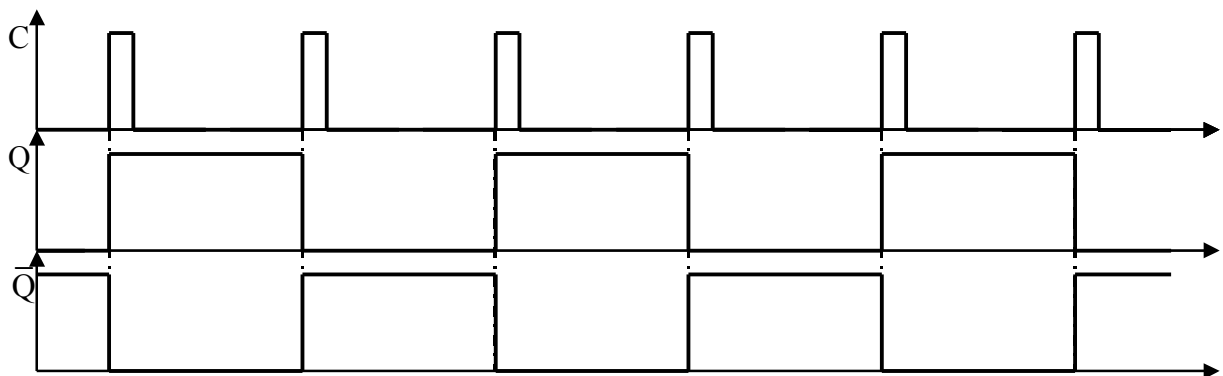
В соответствии с таблицей функционирования при каждом отрицательном перепаде входного сигнала T состояние триггера изменяет свое значение на противоположное.

1) на базе *JK-триггера*,

2) на базе *D-триггера*.



1) на базе JK-триггера



2) на базе D-триггера

Рис. 8. Временная диаграмма T-триггера.

В D-триггере счетный режим реализуется при помощи обратной связи (на вход D подается сигнал с инверсного выхода). Таким образом, всегда существует неравенство сигнала на входе D и сигнала на выходе Q: если $Q=1$, $D=0$.

Следовательно, при каждом положительном перепаде сигнала на счетном входе C в соответствии с принципом действия D-триггера состояние выхода будет изменяться на противоположное.

Таким образом, на каждые два входных тактовых импульса T-триггер формирует один период выходного сигнала Q. Следовательно, триггер осуществляет деление частоты f_T на его входе на 2:

$$f_Q = \frac{f_T}{2}. \quad (3)$$

Порядок работы

Задание 1. Исследование работы схемы RS триггера

1.1. Исследование RS триггера, составленного из двух элементов «ИЛИ-НЕ».

Создайте схему, изображенную на рис. 9.

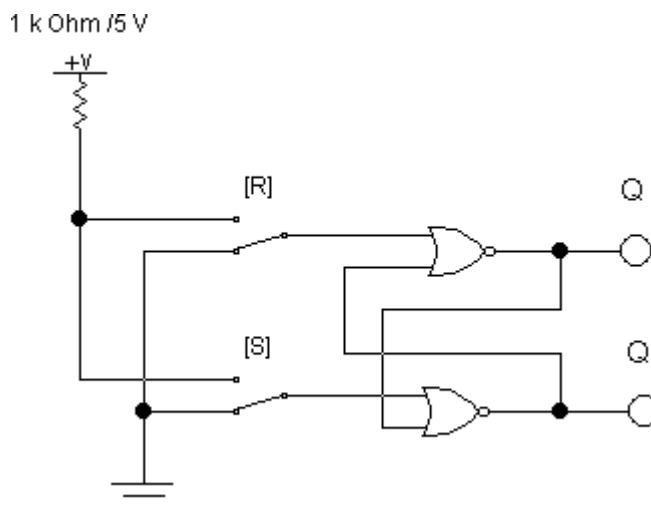


Рис. 9. Схема RS триггера на базе элементов «ИЛИ-НЕ».

Включите схему. Убедитесь в правильности работы триггера (перехода из одного состояния в другое), проверив следующие утверждения:

- при $S=0, R=1$ триггер сбрасывается в состояние $Q=0$;
- при $S=0, R=0$ триггер сохраняет свое прежнее состояние $Q=0$;
- при $S=1, R=0$ триггер устанавливается в состояние $Q=1$;
- при $S=0, R=0$ триггер сохраняет прежнее состояние $Q=1$;
- при $S=1, R=1$ триггер находится в запрещенном состоянии.

Заполните таблицу возбуждения данного типа триггера (табл. 4).

Таблица 4.
RS-триггер на базе элементов «ИЛИ-НЕ»

Q_t	Q_{t+1}	R	S
0		0	0
0		1	0
0		0	1
1		0	0
1		1	0
1		0	1

Примечание. Таблица возбуждения триггера составляется для различных

комбинаций R и S -входов. При этом предварительно необходимо добиться начального состояния триггера Q_i , помня, что установить триггер в состояние $Q_i=0$ можно подав сигнал $R=1$ ($S=0$), а установить триггер в состояние $Q_i=1$ – подав сигнал $S=1$ ($R=0$).

Задание 2. Исследование работы JK-триггера

Составление функции возбуждения JK-триггера.

Соберите схему, изображенную на рис. 11.

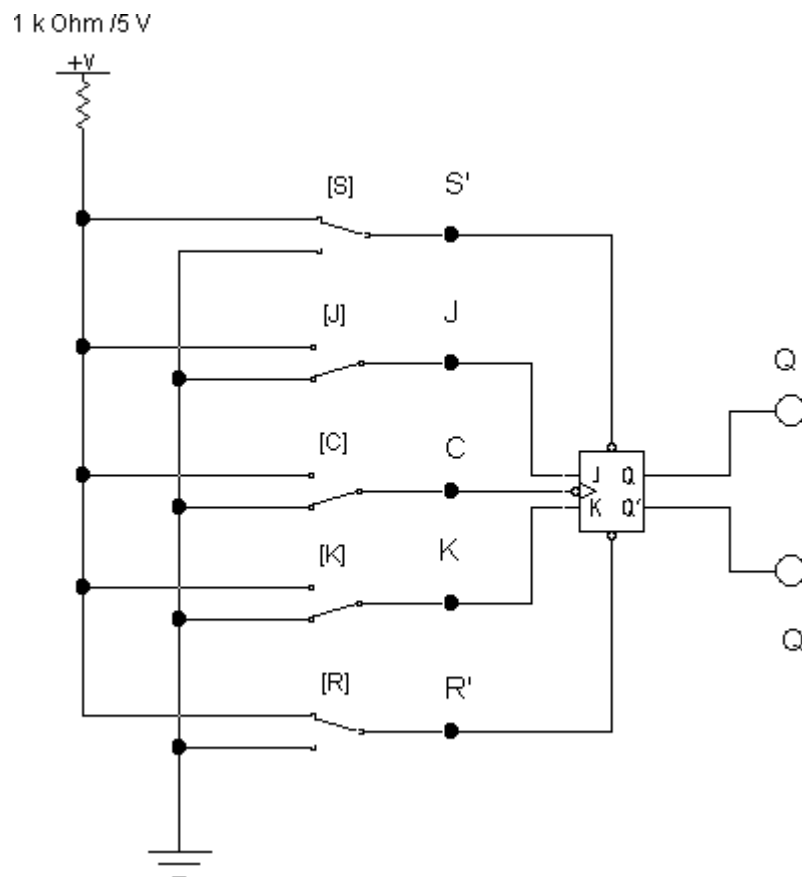


Рис. 11. Схема JK-триггера.

Отметим, что на схеме используется упрощенная микросхема JK-триггера, в которой установочные входы \bar{R} и \bar{S} инверсные (на микросхеме такие входы имеют кружок на линии, входящей в корпус).

Включите схему. Убедитесь в правильности работы установочных входов триггера, проверив следующие утверждения:

при $\bar{S} = 1, \bar{R} = 0$ триггер сбрасывается в состояние $Q=0$ независимо от входов J, K, C ;

при $\bar{S} = 0$, $\bar{R} = 1$ триггер устанавливается в состояние $Q=1$ независимо от входов J , K , C .

Таким образом, для установки триггера в требуемое исходное состояние Q_t необходимо подать на его вход одну из перечисленных комбинаций, затем установить $\bar{S} = 1$, $\bar{R} = 1$.

Установив триггер в необходимое исходное состояние, заполните таблицу возбуждения *JK-триггера* (табл. 6).

