

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

*А. С. Кобайло, Н. А. Жиляк,
И. А. Миронов*

СХЕМОТЕХНИКА ЦИФРОВЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

*Рекомендовано
учебно-методическим объединением
по образованию в области информатики и радиоэлектроники
в качестве учебно-методического пособия по учебной дисциплине
«Арифметико-логические основы цифровых вычислительных
машин и архитектура компьютеров» для студентов
специальности 1-98 01 03 «Программное обеспечение
информационной безопасности мобильных систем»*

Минск 2018

УДК [004.4+004.7](076.5)(075.8)
ББК 32.97я73
К55

Р е ц е н з е н т ы :
кафедра экономической информатики
УО «Белорусский государственный университет
информатики и радиоэлектроники»
(заведующий кафедрой кандидат технических наук,
доцент *В. Н. Комличенко*);
профессор кафедры логистики
и информационно-математических дисциплин
ЧУО «БИП-Институт правоведения» кандидат
физико-математических наук, доцент *А. В. Остапенко*

Все права на данное издание защищены. Воспроизведение всей книги или ее части не может быть осуществлено без разрешения учреждения образования «Белорусский государственный технологический университет».

Кобайло, А. С.

К55 Схемотехника цифровых вычислительных машин. Лабораторный практикум : учеб.-метод. пособие по учебной дисциплине «Арифметико-логические основы цифровых вычислительных машин и архитектура компьютеров» для студентов специальности 1-98 01 03 «Программное обеспечение информационной безопасности мобильных систем» / А. С. Кобайло, Н. А. Жилияк, И. А. Миронов. – Минск : БГТУ, 2018. – 97 с.
ISBN 978-985-530-664-2.

Лабораторный практикум содержит основные понятия, теоретические сведения, методические указания, упражнения и задания для проведения лабораторных работ по курсу «Арифметико-логические основы цифровых вычислительных машин и архитектура компьютеров».

Предназначен для студентов специальности 1-98 01 03 «Программное обеспечение информационной безопасности мобильных систем», а также может быть полезен студентам, изучающим арифметические и логические основы цифровых вычислительных машин и архитектуру компьютеров.

УДК [004.4+004.7](076.5)(075.8)
ББК 32.97я73

ISBN 978-985-530-664-2

© УО «Белорусский государственный
технологический университет, 2018
© Кобайло А. С., Жилияк Н. А.,
Миронов И. А., 2018

ПРЕДИСЛОВИЕ

На начальной стадии подготовки высококвалифицированного инженера-программиста, специализирующегося в области разработки и использования современных информационных технологий и систем, необходимо предусмотреть изучение следующих областей знаний: арифметических основ вычислительной техники на базе двоичной арифметики, логических основ вычислительной техники на базе изучения алгебры логики, схмотехнических основ и архитектурной организации ЭВМ и вычислительных систем.

Для овладения данными областями знаний и получения практических навыков необходимо использовать специализированные программные продукты.

На этапе начального освоения студентами моделирования электронных схем и изучения архитектуры компьютера на цифровом логическом уровне наиболее приемлемым средством считается программная среда NI Multisim, разработанная группой Electronics Workbench (входящей в корпорацию National Instruments).

Лабораторный практикум по дисциплине «Компьютерные системы и сети» разработан на базе программной среды Multisim.

В пособии рассмотрены основные элементы алгебры логики, базовые логические элементы, представление логических функций математическими выражениями, переход от логической функции к логической схеме. Практикум включает исследования запоминающих элементов (на примере триггеров *RS*, *D*, *T* и *JK*), преобразователей кодов, коммутационных узлов, организации и функционирования запоминающих узлов и других типовых элементов и узлов цифровой вычислительной техники.

Использование в учебном процессе разработанного цикла лабораторных работ на базе инструментального средства Multisim позволяет закрепить теоретические знания по арифметическим и логическим основам ЦВТ, приобрести практические навыки проектирования цифровых компонентов вычислительной техники, освоить основы компьютерного моделирования архитектурных составляющих цифрового логического уровня компьютерных систем. Каждая лабораторная работа в пособии содержит методические указания по ее выполнению.

**ИССЛЕДОВАНИЕ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ
В СРЕДЕ СХЕМОТЕХНИЧЕСКОГО
МОДЕЛИРОВАНИЯ MULTISIM**

***1.1. Среда схемотехнического
моделирования Multisim***

Наряду с физическими (реальными) экспериментами в настоящее время широкое распространение получило компьютерное проектирование и анализ цифровых устройств в среде схемотехнического моделирования NI Multisim. Особенностью программной среды Multisim является наличие в ее библиотеке более 16 000 электронных компонентов, а также виртуальных контрольно-измерительных приборов, которые по характеристикам приближены к их промышленным аналогам.

Multisim состоит из редактора схем и подсистемы моделирования, базирующейся на интеграции вычислительных ядер SPICE3F5 (BSpice) и XSpice. Пакет MCU позволяет включать в эмуляцию смешанной схемы определенные микроконтроллеры.

Программа Multisim имитирует реальное рабочее место в исследовательской лаборатории, которое оборудовано измерительными приборами: генераторами, мультиметрами, осциллографами, анализатором спектра, измерителем АЧХ и ФЧХ, измерителем нелинейных искажений, преобразователем и анализатором логических сигналов и др.

Multisim – программа с многооконным графическим интерфейсом, позволяющим строить и редактировать схемы, модели и изображения компонентов, а также представлять результаты расчетов в удобном графическом виде.

Пользовательский интерфейс программы показан на рис. 1.1 и состоит из следующих элементов: строка меню, панель инструментов, панель разработки, окно редактирования, приборная панель.

Панель инструментов Панель разработки Строка меню Окно редактирования приборная панель

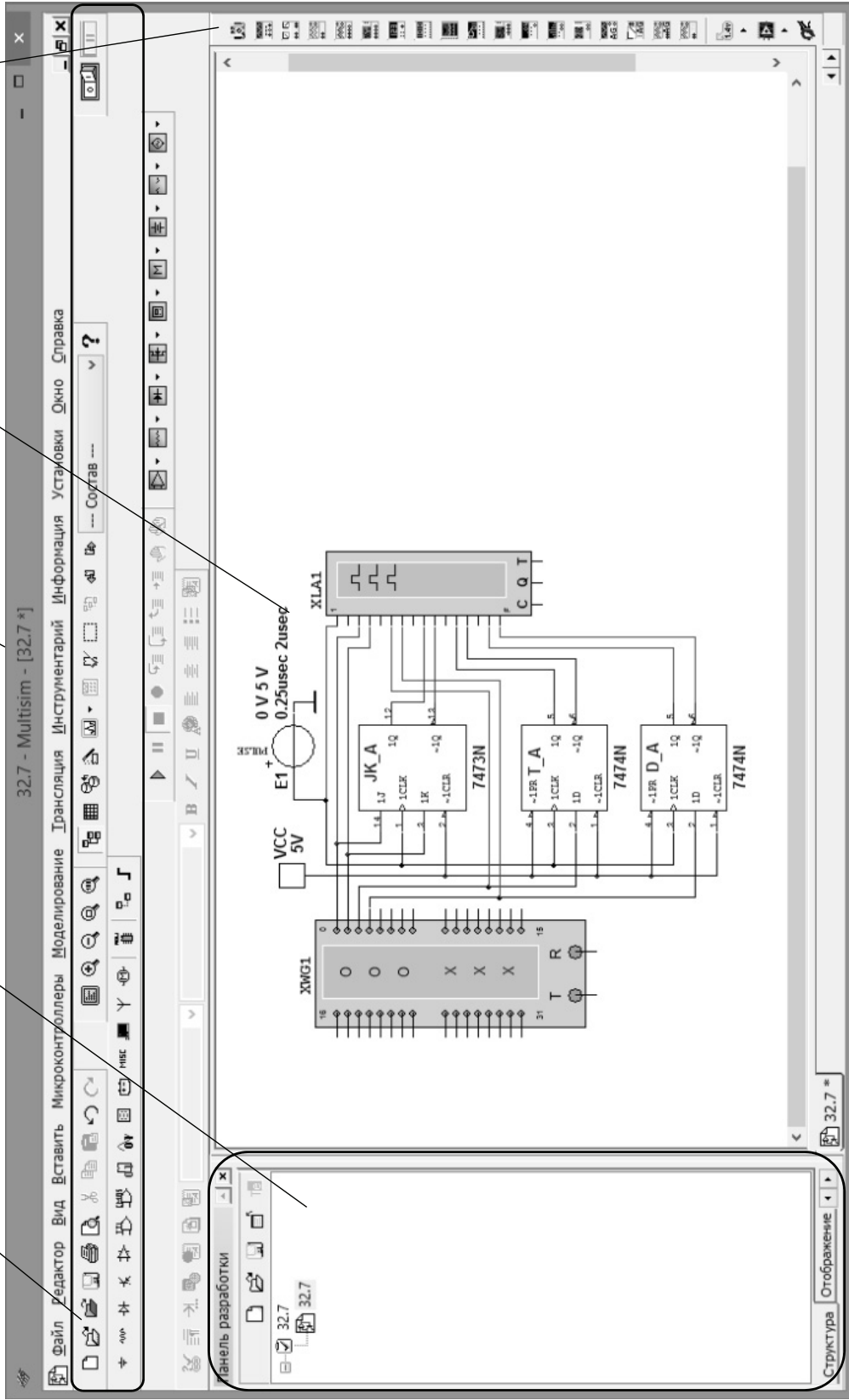


Рис. 1.1. Интерфейс программы Multisim

Пользовательский интерфейс Multisim можно настроить на свой вкус, изменения зависят друг от друга. Панели инструментов можно закрепить в любом месте и изменить их форму. Инструменты всех панелей также можно изменять и создавать новые панели. Система меню полностью настраивается, вплоть до контекстных меню разных объектов.