Таблица 6.
Таблица возбуждения JK-триггера

Q_t	Q_{t+1}	J	K
0		0	0
0		0	1
0		1	0
0		1	1
1		0	0
1		0	1
1		1	0
1		1	1

Примечание. Переход триггера в новое состояние происходит по заднему фронту импульса C . Таким образом, для получения нового состояния Q_{t+1} нужно установить начальное состояние триггера Q_t (ключами \bar{R} и \bar{S}), затем необходимую комбинацию входов J и K (при этом $\bar{S} = 1$, $\bar{R} = 1$) и произвести переключение ключа C .

Составление временных диаграмм JK-триггера.

По таблице возбуждения *JK-триггера* (табл. 6) составьте временную диаграмму работы триггера. Оси графика необходимо выбрать аналогично рис. 4. При этом по оси « C » также откладываются последовательно импульсы синхронизации. Значения, откладываемые на других осях временной диаграммы, необходимо выбирать построчно из табл. 6 по следующим правилам:

до появления импульса C значения J и K должны устанавливаться согласно каждой выбранной строке таблицы возбуждения;

значение по оси Q выбирается равным Q_i (согласно выбранной строке) до каждого момента исчезновения импульса C (до появления так называемого заднего фронта);

после исчезновения импульса C значение по оси Q выбирается равным Q_{i+1} (согласно выбранной строке) до момента установки новых значений J и K (перехода к анализу новой строки таблицы возбуждения);

переход к новым значениям J и K производится после исчезновения исследуемого импульса C , но до появления нового (и далее по каждой строке аналогично).

При этом полученные временные диаграммы будут отличными от приведенных на рис. 4, так как в соответствии с табл. 6 каждый раз триггер необходимо сбрасывать в состояние Q_i , до анализа исследуемой строки.

Задание 3. Исследование работы D-триггера

Исследование D триггера, составленного на базе JK-триггера.

Создайте схему, изображенную на рис. 12.

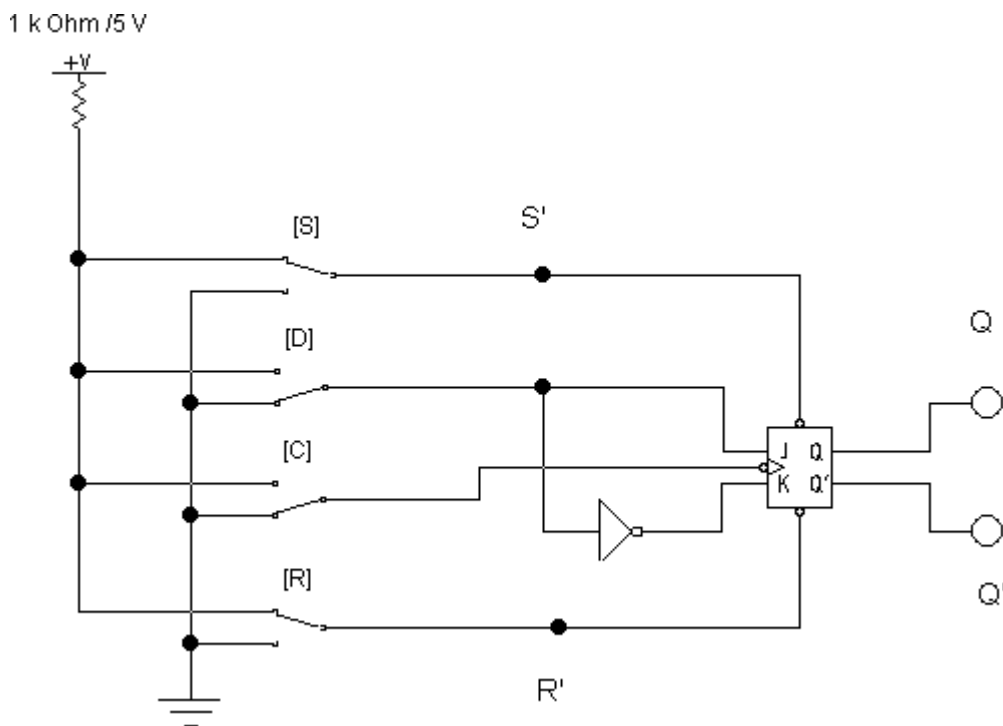


Рис. 12. Схема D-триггера на базе JK-триггера.

Включите схему. Убедитесь в правильности функционирования установочных входов:

при $\bar{S} = 1$, $\bar{R} = 0$ триггер сбрасывается в состояние $Q=0$ независимо от входов D , C ;

при $\bar{S} = 0$, $\bar{R} = 1$ триггер устанавливается в состояние $Q=1$ независимо от входов D , C .

Заполните таблицу переходов (табл. 7), переведя предварительно установочные входы в разрешающее состояние $\bar{S} = 1$ и $\bar{R} = 1$.

Таблица 7.
Таблица переходов D-триггера

D	Q_{t+1}
0	
1	

Получение функций возбуждения D триггера.

Создайте схему, изображенную на рис. 13.

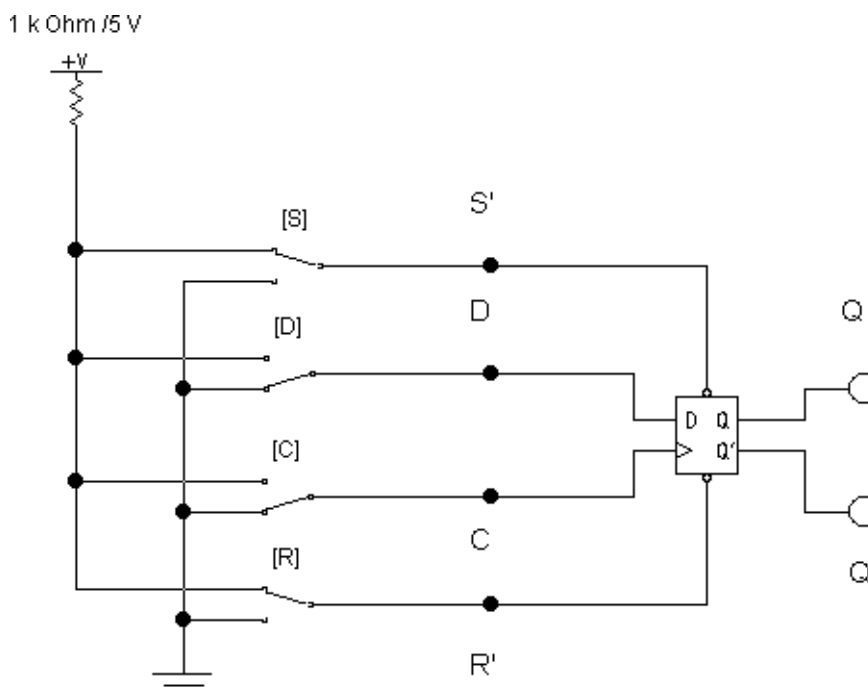


Рис. 13. Схема D-триггера.

Проверьте правильность работы установочных входов \bar{R} и \bar{S} . Убедитесь в идентичной работе схем рис. 12 и рис. 13 с помощью полученной в предыдущем пункте таблицы переходов.

Заполните таблицу возбуждения *D-триггера* (табл. 8), предварительно устанавливая необходимое состояние Q_t сигналами R и \bar{S} (после установки необходимо перевести данные входы в разрешающие положение $\bar{R} = 1, \bar{S} = 1$).

Таблица 8.
Функции возбуждения D-триггера

Q_t	D	Q_{t+1}
0	0	
0	1	
1	0	
1	1	

Примечание. Необходимо помнить, что переключение триггера в новое состояние происходит по переднему фронту импульса C .

3.2. Составление временной диаграммы *D* триггера.

По таблице возбуждения *D-триггера* (табл. 8) составьте временную диаграмму работы триггера. Оси графика необходимо выбрать аналогично рис. 5. При этом на оси « C » так же откладываются последовательно импульсы синхронизации. Значения, откладываемые на других осях временной диаграммы, необходимо выбирать из табл. 8, аналогично пункту 3.1.

Полученные временные диаграммы будут отличны от диаграмм рис. 5, так как в соответствии с табл. 8 каждый раз триггер необходимо сбрасывать в значение Q_t .

Задание 4. Исследование работы *T*-триггера

*Исследование *T*-триггера, составленного на основе *JK*-триггера.*

Соберите *T-триггер* на основе *JK-триггера* согласно схеме.