Рассмотрим некоторые модели контрольно-измерительных приборов, вынесенные на приборную панель

1.1.1. Контрольно-измерительные приборы Multisim

Осциллограф (Oscilloscope) позволяет проводить анализ сигналов во временной области. Внешний вид и лицевая панель осциллографа показаны на рис. 1.2.

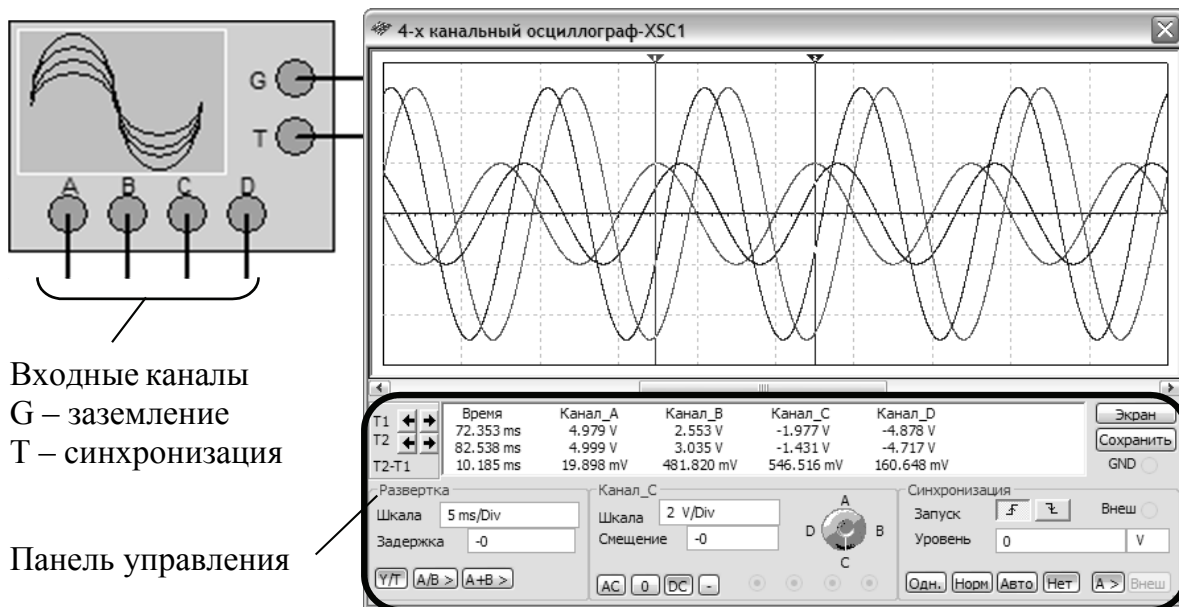


Рис. 1.2. Четырехканальный осциллограф

Модель имеет четыре канала *A*, *B*, *C* и *D* с отдельной регулировкой чувствительности в диапазоне от 10^{-15} В/дел до 10^{15} В/дел, а также регулировкой смещения по горизонтали и вертикали.

Для настройки отображения измеряемого сигнала используется панель управления осциллографа (рис. 1.3). Выбор режима работы по входу осуществляется нажатием клавиш <AC>, <0>, <DC>, <--> (инверсный режим – только для канала *B*). Режим <AC> равносителен введению емкостного фильтра в цепь сигнала, при этом отображается только переменная составляющая сигнала. В режиме <0> входной канал замыкается на землю. В режиме <DC> отображаются обе

составляющие сигнала. В инверсном режиме сигнал инвертируется относительно положения нуля.

Для выбора режима развертки используются кнопки <Y/T>, <A/B>, <A+B>. В режиме <Y/T> реализуется временная развертка для каждого канала, т. е. горизонтальная ось представляет собой ось времени, а сигналы каналов A, B, C, D отображаются по вертикальной оси. Длительность развертки задается в поле Scale (Шкала) параметра Timebase и варьируется в диапазоне от 10^{-15} с/дел до 10^{+15} с/дел. В режиме <A+B> отображается суммарный сигнал по двум выбранным каналам. Для построения передаточной характеристики исследуемой схемы используется режим <A/B>.

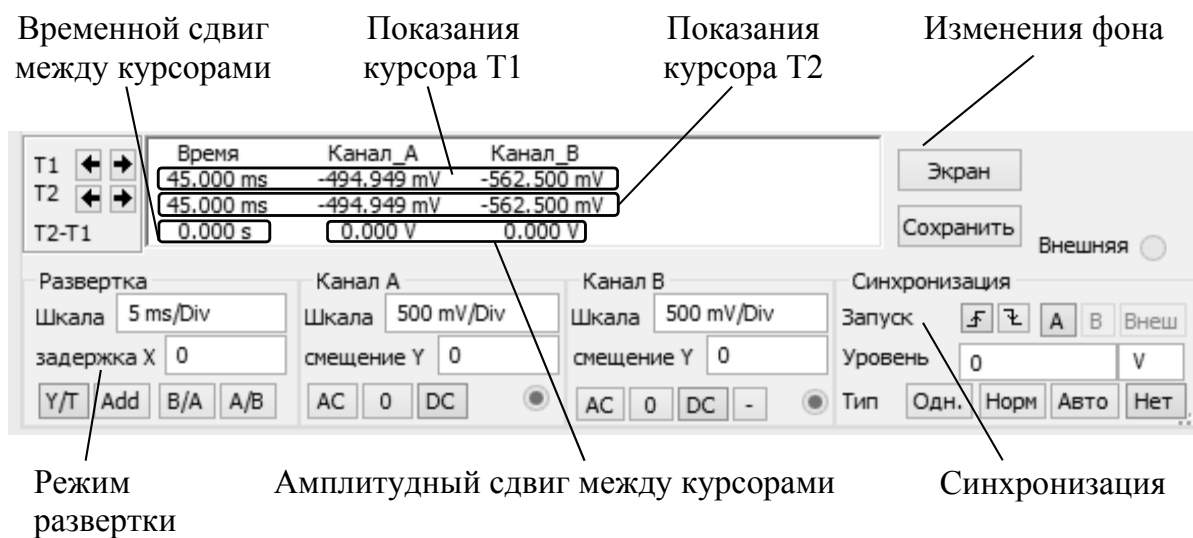


Рис. 1.3. Панель управления осциллографа

Генератор слов (Word Generator) предназначен для генерации до 8192 32-разрядных двоичных слов. Внешний вид и лицевая панель его показаны на рис. 1.4. Ввод генерируемых слов производится в буфере ввода. Формат отображения кодовых слов выбирается с помощью группы кнопок <Display>(<Hex> – шестнадцатеричный; <Dec> – десятичный; <Binary> – двоичный; <ASCII> – символьный код). Частота генерации кодовых слов задается в окне Frequency (Частота) и лежит в диапазоне от 1 Гц до 1000 МГц. В процессе работы на каждом выводе генератора появляется логический уровень согласно разряду двоичного кодового слова, при этом генератор работает в трех режимах:

- Step (Пошаговый) – каждый раз при подаче очередного слова на выход моделирование останавливается;

- Burst (Пакетный) – генерируется последовательность кодовых слов, начиная с начальной ▾ позиции и заканчивая конечной позицией, моделирование останавливается при достижении конечной позиции;

- Cycle (Циклический) – на выводах генератора последовательно появляются логические уровни согласно комбинации слов, генерирование осуществляется до тех пор, пока не будет остановлено моделирование или достигнута точка прерывания (Breakpoint).