Включите схему. Изменяя состояние входа C соответствующим ключом, постройте временную диаграмму *T-триггера*. Сравните полученный результат с рис. 8.

*Исследование *T*-триггера, составленного на основе *D*-триггера.*

Соберите *T-триггер* на основе *D-триггера* согласно схеме, изображенной на рис. 15.

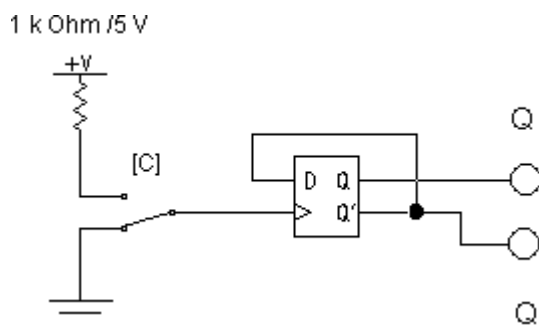


Рис. 15. Схема *T-триггера* на основе *D-триггера*.

Включите схему. Изменяя состояние входа *C* соответствующим ключом, постройте временную диаграмму *T-триггера*.

Сравните полученный результат.

Контрольные вопросы

1. Дайте определение триггера.
2. Чем отличается *RS-триггер* с обычными входами от \overline{RS} -триггера с инверсными входами?
3. Что называется запрещенным и неопределенным состоянием триггера?
4. Приведите примеры запрещенного и неопределенного состояния различных триггерных схем.
5. Чем отличается таблица переходов от таблицы возбуждения?
6. Приведите характеристическое уравнение *RS-триггера*.
7. Чем отличается *JK-триггер* от *RS-триггера*?
8. Приведите характеристическое уравнение *JK-триггера*.
9. Зачем нужны установочные входы в триггерах?
10. Каков приоритет установочных входов в триггерах по сравнению с информационными входами?
11. Охарактеризуйте основное применение *D-триггера*.
12. Приведите характеристическое уравнение *D-триггера*.
13. Чем отличается *D-триггер* от *T-триггера*?

14. Чему равна частота следования импульсов в T -триггере?

Упражнения

Построение временных диаграмм

По заданной таблице истинности некоторой логической функции постройте временную диаграмму. Считайте, что переключение в новое состояние схемы должно происходить по отрицательному фронту импульса C .

Варианты таблицы истинности приведены в табл. 9.

Таблица 9.
Варианты задания логической функции.

Значения логических переменных (для всех вариантов)			Варианты задания									
			1	2	3	4	5	6	7	8	9	10
X	Y	Z	Значения логической функции (для каждого варианта)									
0	0	0	0	0	1	1	0	1	0	0	1	0
0	0	1	0	1	0	0	0	0	1	0	1	0
0	1	0	0	1	1	0	1	1	0	0	0	0
0	1	1	1	0	1	1	0	0	0	1	0	1
1	0	0	0	1	0	1	0	1	1	1	1	1
1	0	1	1	1	0	1	0	1	1	1	1	1
1	1	0	0	0	1	1	0	0	0	1	0	1
1	1	1	0	0	1	0	0	1	0	0	0	0

ЛАБОРАТОРНАЯ РАБОТА № 4

ИЗУЧЕНИЕ СУММАТОРОВ, ПОЛУСУММАТОРОВ, РЕГИСТРОВ И СЧЕТЧИКОВ

Цель работы.

1. Исследование сумматоров и полусумматоров.
2. Изучение структуры и исследование работы суммирующих и вычитающих счетчиков, счетчиков с измененным коэффициентом пересчета.
3. Изучение регистров.

Приборы и элементы.

Генератор слов (панель «Instruments/Word Generator»).

Логические пробники (панель «Indicators/Red probe»).

Источник напряжения + 5 В (панель «Basic/Pull-Up Resistor»).

Земля (панель «Sources/Ground»).

Двухпозиционные переключатели (панель «Basic/Switch»).

Двухвходовые элементы И, И-НЕ, ИЛИ, ИЛИ-НЕ
(панель «Logic Gates/2-Input AND, NAND, OR, NOR Gates»).

Сумматор (панель «Digital/Half-Adder»).

D-триггер (панель «Digital/»).

Декодер (панель «Indicators/Decoded 7 segment display»).

Краткие теоретические сведения

Сумматоры и полусумматоры

Широкое применение в цифровой технике находят элементы, выполняющие различные арифметические действия. Операция суммирования – базовая арифметическая операция в двоичной алгебре. Поэтому для дальнейшего изучения цифровой техники необходимо исследовать способы получения сумматоров.

1. Сумматоры по модулю два

Построение двоичных сумматоров обычно начинается с сумматора по модулю 2. Ниже представлена таблица истинности этого сумматора.

Таблица 1.
Таблица истинности сумматора по модулю два

Входы		Выход
X	Y	Out
0	0	0
0	1	1
1	0	1
1	1	0

Из табл. 1 видно, что логическая функция, выражающая принцип работы сумматора по модулю два, имеет вид:

$$F = X \oplus Y = X\bar{Y} \vee \bar{X}Y \quad (1)$$

и представляет собой описанную ранее (формула 7, в лабораторной работе №1) функцию «исключающего ИЛИ».

На рис. 1.1 представлена схемная реализация сумматора по модулю два, составленная по табл. 1. На рис. 1.2 приведено условное обозначение этой же схемы в виде одного элемента – «Исключающего ИЛИ».

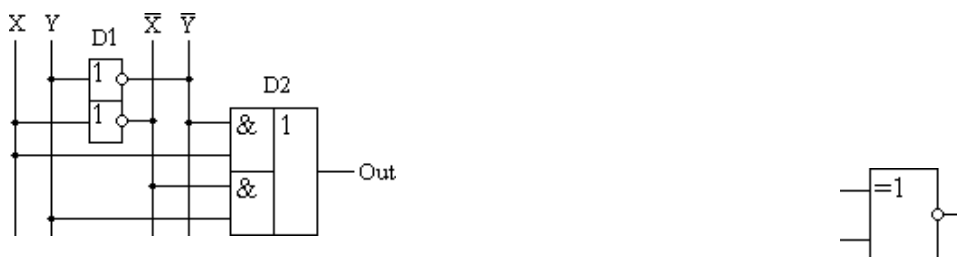


Рис. 1. Сумматор по модулю два:

- 1) расширенная схема,
- 2) «исключающее ИЛИ».

2. Полу сумматоры

Таблица истинности полу сумматора приведена ниже.

Таблица 2.
Таблица истинности полу сумматора

Входы		Выходы	
A	B	S	PO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

На рис. 2 представлена соответствующая схемная реализация полу сумматора на базе логических элементов (рис. 2.1) и в виде одного устройства (рис. 2.2).

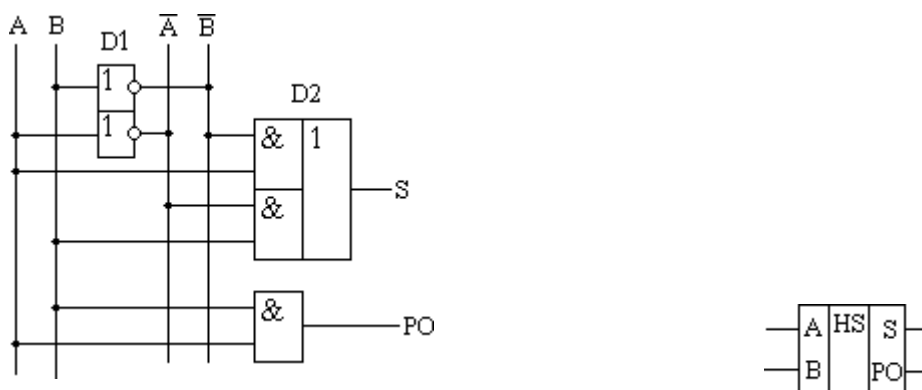


Рис. 2. Схема полу сумматора:

1) расширенная схема,

2) условное обозначение.

Схема полусумматора формирует перенос в следующий разряд (PO), но не может учитывать перенос из предыдущего разряда, поэтому она и называется полусумматором. Для реализации же полного суммирования (пусть пока и одноразрядного) необходимо, помимо формирования переноса в следующий разряд, учитывать еще и перенос из предыдущего разряда (это нужно для формирования многоразрядных сумматоров).

3. Одноразрядные сумматоры

Таблица истинности полного двоичного одноразрядного сумматора приведена в табл. 3.