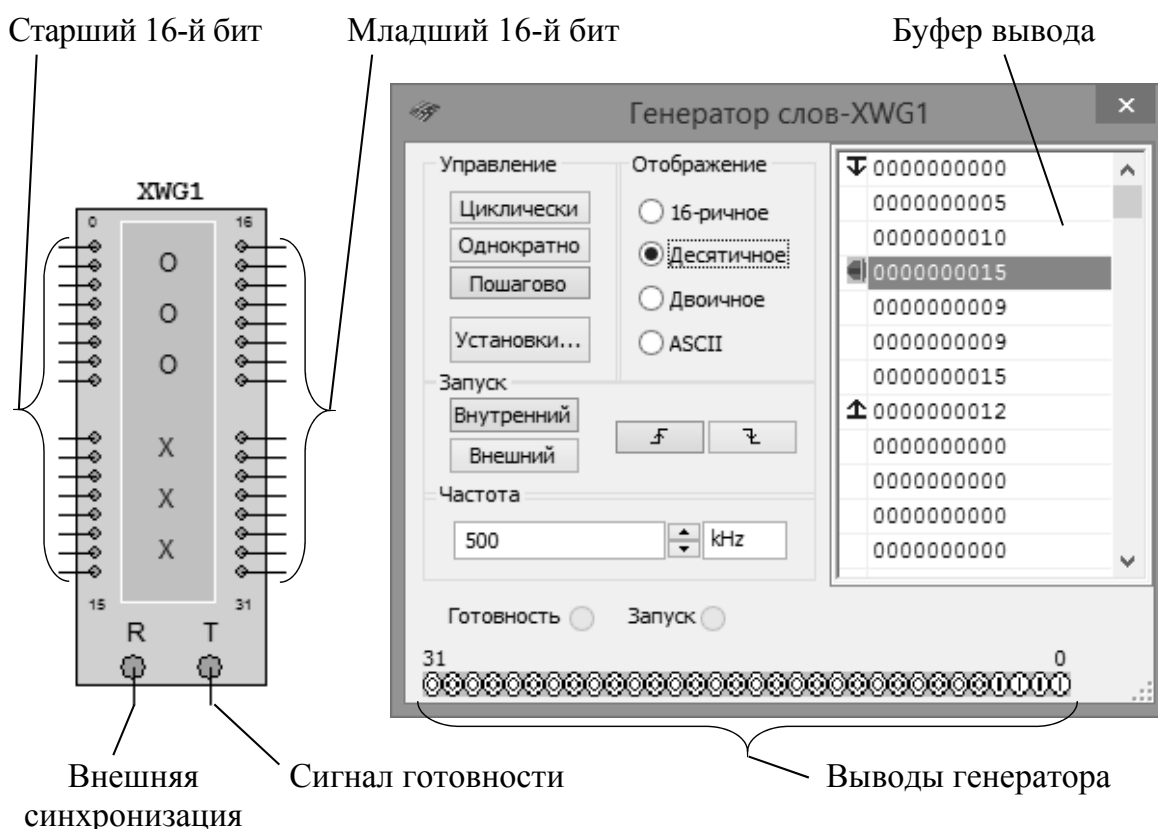


Рис. 1.4. Генератор слов

Во время моделирования курсор ► в окне «Буфер вывода» указывает на текущее генерируемое слово. Остановив моделирование, можно изменить положение курсора, начальную позицию, конечную позицию, а также точку прерывания.

При нажатии кнопки «Установки» открывается диалоговое окно свойств буфера (рис. 1.5):

- Без изменений – оставить без изменений;
- Загрузить – загрузить кодовые слова из файла;
- Сохранить – сохранить кодовые слова в файл;

- Очистить буфер – обнулить содержимое буфера;
- Вверх – заполнить буфер кодовыми словами начиная с кода, указанного в поле Инициализировать конфигурацию (по умолчанию 0×0000), с последующим увеличением на 1 в каждой следующей строке;
- Вниз – заполнить буфер кодовыми словами начиная с кода, указанного в поле Инициализировать конфигурацию (по умолчанию 0×0400), с последующим уменьшением на 1 в каждой следующей строке;
- Вправо – заполнить буфер кодовыми словами начиная с кода, указанного в поле Инициализировать конфигурацию (по умолчанию 0×80000000), с последующим двоичным сдвигом вправо на 1 разряд в каждой следующей строке;
- Влево – заполнить буфер кодовыми словами начиная с кода, указанного в поле Инициализировать конфигурацию (по умолчанию 0×0001), с последующим двоичным сдвигом влево на 1 разряд в каждой следующей строке. Запуск генератора может синхронизироваться как внутренним (Internal), так и внешним (External) сигналом синхронизации. На вывод Ready подается сигнал готовности.

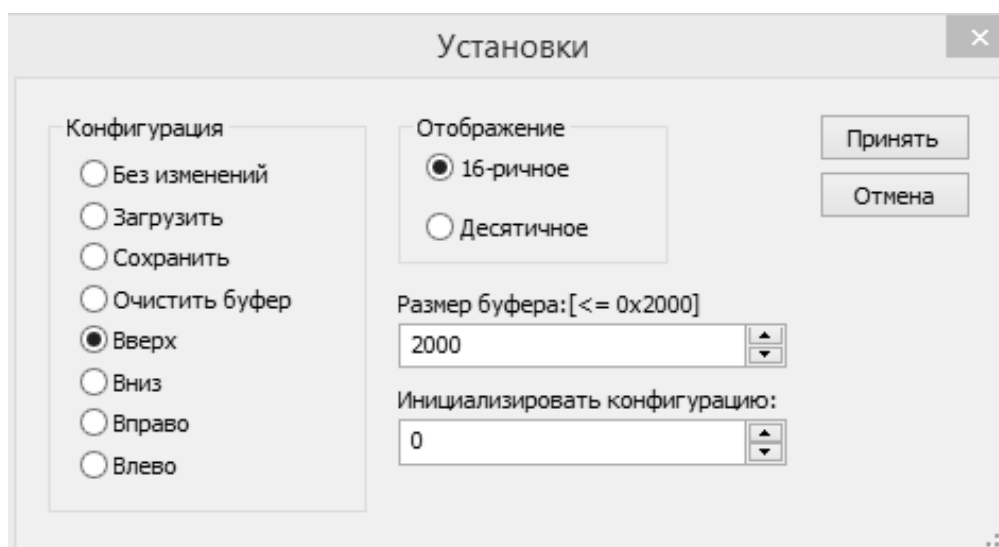


Рис. 1.5. Окно свойств буфера

Логический анализатор (Logic Analyzer). Логический анализатор (ЛА) – устройство, предназначенное для диагностики цифровых схем. ЛА позволяет отслеживать и записывать состояния логических элементов цифровых электронных устройств, анализировать и визуализировать их. Внешний вид и лицевая панель данного анализатора показаны на рис. 1.6.

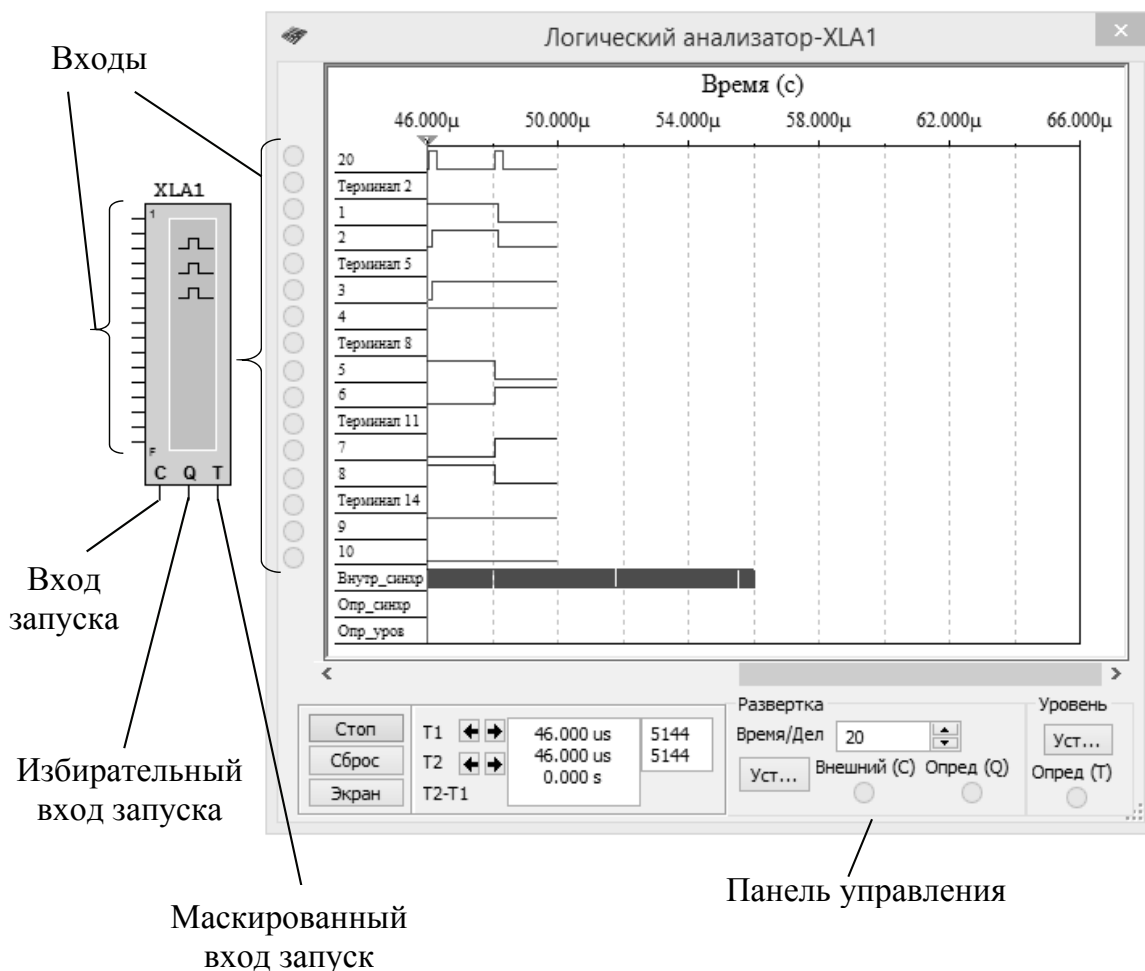


Рис. 1.6. Логический анализатор

ЛА имеет 16 каналов для съема сигналов, а также несколько входов запуска. Кроме этого, прибор снабжен двумя курсорами, позволяющими проводить измерения во временной области.

Если вход 1 считать младшим разрядом, а вход 16 – старшим, то состояние всех входов может быть представлено 16-разрядным двоичным кодом. Код, соответствующий позиции курсора, отображается в поле Входной код (рис. 1.7).

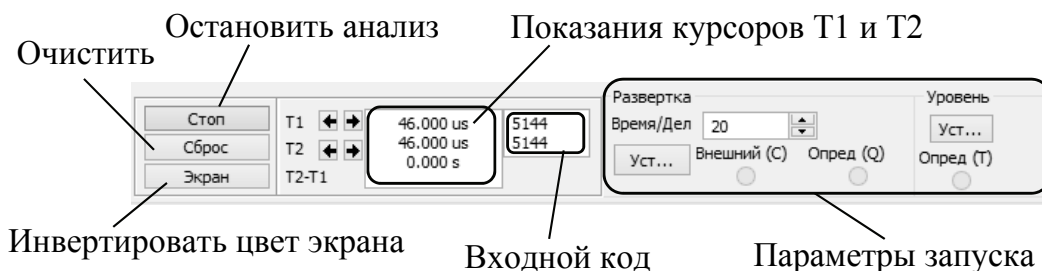


Рис. 1.7. Панель управления логического анализатора

При нажатии кнопки Установки в группе Развертка (тактовый генератор) открывается диалоговое окно настройки параметров тактирования входных сигналов (рис. 1.8).

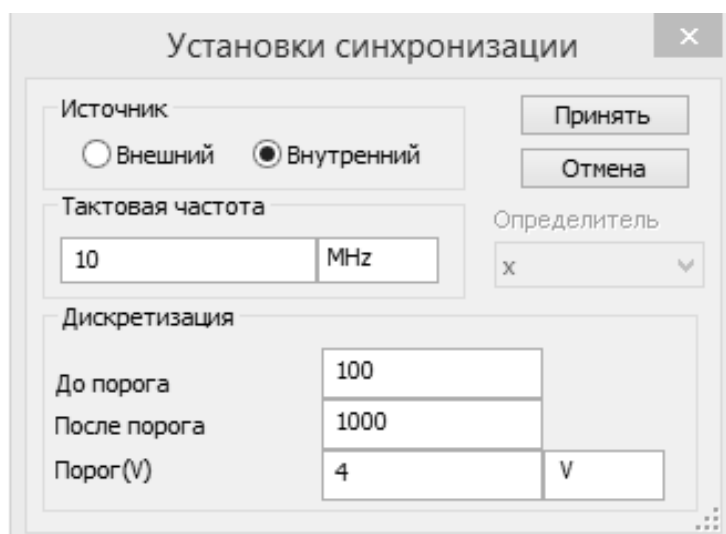


Рис. 1.8. Настройка параметров синхронизации

Тактирование сигналов осуществляется с использованием внешнего (External) или внутреннего (Internal) источников.

В поле Clock Qualifier (Определитель) устанавливается активный уровень сигнала синхронизации. В поле Clock Rate (Тактовая частота) устанавливается частота выборки анализатора.

В группе Sampling Setting (Дискретизация) задаются параметры выборки сигналов:

- Pre-trigger Samples (До порога) – сбор данных производится до поступления импульса запуска;
- Post-trigger Samples (После порога) – сбор данных начинается после поступления импульса запуска и продолжается до тех пор, пока не будет набрано заданное количество отсчетов;
- Threshold Volt (Порог) – пороговая величина.

Дополнительные условия запуска анализатора осуществляются с помощью диалогового окна Trigger Settings.

В данном окне настраивается маска, по которой осуществляются фильтрация логических уровней и синхронизация входных каналов.

1.1.2. Компоненты Multisim

Компоненты – это основа любой схемы, т. е. элементная база, из которой состоит схема. В Multisim работа осуществляется с двумя

категориями компонентов: виртуальными (virtual) (рис. 1.9, а) и реальными (real) (рис. 1.9, б).

Реальные компоненты являются полными аналогами компонентов, выпускаемых или выпущенных радиоэлектронной промышленностью. Виртуальные компоненты – математические модели семейств (Family) компонентов (резисторы, конденсаторы и т. д.) с любыми произвольными параметрами, присущими данной категории.

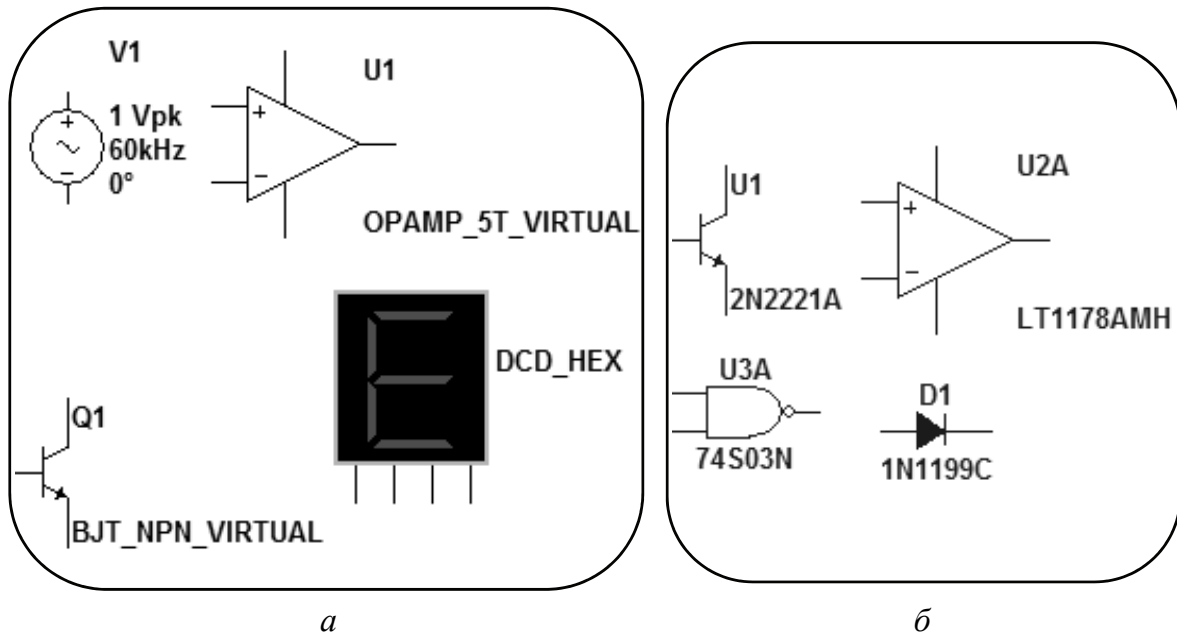


Рис. 1.9. Компоненты Multisim:
а – виртуальные; б – реальные

Добавление компонентов в схему происходит из меню Place либо панели инструментов Components.

Рассмотрим основные группы компонентов базы данных Multisim.

Группа **Sources** (Источники). В данной группе содержатся модели источников питания (однофазный источник питания постоянного (DC_POWER) и переменного напряжения (AC_POWER), трехфазные источники питания, источники питания постоянного тока (VCC1, VDD2, VEE3, VSS4), а также заземление (GROUND)), источников напряжения (источник прямоугольного сигнала, кусочно-линейного сигнала (PWL Voltage) и др.), источников тока и т. д.

Группа **Basic** (Базовые компоненты). В базовую группу входят модели резисторов, конденсаторов, индуктивностей, трансформаторов, виртуальных механических ключей и т. д.

Группа **Diodes** (Диоды). В этой группе содержатся модели таких компонентов, как диод, стабилитрон (zener), светодиод, диодный мост (FWB), диод Шоттки, тиристор и др.

Группа **Transistors** (Транзисторы). В данной группе находятся модели биполярных транзисторов (BJT), полевых транзисторов (JFET), МОП-транзисторов и др.

Группа **Analog** (Аналоговые компоненты) содержит модели операционных усилителей (OPAMP), компараторов (COMPARATOR) и др.

Группа **TTL** (цифровые микросхемы по технологии TTL). В данной группе содержатся модели микросхем серий 74Sxx, 74LSxx, 74ALSxx и др.

Группа **CMOS** (цифровые микросхемы по технологии КМОП) содержит модели микросхем серии 74НСxx, NC7Sx (Tiny Logic) и др.

Группа **Misc Digital** (Цифровые устройства) включает виртуальные модели цифровых устройств (TIL) (логические элементы, триггеры, регистры, счетчики, мультиплексоры, декодеры, элементы арифметико-логических устройств и др.), микросхемы цифровой обработки сигналов (DSP), программируемые логические интегральные схемы, микросхемы памяти, микроконтроллеры и др.

Группа **Indicators** (Индикаторные устройства) включает следующие модели компонентов: индикаторы напряжения и тока, логические пробники, семисегментные индикаторы, звуковые индикаторы и др.

1.1.3. Создание схем в Multisim

Процесс создания схемы начинается с выбора ее компонентов. Выбранный компонент автоматически прикрепляется к курсору мыши, после чего размещается в любом месте рабочего окна (рис. 1.10).

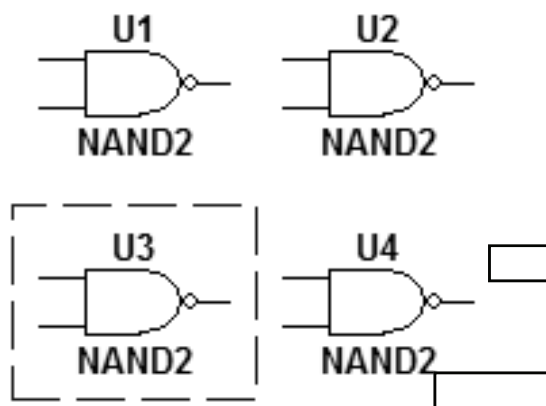


Рис. 1.10. Размещение компонентов схемы

Далее устанавливается ориентация компонентов (если это необходимо) с помощью команд всплывающего меню (нажатие правой кнопки мыши на выбранном компоненте) или комбинации клавиш $\langle Alt + X \rangle$, $\langle Alt + Y \rangle$ и устанавливаются параметры элементов питания.