Таблица 3.
Таблица истинности одноразрядного сумматора

Входы			Выходы	
PI	A	B	S	PO
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Здесь, помимо формирования переноса в следующий разряд (PO), учитывается еще и перенос из предыдущего разряда (PI).

На рис. 3 представлена соответствующая схемная реализация сумматора на базе логических элементов (рис. 3.1) и в виде одного устройства (рис. 3.2).

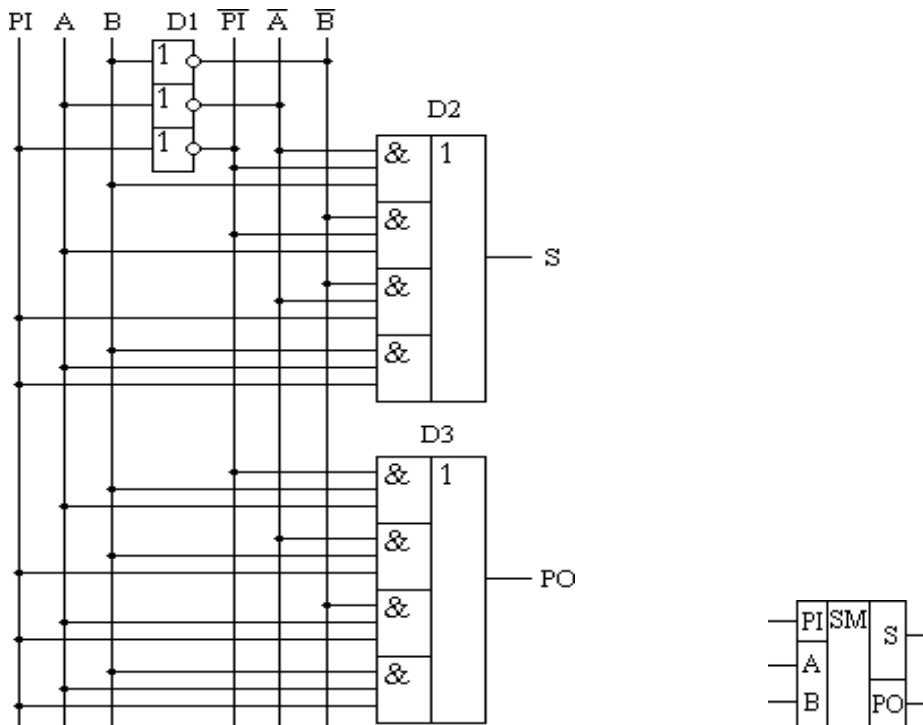


Рис. 3. Схема двоичного одноразрядного сумматора:

- 1) расширенная схема, 2) условное обозначение.

4. Многоразрядные сумматоры

Чтобы получить многоразрядный сумматор, необходимо соединить входы и выходы переносов соответствующих двоичных разрядов. Схема соединения для трехразрядного сумматора показана на рис. 4.1. Здесь же приведено условное обозначение данного сумматора, применяемого на схемах (рис. 4.2).

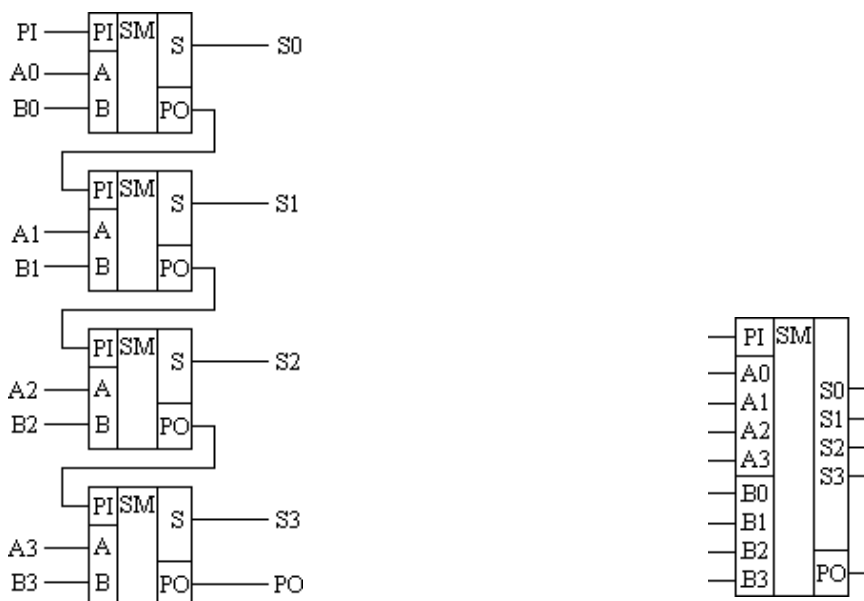


Рис. 4. Схема полного двоичного трехразрядного сумматора:

1) расширенная схема,

2) условное обозначение.

На схеме рис. 4.1 представлен принцип построения двоичного последовательного сумматора. Данной схеме присущ один недостаток – невысокое быстродействие. В реальных же схемах для увеличения скорости работы применяется отдельная схема формирования переносов для каждого двоичного разряда.

Таблицу истинности для такой схемы легко получить, следуя правилам суммирования двоичных чисел, а затем применить хорошо известные принципы построения схемы по произвольной таблице истинности.

Счетчики

Счетчик – это устройство, предназначенное для подсчета числа входных импульсов. Число, представляющее состояние его выходов, с приходом нового импульса изменяется на единицу. Счетчик можно реализовать на нескольких триггерах. В суммирующих счетчиках каждый входной импульс увеличивает число на его выходе на единицу, в вычитающих счетчиках каждый входной импульс уменьшает это число на единицу. Наиболее простые счетчики – двоичные. На рис. 5 представлен суммирующий двоичный счетчик. Диаграмма работы двоичного суммирующего счетчика представлена на рис. 6.

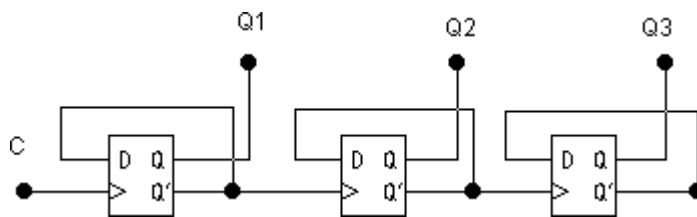


Рис. 5. Суммирующий двоичный счетчик.

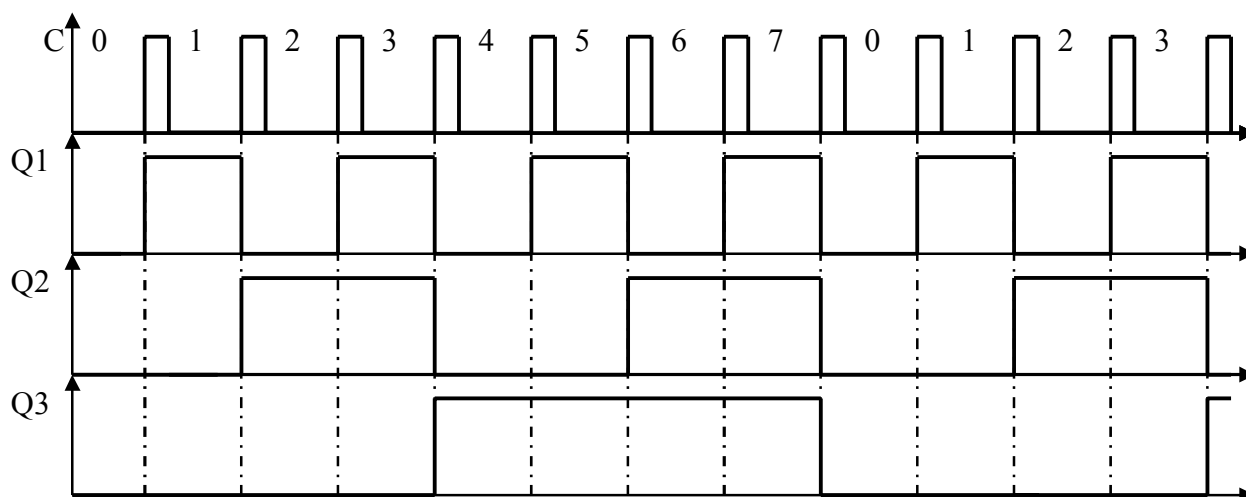


Рис. 6. Диаграмма работы двоичного суммирующего счетчика.