После размещения компонентов схемы производится соединение их выводов проводниками. При этом необходимо учитывать, что к выводу подключается один проводник. Для выполнения подключения курсор мыши подводится к выводу компонента и после изменения вида курсора \blacklozenge устанавливается соединение между выводами элементов или соединительной точкой (рис. 1.11). Для добавления соединительной точки необходимо нажать комбинацию клавиш $\langle Ctrl + J \rangle$ или выбрать соответствующий пункт во всплывающем меню нажатием правой кнопки мыши.

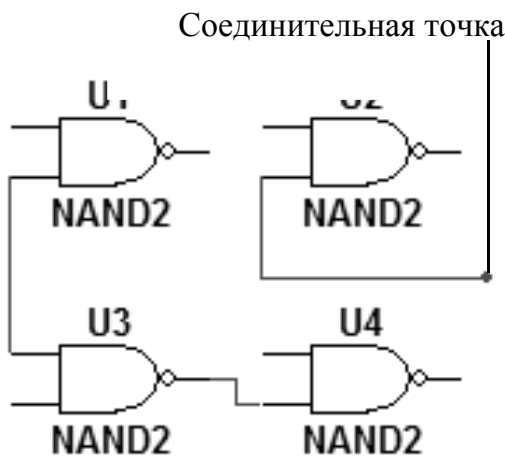


Рис. 1.11. Соединение элементов схемы

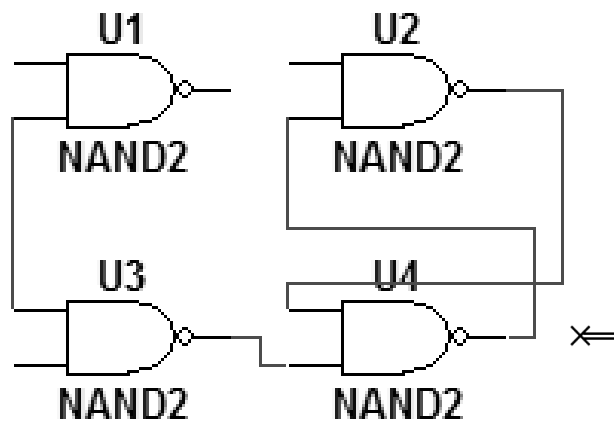


Рис. 1.12. Изменение соединения элементов схемы

При изменении существующего соединения курсор отображается в виде перекрестия с двумя отрезками вдоль соединительной линии $\times \rightleftharpoons$ (рис. 1.12).

1.2. Определение комбинационных и последовательных устройств

Устройства, реализующие функции алгебры логики, называют *логическими* или *цифровыми* и классифицируют по различным отличительным признакам. Так, по характеру информации на входах

и выходах логические устройства подразделяют на устройства последовательного, параллельного и смешанного действия, а по схемному решению и характеру связи между входными и выходными переменными с учетом их изменения по тактам работы – комбинационные и последовательностные.

В *комбинационных* устройствах значения (0 или 1) сигналов на выходах в каждый конкретный момент времени полностью определяются значениями (комбинацией, набором) действующих в данный момент цифровых входных сигналов. В *последовательностных* же устройствах значения выходных сигналов в n -такте определяются не только значениями входных сигналов в этом такте, но и зависят от внутренних состояний устройств, которые произошли в результате воздействия входных сигналов в предшествующие такты.

Данная работа посвящена изучению простейших комбинационных логических устройств, реализующих логические функции сложения, умножения и отрицания.

1.3. Основные элементы алгебры логики

Анализ комбинационных устройств удобно проводить с помощью алгебры логики, оперирующей только с двумя понятиями: истинным (логическая 1) и ложным (логический 0). В результате функции, отображающие информацию, принимают в каждый момент времени только значения 0 или 1. Такие функции называют *логическими*, а сигналы (входные и выходные переменные) – *двоичными* (бинарными).

Схемные элементы, при помощи которых осуществляется преобразование поступающих на их входы двоичных сигналов и непосредственное выполнение предусмотренных логических операций, называют *логическими* устройствами.

В общем случае логическое устройство может иметь n входов и m выходов. Рассматривая входные сигналы x_1, x_2, \dots, x_n в качестве аргументов, можно соответствующие выходные сигналы представлять в виде функции $y_i = f(x_0, x_1, x_2, \dots, x_n)$ с помощью операций алгебры логики.

Функции алгебры логики (ФАЛ), иногда называемые *переключательными* функциями, обычно представляют в алгебраической форме (в виде математического выражения), например $y_i = (x_0 \wedge x_1) \vee (x_1 \wedge x_2)$, или в виде таблиц истинности (комбинационных таблиц).

Таблица истинности содержит всевозможные комбинации (наборы) бинарных значений входных переменных с соответствующими им бинарными значениями выходных переменных; каждому набору входных сигналов соответствует определенное значение выходного сигнала – значение логической функции y_i . Максимальное число возможных различных наборов (строк) зависит от числа входных переменных n и равно 2^n .

В булевой алгебре выделяют три основные функции: конъюнкция, дизъюнкция, отрицание. Остальные функции являются производными от приведенных выше. Формы отображения основных логических функции представлены в табл. 1.1.

Основные логические операции состоят из следующих элементарных преобразований двоичных сигналов:

– *логическое сложение*, или *дизъюнкция*, обозначаемое символом « \vee » (или « $+$ ») и называемое также операцией ИЛИ. При этом число аргументов (слагаемых x) может быть любым. Эта операция для функции двух переменных x_1 и x_2 описывается в виде логической формулы (1.1):

$$y = x_1 \vee x_2 = x_1 + x_2. \quad (1.1)$$

Это значит, что y истинно (равно 1), если истинно хотя бы одно из слагаемых x_1 или x_2 . И только в случае, если все слагаемые x равны 0, результат логического сложения y также равен 0;

– *логическое умножение*, или *конъюнкция*, обозначаемое символом « \wedge » (или « \cdot ») и называемое также операцией И. При этом число аргументов (сомножителей x) может быть любым. Эта операция для функции двух переменных x_1 и x_2 описывается в виде логической формулы (1.2):

$$y = x_1 \wedge x_2 = x_1 \cdot x_2 = x_1 x_2. \quad (1.2)$$

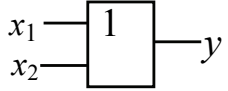
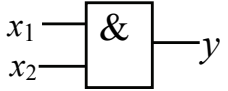
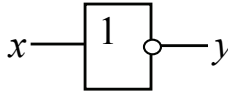
Это значит, что y истинно (равно 1), если истинны сомножители x_1 и x_2 . В случае, если хотя бы один из сомножителей равен 0, результат логического умножения y равен 0;

– *логическое отрицание*, или *инверсия*, обозначаемое черточкой над переменной и называемое операцией НЕ. Эта операция записывается в виде $y = \bar{x}$.

Это значит, что y истинно (равно 1), если x ложно (равно 0), и наоборот. Очевидно, что операция y выполняется над одной переменной x и ее значение всегда противоположно этой переменной.

Таблица 1.1

Формы отображения основных логических функций

| Наименование функции | Дизъюнкция | Конъюнкция | Инверсия | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------------|--|--|---|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|-------|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|---|---|---|---|
| Символическая | \vee или $+$ | \wedge или \cdot | \bar{x} | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Буквенная | ИЛИ | И | НЕ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Условно-графическая |  |  |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Аналитическая | $y = x_1 \vee x_2 = x_1 + x_2$ | $y = x_1 \wedge x_2 = x_1 x_2$ | $y = \bar{x}$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Табличная (истинности) | <table border="1" data-bbox="432 674 735 898"> <thead> <tr><th>x_1</th><th>x_2</th><th>y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> | x_1 | x_2 | y | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | <table border="1" data-bbox="746 674 1050 898"> <thead> <tr><th>x_1</th><th>x_2</th><th>y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> | x_1 | x_2 | y | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | <table border="1" data-bbox="1061 674 1353 898"> <thead> <tr><th>x</th><th>y</th></tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table> | x | y | 0 | 1 | 1 | 0 |
| x_1 | x_2 | y | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| x_1 | x_2 | y | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| x | y | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Основные логические операции ИЛИ, И и НЕ позволяют аналитически описать, а логические элементы ИЛИ (*дизъюнктор*), И (*конъюнктор*) и НЕ (*инвертор*) – реализовать комбинационное устройство любой степени сложности, т. е. операции $y = x_1 + x_2$, $y = x_1 x_2$ и $y = \bar{x}$ обладают функциональной полнотой и составляют функционально полный набор.

В качестве примера рассмотрим функцию неравнозначности у двух переменных x_1 и x_2 , принимающую значение 1 при $x_1 \neq x_2$ и значение 0 при $x_1 = x_2 = 0$ или при $x_1 = x_2 = 1$, т. е. $y = \bar{x}_1 x_2 + x_1 \bar{x}_2$.

Операцию неравнозначности чаще называют *суммированием по модулю 2* и обозначают $y = x_1 \oplus x_2$.

1.4. Базовые логические элементы

Особое значение в цифровой электронике имеют универсальные (базовые) логические элементы, способные образовать функционально полный набор, с помощью которых можно реализовать синтез устройств любой сложности.

При интегральной технологии удобство изготовления одного базового элемента имеет решающее значение. Поэтому базовые логические устройства составляют основу большинства цифровых ИМС.

К универсальным логическим операциям (устройствам) относят две разновидности базовых элементов:

- функцию Пирса, обозначаемую символически вертикальной стрелкой \downarrow (стрелка Пирса) и отображающую операцию ИЛИ-НЕ. Для простейшей функции двух переменных x_1 и x_2 функция $y = 1$ тогда и только тогда, когда $x_1 = x_2 = 0$: $y = x_1 \downarrow x_2 = \overline{x_1 + x_2}$;

- функцию Шеффера, обозначаемую символически вертикальной черточкой $|$ (штрих Шеффера) и отображающую операцию И-НЕ. Для простейшей функции двух переменных x_1 и x_2 функция $y = 0$ тогда и только тогда, когда $x_1 = x_2 = 1$: $y = x_1 | x_2 = \overline{x_1 x_2}$.

При одних и тех же значениях аргументов обе функции отображают операцию инверсии.

1.5. Представление логических функций математическими выражениями

Наиболее распространенным способом задания логических функций является табличная форма.

Таблицы истинности позволяют полно и однозначно установить все существующие логические связи.

При табличном представлении логических функций их записывают в одной из канонических форм: совершенной дизъюнктивной нормальной форме (СДНФ) или совершенной конъюнктивной нормальной форме (СКНФ).

Математическое выражение логической функции в СДНФ получают из таблицы истинности следующим образом: для каждого набора аргументов, на котором функция равна 1, записывают элементарные произведения переменных, причем переменные, значения которых равны нулю, записывают с инверсией. Полученные произведения, называемые *конституентами единицы* или *минтермами*, суммируют.

Запишем логическую функцию (1.3) y трех переменных a , b и c , представленную в виде табл. 1.2, в СДНФ:

$$y(a, b, c) = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc. \quad (1.3)$$

Таблица истинности

| № | a | b | c | y |
|---|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

Совершенной конъюнктивной нормальной формой называют логическое произведение элементарных сумм, в каждую из которых аргумент или его отрицание входят один раз.

При этом для каждого набора аргументов таблицы истинности, на котором функция y равна 0, составляют элементарную сумму, причем переменные, значение которых равно 1, записывают с отрицанием. Полученные суммы, называемые *конституентами нуля* или *макстермами*, объединяют операцией логического умножения.

Для функции СКНФ (1.4):

$$y(a, b, c) = (a + b + c)(a + b + \bar{c})(a + \bar{b} + c)(\bar{a} + b + c). \quad (1.4)$$

1.6. Переход от логической функции к логической схеме

Для построения логической схемы необходимо логические элементы, предназначенные для выполнения логических операций, располагать, начиная от входа, в порядке, указанном в булевом выражении.

Построим структуру логического устройства, реализующего логическую функцию трех переменных:

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

Слева располагаем входы a , b и c с ответвлениями на три инвертора, затем четыре элемента ИЛИ и, наконец, элемент И на выходе (рис. 1.13).

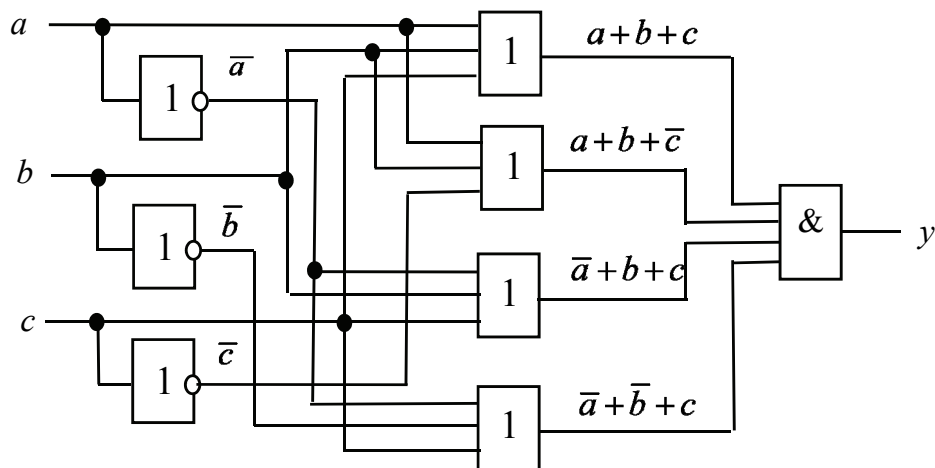


Рис. 1.13. Структура логического устройства

Итак, любую логическую функцию можно реализовать непосредственно по выражениям, представленным в виде СДНФ или СКНФ. Однако полученная таким образом схема, как правило, не оптимальна с точки зрения ее практической реализации: она громоздка, содержит много логических элементов и возникают трудности в обеспечении ее высокой надежности.

Алгебра логики позволяет преобразовать формулы, описывающие сложные высказывания, с целью их упрощения. Это помогает в конечном итоге определить оптимальную структуру того или иного логического устройства, реализующего любую сложную функцию. Под оптимальной структурой принято понимать такое построение логического устройства, при котором число входящих в его состав элементов минимально.

Лабораторная работа № 1

Задание 1. Запустить среду разработки Multisim и собрать на рабочем поле среды Multisim схему для испытания *основных и базовых логических элементов* (рис. 1.14), установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

Схема (рис. 1.14) собрана на двоичных основных [OR (ИЛИ), AND (И) и NOT (НЕ)] и универсальных (базовых) [NAND (И-НЕ) и NOR (ИЛИ-НЕ)] логических элементах, расположенных в библиотеке **Misc Digital/TIL (Цифровые микросхемы/TIL)** с уров-

нем высокого постоянного напряжения 5 В. В схему включены ключи 1, 2, ..., 9, пробники (лампочки) X1, X2, ..., X5 с пороговыми напряжениями 5 В, генератор постоянных токов E1 с напряжением $E = 5$ В и логический анализатор XLA1.

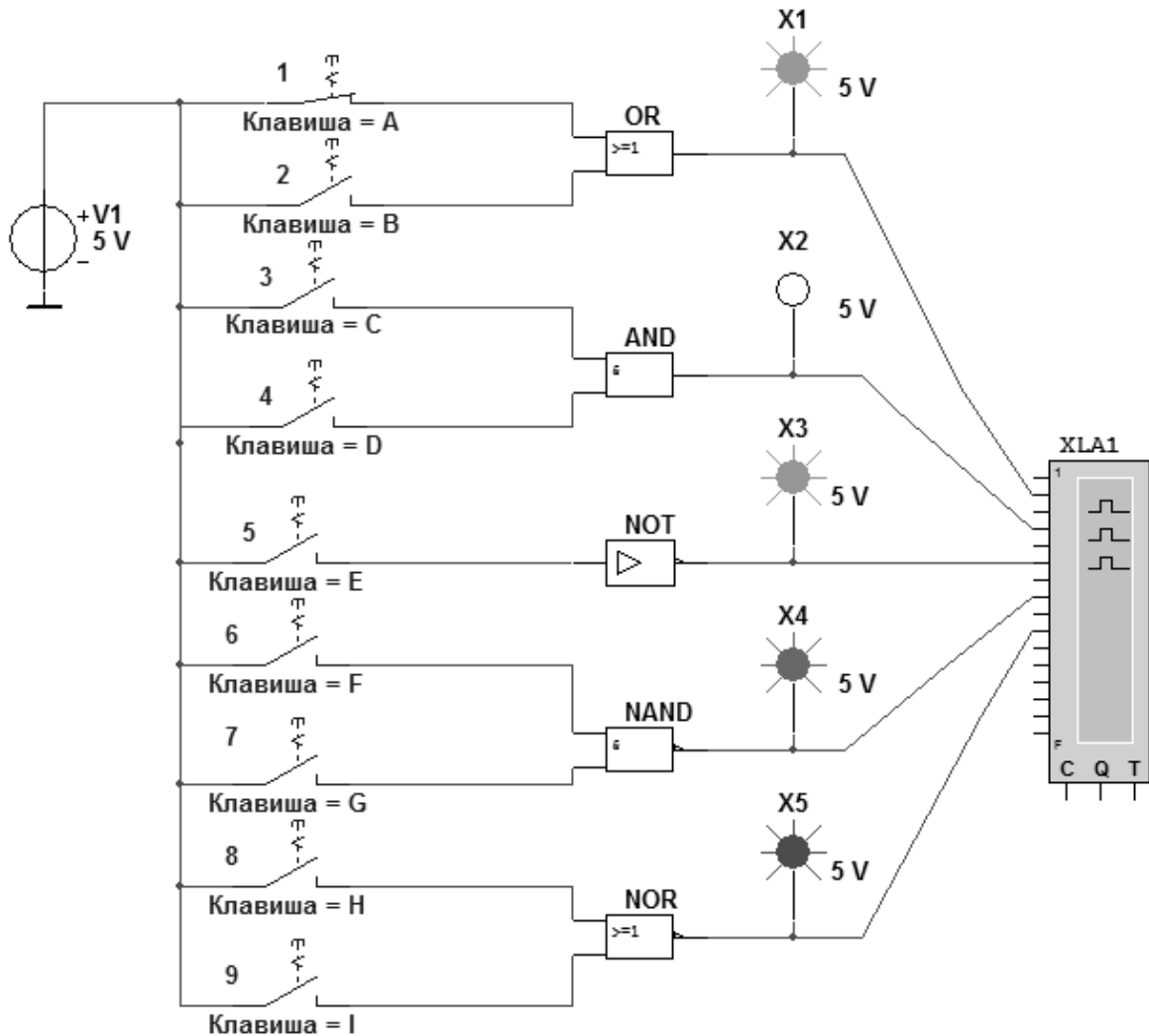


Рис. 1.14. Схема, собранная на двоичных основных и универсальных логических элементах

Для удобства измерения сигналов выходы логических элементов подключены к входам 2, 4, 6, 8 и 10 анализатора XLA1 (рис. 1.15). При моделировании происходит медленная развертка временных диаграмм в окне анализатора. По достижении интервала времени, равного 70–80% ширины окна, следует посредством кнопки <Run/Stop> выключать процесс моделирования.