1. Изменение направления счета

Как уже говорилось, счетчики можно реализовать на триггерах. При этом триггеры соединяют последовательно. Выход каждого триггера непосредственно действует на тактовый вход следующего. Чтобы реализовать суммирующий счетчик, необходимо счетный вход очередного триггера подключать к инверсному выходу предыдущего. А чтобы изменить направление счета (реализовать вычитающий счетчик), можно предложить несколько способы.

1. Считывать выходные сигналы счетчика не с прямых, а с инверсных выходов триггеров. Число, образуемое состоянием инверсных выходов триггеров счетчика, связано с числом, образованным состоянием прямых выходов триггеров, соотношением:

$$N_{пр} = 2^n - N_{инв} - 1, \quad (2)$$

где n – разрядность выхода счетчика. В табл. 4 приведен пример связи числа на прямых выходах с числом на инверсных выходах триггеров счетчика.

Таблица 4.

Связь между прямыми и инверсными выходами счетчика

Состояние прямых выходов			Число	Состояние инверсных выходов			Число
Q3	Q2	Q1	N	$\overline{Q3}$	$\overline{Q2}$	$\overline{Q1}$	N
0	0	0	0	1	1	1	7
0	0	1	1	1	1	0	6
0	1	0	2	1	0	1	5

2. Изменить структуру связей в счетчике: подавать на счетный вход следующего триггера сигнал не с инверсного, а с прямого выхода предыдущего, как показано на рис. 7. Временная диаграмма для такого способа реализации счетчика приведена на рис. 8. В этом случае изменяется последовательность переключения триггеров.

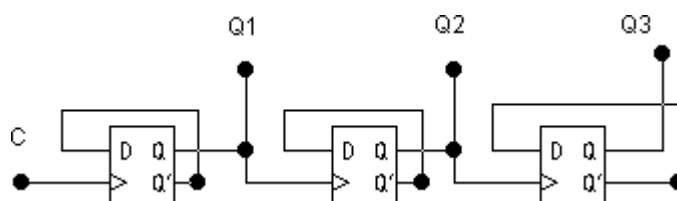


Рис. 7. Вычитающий счетчик.

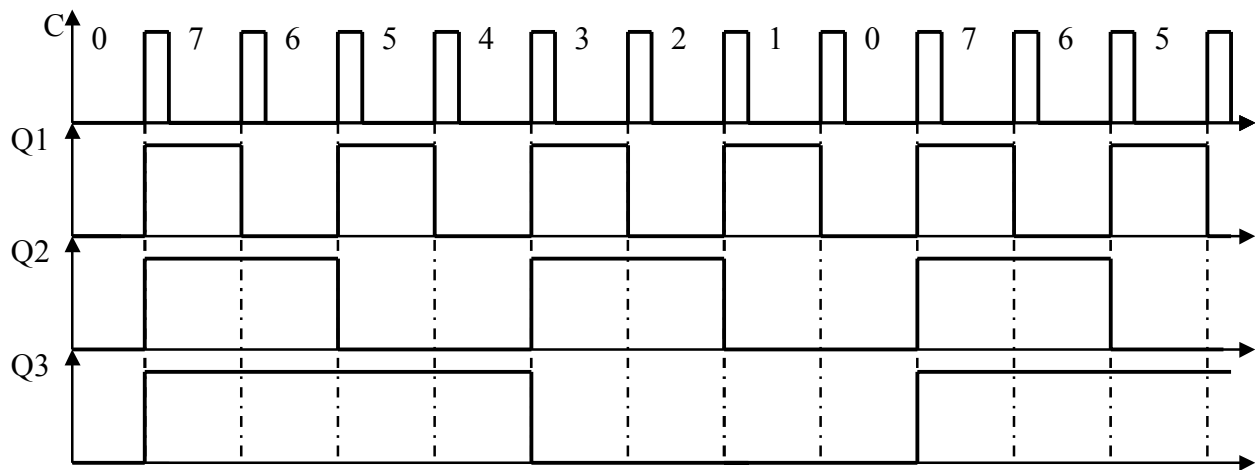


Рис. 8. Диаграмма работы вычитающего счетчика.

2. Изменение коэффициента пересчета

Счетчики характеризуются числом состояний в течение одного периода (цикла). Для схем на рис. 5 и рис. 7 цикл содержит $N=2^3=8$ состояний (от «000» до «111»). Часто число состояний называют *коэффициентом пересчета* $K_{сч}$, который равен отношению числа импульсов N_c на входе к числу импульсов $N_{Qст}$ на выходе старшего разряда за период:

$$K_{сч} = \frac{N_c}{N_{Qст}}. \quad (3)$$

Если на вход счетчика подавать периодическую последовательность импульсов с частотой f_c , то частота f_Q на выходе старшего, разряда счетчика будет меньше в $K_{сч}$ раз:

$$K_{сч} = \frac{F_c}{F_Q}. \quad (4)$$

Поэтому счетчики также называют делителями частоты, а величину $K_{сч}$ – коэффициентом деления. Для увеличения величины $K_{сч}$ приходится увеличивать число триггеров в цепочке. Каждый дополнительный триггер удваивает число состояний счетчика и число $K_{сч}$. Для уменьшения коэффициента $K_{сч}$ можно в качестве выхода счетчика рассматривать выходы триггеров промежуточных каскадов. Например, для счетчика на трех триггерах $K_{сч}=8$, если взять выход 1 и 2-го триггера, $K_{сч}=4$. При этом $K_{сч}$ является целой степенью числа 2: 2, 4, 8, 16 и т. д.

Можно реализовать счетчик, для которого $K_{сч}$ – любое целое число. Например, для счетчика на трех триггерах можно сделать $K_{сч}$ от «2» до «7», но при этом один или два триггера могут оказаться лишними. При использовании всех трех триггеров можно получить $K_{сч}=5...7$:

$$2^2 < K_{сч} < 2^3 . \quad (5)$$

Счетчик с $K_{сч}=5$ должен иметь 5 состояний, которые в простейшем случае образуют последовательность: $\{0, 1, 2, 3, 4\}$. Циклическое повторение этой последовательности означает, что коэффициент деления счетчика равен «5».

Для построения суммирующего счетчика с $K_{сч}=5$ надо, чтобы после формирования последнего числа из последовательности $\{0, 1, 2, 3, 4\}$ счетчик переходил не к числу «5», а к числу «0». В двоичном коде это означает, что от числа «100» нужно перейти к числу «000», а не «101». Изменение естественного порядка счета возможно при введении дополнительных связей между триггерами счетчика. Можно воспользоваться следующим способом: как только счетчик попадает в нерабочее состояние (в данном случае «101»), этот факт должен быть опознан и повлечь последующую выработку сигнала, который перевел бы счетчик в состояние «000». Рассмотрим этот способ более детально.

Факт попадания счетчика в нерабочее состояние описывается логическим уравнением:

$$\begin{aligned} F &= (101) \vee (110) \vee (111) = Q_3 \cdot \bar{Q}_2 \cdot Q_1 \vee Q_3 \cdot Q_2 \cdot \bar{Q}_1 \vee Q_3 \cdot Q_2 \cdot Q_1 = \\ &= Q_3 \cdot Q_1 \vee Q_3 \cdot Q_2 \end{aligned} \quad (6)$$

Состояния «110» и «111» также являются нерабочими и поэтому учтены

при составлении уравнения. Если на выходе эквивалентной логической схемы $F=0$, – значит, счетчик находится в одном из рабочих состояний: «0» или «1» или «2» или «3» или «4». Как только он попадает в одно из нерабочих состояний – «5», «6» или «7», – формируется сигнал $F=1$. Появление сигнала $F=1$ должно переводить счетчик в начальное состояние «000». Следовательно, этот сигнал нужно использовать для воздействия на установочные входы триггеров счетчика, которые осуществляли бы сброс в состояние $Q_1 = Q_2 = Q_3 = 0$. При

реализации счетчика на триггерах с входами установки логическим нулем для сброса триггеров требуется подать на входы сброса сигнал $R'=0$, следовательно, логическую функцию F необходимо инвертировать. Для обнаружения факта попадания в нерабочее состояние применим схему, реализующую функцию F и выполненную на элементах И-НЕ. Для этого преобразуем выражение для функции:

$$\overline{F} = \overline{Q_3 \cdot Q_1 \vee Q_3 \cdot Q_2} = \overline{Q_3 \cdot (Q_1 \cdot Q_2)}. \quad (7)$$

Счетчик будет работать следующим образом: при счете от «0» до «4» все происходит как в обычном суммирующем счетчике с $K_{сч}=8$. Установочные сигналы равны «1» и естественному порядку счета не препятствуют. Счет происходит по положительному фронту импульса на счетном входе C . В тот момент, когда счетчик находится в состоянии «4» («100»), следующий тактовый импульс сначала переводит его в состояние «5» («101»), что немедленно (задолго до прихода следующего тактового импульса) приводит к формированию сигнала сброса, который поступает на установочный вход R' триггеров. В результате счетчик сбрасывается в «0» и ждет прихода следующего тактового импульса на счетный вход. Один цикл счета закончился, счетчик готов к началу следующего цикла.