Оперируя ключами 1, 2, ..., 9, сформировать все возможные комбинации аргументов x_1 и x_2 (00, 10, 01 и 11) на входе дизъюнктора (OR),

конъюнктора (**AND**), штриха Шеффера (**NAND**) и стрелки Пирса (**NOR**) и записать значения выходных логических функций y_k (0 или 1) в табл. 1.3.

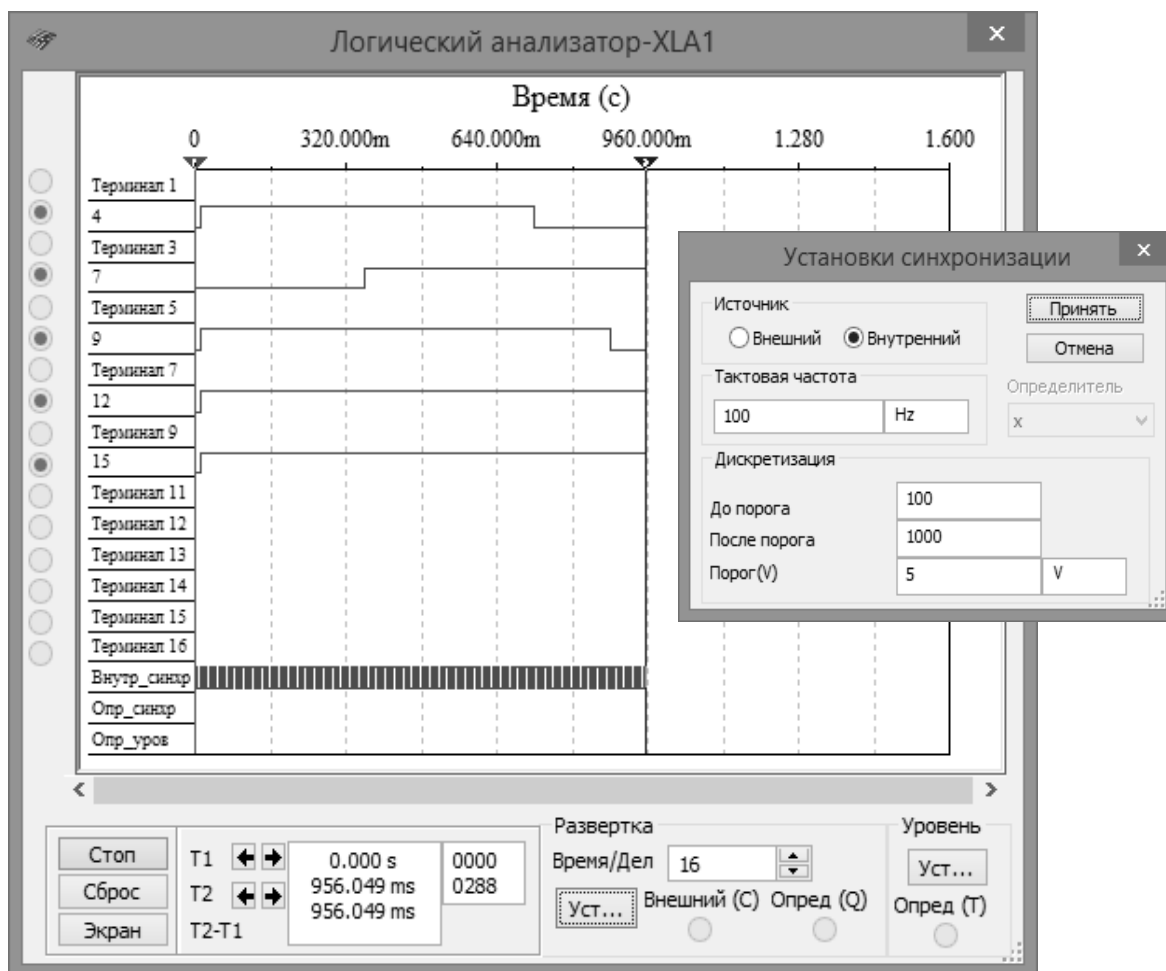


Рис. 1.15. Использование логического анализатора

Заметим, что если ключ замкнут, то на этот вход элемента будет подана логическая единица (положительный потенциал 5 В), а при разомкнутом ключе – логический ноль. Поскольку инвертор (**NOT**) имеет один вход, то для формирования двух значений входного сигнала (логической единицы или логического нуля) достаточно одного ключа **5**.

Значения функций исследуемых элементов можно контролировать с помощью пробников **X1**, **X2**, ..., **X5**: если выходной сигнал элемента равен логической единице, то включенный на выходе этого элемента пробник светится. Так, при положении ключей схемы (рис. 1.14) функции элементов **OR**, **AND** и **NOR** равны логической единице.

Таблица 1.3

Комбинации аргументов

| Дизъюнктор [ИЛИ (OR)] | | | Конъюнктор [И (AND)] | | | Инвертор [НЕ NOT] | | Штрих Шеффера [И-НЕ NAND] | | | Стрелка Пирса [ИЛИ-НЕ (NOR)] | | |
|--------------------------|-------|-----|-------------------------|-------|-----|----------------------|-----|------------------------------|-------|-----|---------------------------------|-------|-----|
| x_1 | x_2 | y | x_1 | x_2 | y | x | y | x_1 | x_2 | y | x_1 | x_2 | y |
| 0 | 0 | | 0 | 0 | | 0 | | 0 | 0 | | 0 | 0 | |
| 0 | 1 | | 0 | 1 | | | | 0 | 1 | | 0 | 1 | |
| 1 | 0 | | 1 | 0 | | 1 | | 1 | 0 | | 1 | 0 | |
| 1 | 1 | | 1 | 1 | | | | 1 | 1 | | 1 | 1 | |

Задание 2. «Перетащить» из библиотеки **Misc Digital\TTL** на рабочее поле среды Multisim необходимые логические элементы и собрать схему для реализации заданной в табл. 1.4 логической функции y с тремя аргументами a , b и c . Скопировать собранную логическую схему на страницу отчета.

Таблица 1.4

Список заданий

| Вариант | Логическая функция |
|----------|--|
| 1, 5, 8 | $y = (a + b + \bar{c})(\bar{a} + \bar{b}c)(a + \bar{b} + \bar{c})$ |
| 2, 6, 9 | $y = (b + a\bar{c})(\bar{a} + bc)(a + \bar{b} + c)$ |
| 3, 7, 10 | $y = (\bar{a}\bar{b} + \bar{c})(a + \bar{b} + c)(ab + \bar{c})$ |
| 4, 8, 11 | $y = (a + \bar{b}c)(\bar{a} + b + \bar{c})(ab + c)$ |

В качестве примера соберем схему для реализации логической функции: $y = (ab + \bar{c})(\bar{a} + \bar{b} + c)(a + b + c)$.

Анализ функции показывает, что для построения логической схемы нам потребуются три инвертора, три дизъюнктора, причем один дизъюнктор с двумя, а два – с тремя входами, и два конъюнктора, причем один с двумя, а другой – с тремя входами.

«Перетащим» на рабочее поле среды Multisim необходимые модели логических элементов из библиотеки **Misc Digital\TTL**, располагая их, начиная с входа, а именно:

– три инвертора NOT (**NOT1**, **NOT2** и **NOT3**) для получения инверсий \bar{a} , \bar{b} и \bar{c} аргументов a , b и c ;

– конъюнктор **AND1** с двумя входами для реализации функции ab ;

– три дизъюнктора: **OR2** для реализации функции $y_1 = a + b + c$, **OR3** для реализации функции $y_2 = \bar{a} + \bar{b} + c$ и **OR1**, реализующий функцию $y_3 = ab + \bar{c}$, разместив их друг под другом (рис. 1.16).

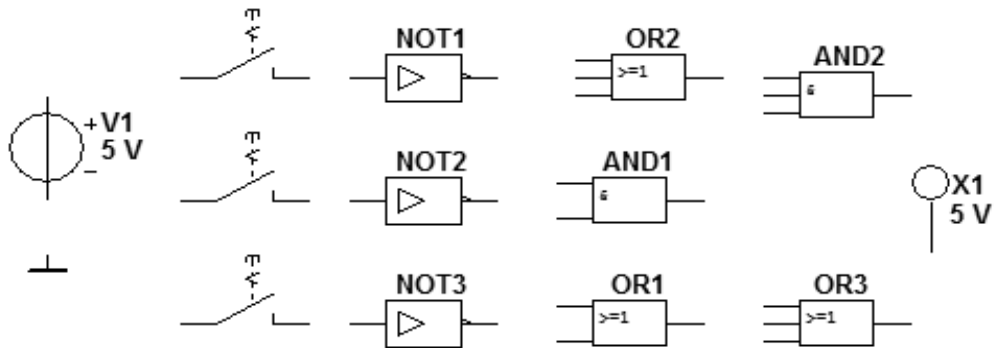


Рис. 1.16. Элементы для построения схемы

Для выполнения функции логического умножения $y = y_1 y_2 y_3$ добавим в схему конъюнктор **AND2** с тремя входами, к выходу которого подключим логический пробник **X1** (уровень высокого напряжения 5 В) для сигнализации появления логической единицы на выходе схемы. «Перетащим» из соответствующих библиотек на рабочее поле источник прямоугольных сигналов **E1** и ключ **1**, расположив их на входе схемы.