Применяя такие схемы с обратной связью для сброса счетчика, нужно иметь в виду, что операция сброса занимает конечное время, поэтому непосредственно перед сбросом счетчика в «0» на выходе первого триггера появляются кратковременные импульсы, или «иголки». Это не имеет значения при подключении счетчика напрямую к индикатору, но при использовании этого выхода счетчика в качестве источника тактовых импульсов могут возникнуть определенные проблемы.

Важным отличием является то, что схема обнаруживает не факт попадания в нерабочее состояние «101», а факт попадания в состояние «100» и в следующем такте вырабатывает сигнал сброса.

Регистры

Регистром называется последовательное или параллельное соединение триггеров. Регистры обычно строятся на основе *D-триггеров*. При этом для их построения могут использоваться как универсальные *D-триггеры*, так и триггеры-защелки.

1. Параллельный регистр

Параллельный регистр служит для запоминания многоразрядного двоичного слова. При использовании для построения параллельного регистра *триггеров-защелок* регистр называется *регистр-защелка*. Количество триггеров, входящее в состав параллельного регистра, определяет его разрядность. При записи информации в параллельный регистр все биты (двоичные разряды) записываются одновременно..

2. Последовательный регистр

Последовательный регистр (регистр сдвига) обычно служит для преобразования последовательного кода в параллельный и наоборот. Схема регистра, осуществляющего преобразование последовательного кода в параллельный, приведена на рис. 12.1, а его изображение на принципиальных схемах – на рис. 12.2.

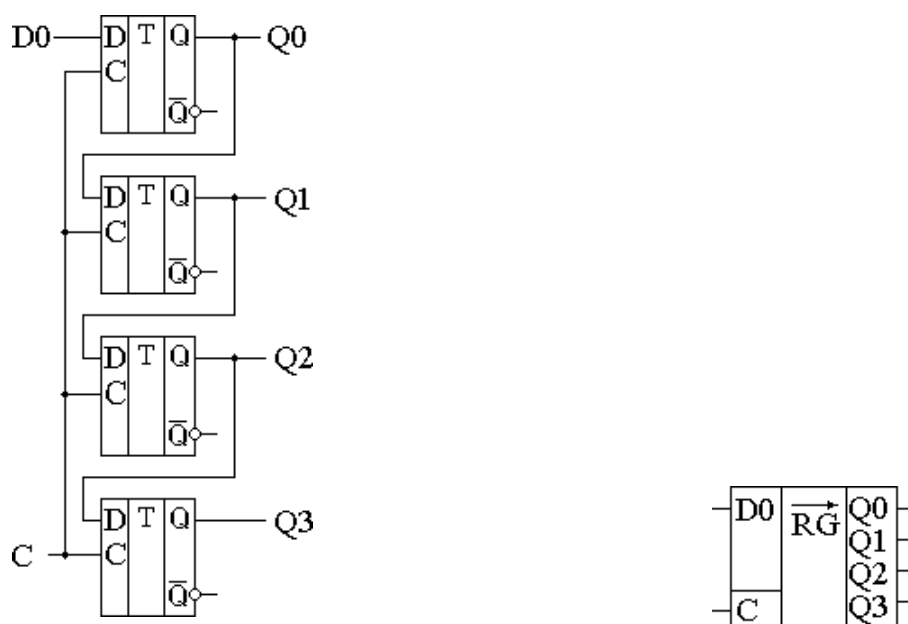


Рис. 12. Схема последовательного регистра:

1) расширенная схема,

2) условное обозначение.

Регистры сдвига выполняются обычно как универсальные *последовательно-параллельные микросхемы*. Переключение регистра из параллельного режима в последовательный и наоборот осуществляется при помощи мультиплексора. Схема такого регистра приведена на рис. 13.1, а его изображение на принципиальных схемах – на рис. 13.2.

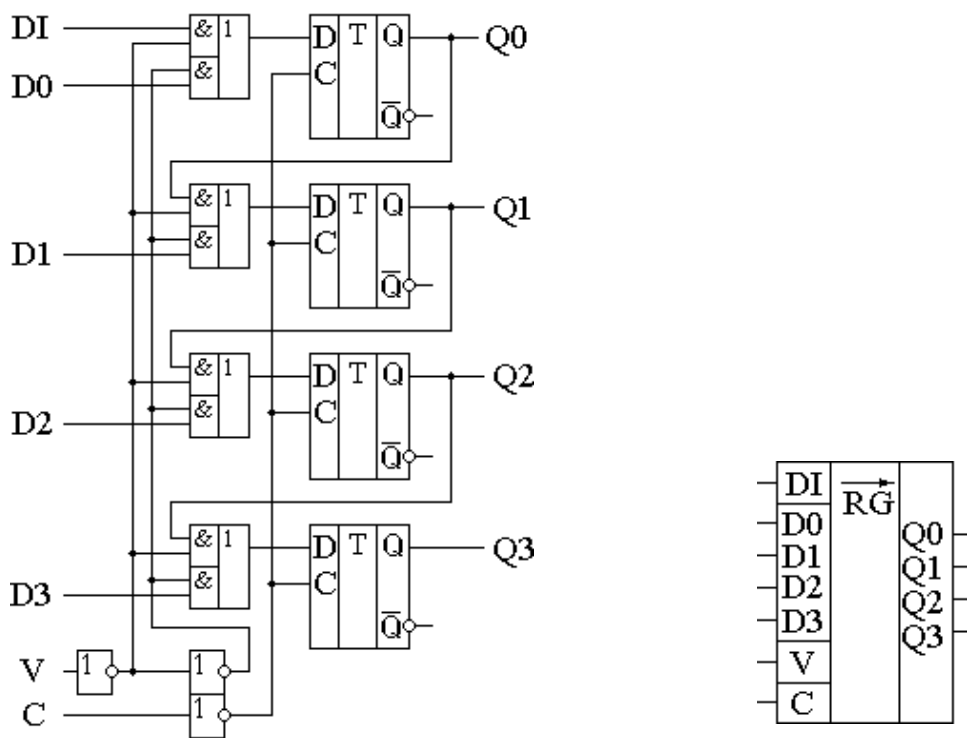


Рис. 13. Схема универсального регистра:

1) расширенная схема,

2) условное обозначение.

Порядок работы

Задание 1. Изучение полусумматоров и сумматоров

Изучение работы полусумматора.

Соберите схему, приведенную на рис. 14.

Задавая различные комбинации логических уровней на входах полусумматора *A* и *B*, заполните таблицу истинности (табл. 5). Сравните полученную таблицу с табл. 1, сделайте выводы.

Таблица 5.

Таблица истинности полусумматора

Входы		Выходы	
A	B	S	PO
0	0		
0	1		
1	0		
1	1		

Изучение работы одноразрядного сумматора.

Задавая различные комбинации логических уровней на входах сумматора A и B , а также уровень сигнала переноса из предыдущего разряда PI , заполните таблицу истинности (табл. 6).

Сравните полученную табл. 6 с табл. 2, сделайте выводы.

Таблица 6.

Таблица истинности одноразрядного сумматора

Входы			Выходы	
PI	A	B	S	PO
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Изучение работы двоичного трехразрядного сумматора.

Соберите схему, аналогичную рис. 16. Здесь в качестве сумматоров используются специальные блоки EWB

«Half-Adder», имеющие аналогично условному обозначению рис. 3.2 два входа A и B , выход сума «S» (в нашем случае – « Σ ») и сигнал переноса в следующий разряд «PO» (в нашем случае – « C_0 »). В отличие от сумматора, приведенного на рис. 3.2 у данного сумматора нет учета переноса из предыдущего разряда. Для формирования переноса в следующий разряд в схеме рис. 16 используются дополнительные сумматоры и логические элементы.

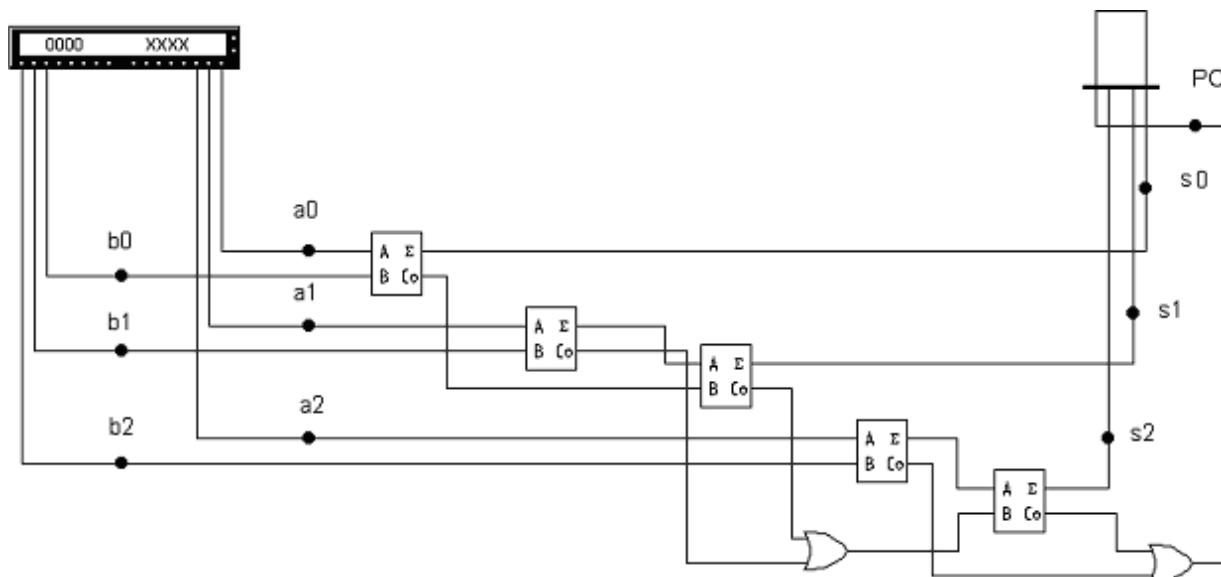


Рис. 16. Схема трехразрядного двоичного сумматора.

Для проверки правильности функционирования данной схемы необходимо предварительно запрограммировать генератор слов так, чтобы на его используемых выходах формировались все возможные комбинации суммируемых двоичных числе «a2 a1 a0» и «b2 b1 b0». Ниже приведена таблица, поясняющая принцип программирования генератора слов (табл. 7).

Таким образом, из табл. 7 видно, что при подключении к трем младшим разрядам генератора слов одного двоичного числа «a2 a1 a0», а к трем старшим разрядам другого двоичного числа «b2 b1 b0» можно просуммировать все возможные комбинации двух-, трехразрядных чисел.

Примечание. Генератор слов удобнее программировать в 16-теричном коде (см. лабораторную работу №1); семисегментный дисплей («Decoded Seven-Segment Display») сразу декодирует двоичное число (переводит его в 16-теричное) «s2 s1 s0» и отображает его в удобном для анализа виде.

Запрограммировав генератор слов, необходимо изучить работу трехразрядного двоичного сумматора, заполнив табл. 8.

Таблица 7.
Программирование генератора слов

16-теричный код	Двоичный код
0000	0000 0000
0001	0000 0001
0002	0000 0010
0003	0000 0011
0004	0000 0100
0005	0000 0101
0006	0000 0110
0007	0000 0111
2000	0010 0000
2001	0010 0001
2002	0010 0010
2003	0010 0011
2004	0010 0100
2005	0010 0101
2006	0010 0110
2007	0010 0111
4000	0100 0000
...	...
4007	0100 0111
6000	0110 0000
...	...

1	1	1	1	0	1							
0	0	0	1	1	0							
...
1	1	1	1	1	0							
0	0	0	1	1	1							
...
1	1	1	1	1	1							

**Задание 2. Исследование работы суммирующего
и вычитающего счетчиков**

Исследование первой реализации счетчика.

Включите схему. Подавая на вход схемы тактовые импульсы при помощи ключа *C* и наблюдая состояние выходов счетчика при помощи семисегментного дисплея декодера и логических пробников, составьте временные диаграммы работы суммирующего (вычитающего) счетчика. Сделайте выводы о работе схемы.

Примечание. В данном случае семисегментный дисплей предназначен для автоматического перевода поступающего на его вход двоичного числа в шестнадцатеричное и отображение последнего.

**Задание 3. Исследование счетчика с измененным
коэффициентом пересчета**

Первый вариант.

Включите схему. Подавая на вход схемы тактовые импульсы при помощи ключа *C* и наблюдая состояние выходов счетчика при помощи логических пробников, составьте временные диаграммы работы счетчика и определите коэффициент пересчета.

Второй вариант.

Соберите схему, изображенную на рис. 19.

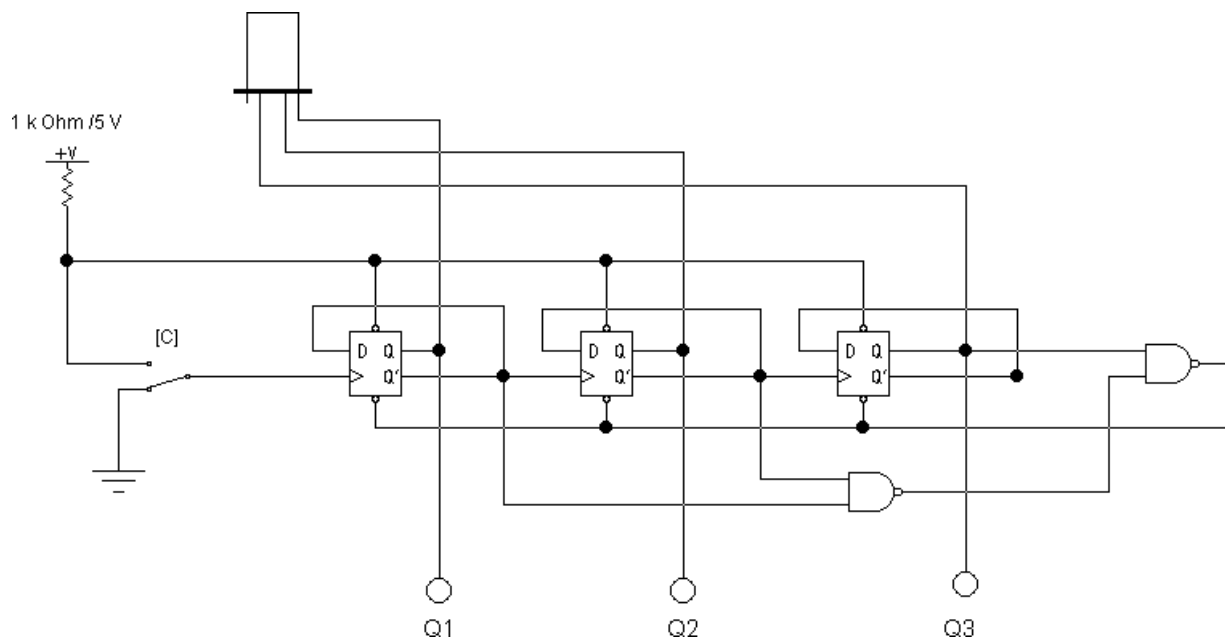


Рис. 19. Счетчик с измененным коэффициентом пересчета.

Включите схему. Подавая на вход схемы тактовые импульсы при помощи ключа *C* и наблюдая состояние выходов счетчика при помощи логических пробников, составьте временные диаграммы работы счетчика и определите коэффициент пересчета.

Задание 4. Исследование регистров

4.1. Исследование параллельного регистра.

Создайте схему, изображенную на рис. 20.

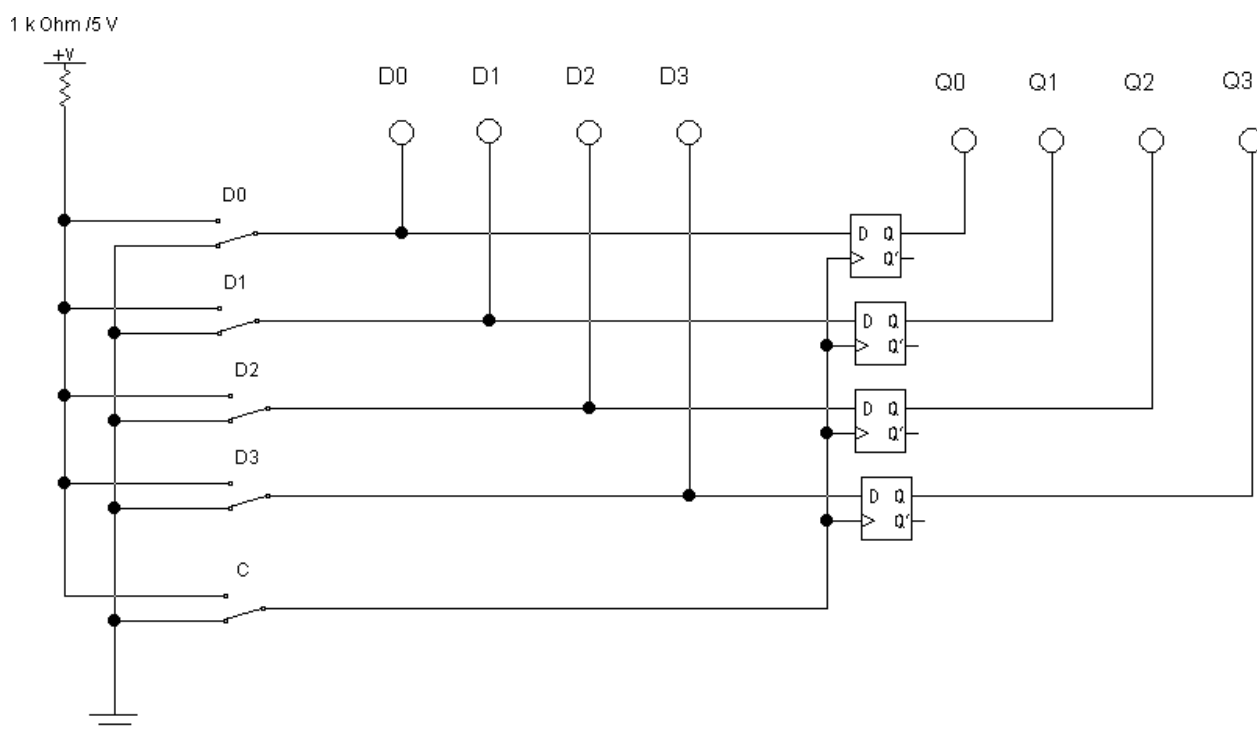


Рис. 20. Схема четырехразрядного параллельного регистра.

Включите схему. Исследуйте работу полученного двоичного четырехразрядного регистра, заполнив таблицу истинности, приведенную ниже (табл. 9).

Примечание. Запись числа в регистр происходит по переднему фронту разрешающего импульса и сохраняется до появления следующего импульса.

Таблица 9.
Параллельный регистр

Входы				Выходы			
D0	D1	D2	D3	Q0	Q1	Q2	Q3
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

4.2. Исследование параллельного регистра.

Создайте схему, изображенную на рис. 21.

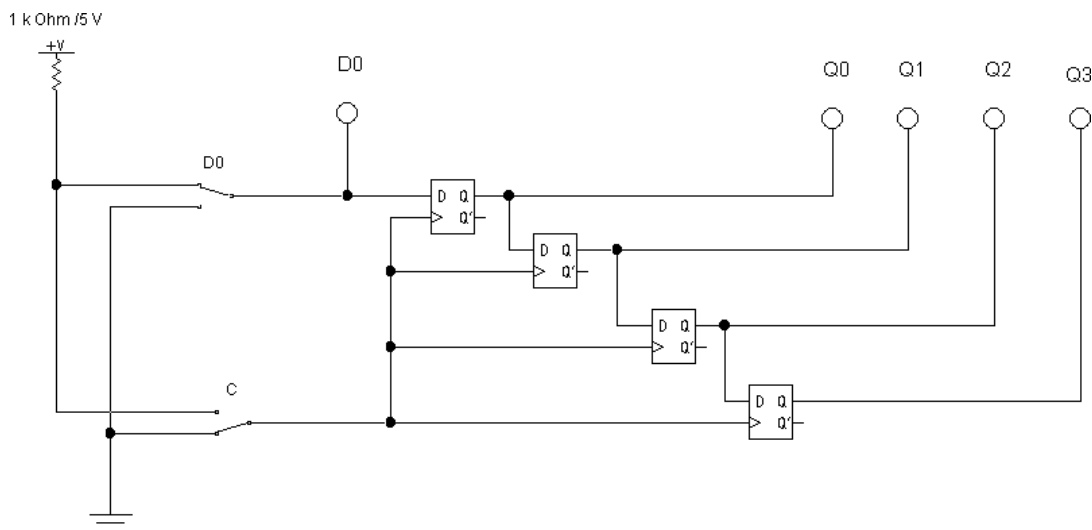


Рис. 21. Схема четырехразрядного последовательного регистра.

Включите схему. Согласно вашему варианту из табл. 10 необходимо вы-

брать двоичное число, которое требуется получить на выходе регистра сдвига (последовательного регистра).

Таблица 10.

Варианты задания числа для последовательного регистра

Вариант	Число			
	Q0	Q1	Q2	Q3
1	0	0	1	1
2	0	1	0	0
3	0	1	0	1
4	0	1	1	0
5	0	1	1	1
6	1	0	0	0
7	1	0	0	1
8	1	0	1	0
9	1	0	1	1
10	1	1	0	0

Задавая необходимые комбинации на входе $D0$ и занося информацию в регистр с помощью сигнала разрешения C , добейтесь получения на выходе $Q0 Q1 Q2 Q3$ требуемого числа.

Постройте временную диаграмму получения в регистре сдвига требуемого числа.

Контрольные вопросы

1. Дайте определение сумматора.
2. Опишите принцип работы сумматоров.
3. Чем отличается полусумматор от сумматора?
4. Как можно составить схему четырехразрядного сумматора в EWB?
5. Приведите примеры применения сумматоров.
6. На основе каких видов триггеров можно получить схему счетчика?
7. Как из суммирующего счетчика получить вычитающий?
8. Что такое коэффициент пересчета счетчика?
9. Как можно вычислить коэффициент пересчета счетчика?
10. Как можно изменить коэффициент пересчета счетчика?
11. Дайте определение регистра.
12. Чем отличается последовательный регистр от параллельного?
13. Приведите примеры использования параллельных регистров.
14. Приведите примеры использования последовательных регистров.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Янсен Й. Курс цифровой электроники: В 4-х т. – Т. 2. Проектирование устройств на цифровых ИС /пер. с голланд. – М.: Мир, 1987.
2. Блейкли Т.Р. Проектирование цифровых устройств с малыми и большими интегральными схемами /пер. с англ. – К.: Выща школа, 1981.
3. Трачик В. Дискретные устройства автоматики /пер. с польск., под ред. Д.А. Поспелова. – М.: Энергия, 1978.
4. Аппаратные средства макетирования узлов и устройств ЭВМ /Ковригин Б.Н., Сидуков В.М., Мифтахов Р.К., Тышкевич В.Г., Иванов М.А.; под ред. Б.Н. Ковригина: учебное пособие. – М.: МИФИ, 1991.
5. Ковригин Б.Н. Триггерные схемы. – Ч. 1. Описание и классификация. – М.: МИФИ, 1976.
6. Ковригин Б.Н. Триггерные схемы. – Ч. 2. Синтез и анализ. – М.: МИФИ, 1977.
7. Миллер Р. Теория переключательных схем. В 2-х т. – Т. 1. Комбинационные схемы /пер. с англ.; под ред. П.П. Пархоменко. – М.: Наука, 1970.
8. Голдсуорт Б. Проектирование цифровых логических устройств /пер. с англ.; под ред. Ю.И.Толчеева. – М.: Машиностроение, 1985.
9. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П. Цифровые ЭВМ: Теория и проектирование /под общ. ред. К.Г. Самофалова. – 3-е изд., перераб. и доп. – К.: Выща школа, 1989.
10. Антонию А. Цифровые фильтры: анализ и проектирование /пер. с англ. – М.: Радио и связь, 1983.
11. Электротехника и электроника в экспериментах и упражнениях. Практикум на Electronics Workbench /под общ. ред. Д. И. Панфилова. – Т. 2 – Электроника. – М.: Додека, 2000.
12. Лачин В.И., Савелов Н.С.. Электроника: учеб. пособие. – Ростов н/Д: Феникс, 2004.
13. Прянишников В.А. Электроника. Полный курс лекций. – СПб.: Корона принт, 2004.