Соединив «проводниками» входы и выходы составляющих элементов в соответствии с логическими выражениями заданной функции и записав в отчете ожидаемые результаты выполнения операций на выходах элементов (рис. 1.17), приступим к моделированию.

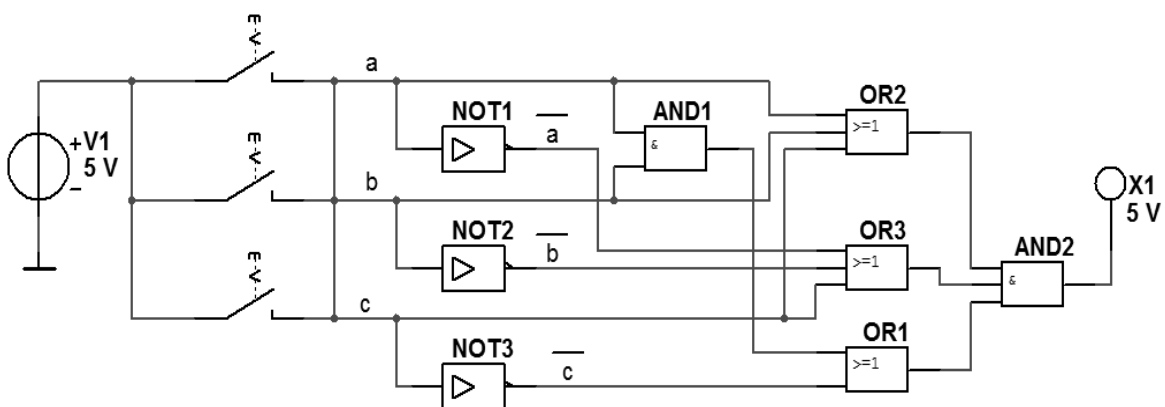


Рис. 1.17. Логическая схема

С этой целью вначале щелкнем мышью на кнопке **<Run/Stop>**, затем нажмем управляющую ключом клавишу с цифрой **1** клавиатуры.

Если соединения элементов выполнены правильно, то пробник **X1** засветится. При выключении ключа **1** пробник гаснет и т. д. По окончании моделирования щелчком мышью на кнопке **<Run/Stop >**.

Примечание. Основным измерительным прибором для проверки цифровых электронных схем является логический пробник. После двойного щелчка мышью на его изображении в открывшемся окне нужно задать уровень высокого напряжения, например 5 В (см. рис 1.4), при котором он светится. Если пробник не светится, то это обычно означает, что уровень проверяемого напряжения находится в промежутке между высоким и низким. Поиск неисправностей нужно начинать с проверки подачи сигналов высокого уровня генератором сигналов на входы элементов, затем проверить правильность выполнения ими логических функций в схеме и проконтролировать появление сигналов на выходах.

Содержание отчета

1. Название и цель работы.
2. Перечень приборов, использованных в экспериментах, с их краткими характеристиками.
3. Изображения электрической схемы для испытания логических элементов и собранной схемы для реализации заданной логической функции.
4. Таблицы истинности, отображающие работу исследуемых логических элементов.
5. Выводы по работе.

ЗАПОМИНАЮЩИЕ ЭЛЕМЕНТЫ. ТРИГГЕРЫ

Триггер – это устройство последовательного типа с двумя устойчивыми состояниями равновесия, предназначенное для записи и хранения информации. Под действием входных сигналов триггер может переключаться из одного устойчивого состояния в другое. При этом напряжение на его выходе скачкообразно изменяется с низкого уровня на высокий или наоборот.

По способу записи информации триггеры делят на *асинхронные*, которые переключаются в момент подачи входного сигнала, и *синхронные* (тактируемые), которые переключаются только при подаче синхронизирующих импульсов, а момент переключения связан с определенным уровнем синхросигнала (*статические* триггеры) либо с моментом перепада напряжения на тактируемом входе (*динамические* триггеры).

Как правило, триггер имеет два выхода: прямой Q и инверсный \bar{Q} . Число входов зависит от структуры и функций, выполняемых триггером. Например, асинхронные RS -триггеры имеют два входа: вход S установки в *единичное* состояние прямого выхода Q и вход R установки в *нулевое* состояние выхода Q . Синхронные триггеры для занесения в них информации, помимо информационных входов S (J) и R (K), имеют синхронизирующий C или счетный T вход, а триггеры задержки – информационный вход D .

Наибольшее распространение в цифровых устройствах получили триггеры RS , D , T и JK .

2.1. Асинхронный и синхронный RS -триггеры

Простейшим триггером является *асинхронный* RS -триггер, условное графическое изображение которого представлено на рис. 2.1, а, а принцип его работы поясняется таблицей переходов на рис. 2.1, д.

Триггер имеет два отдельных информационных входа (R и S) и два выхода (Q и \bar{Q}). Независимым является один (прямой) выход Q , так как инверсный сигнал \bar{Q} можно получить с помощью внешнего инвертора.

Рассмотрим таблицу переходов на рис. 2.1, д. Обозначим Q^t сигнал на выходе триггера до поступления сигнала 1 на его вход S . При подаче сигналов $S = 1$ и $R = 0$ триггер переходит в состояние $Q^{t+1} = 1$. При поступлении сигналов $R = 1$ и $S = 0$ на выходе устанавливается $Q^{t+1} = 0$. При отсутствии новых команд состояние триггера не изменяется: он сохраняет информацию о последней из поступивших команд. Естественно, что комбинация сигналов $S = 1$ и $R = 1$ относится к запрещенным, так как при ее подаче на входы триггера на его выходе Q^{t+1} устанавливается либо 1, либо 0.

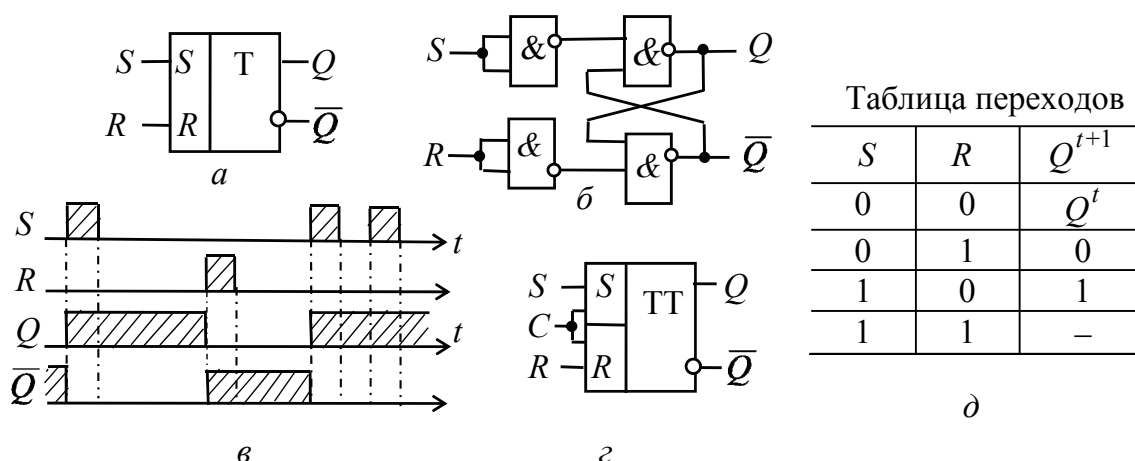


Рис. 2.1. Асинхронный RS-триггер:

a – условное графическое обозначение; *б* – схема реализации на базовых элементах И-НЕ; *в* – временная диаграмма; *г* – преобразование в синхронный; *д* – таблица переходов

На основании таблицы переходов (рис. 2.1, д) запишем аналитическое выражение функционирования RS-триггера (2.1):

$$Q^{t+1} = S + Q^t \bar{R}. \quad (2.1)$$

На рис. 2.1, в изображена временная диаграмма, иллюстрирующая его работу. В момент подачи сигнала $S = 1$ триггер переходит в состояние $Q = 1$. При отсутствии входных сигналов состояние триггера не изменяется, а в момент подачи сигнала $R = 1$ триггер переключается в состояние $Q = 0$, в котором пребывает до поступления нового единичного сигнала на S -вход.

RS-триггер может быть построен на различных логических элементах. На рис. 2.1, б показана схема реализации *RS*-триггера на базовых элементах И-НЕ, в которой использована положительная обратная связь (ПОС) с выходов триггера на входы логических элементов. Именно наличие ПОС отличает триггер от ранее рассмотренных комбинационных логических устройств: посредством сигналов ПОС в триггере фиксируется его предшествующее состояние.

Асинхронный *RS*-триггер можно преобразовать в *синхронный*, если добавить третий синхронизирующий вход *C* (рис. 2.1, з), соединенный, например, с нижними, предварительно разделенными входами двух левых элементов И-НЕ (см. рис. 2.1, б). Вход *C* обеспечивает функционирование *RS*-триггера по закону (2.2):

$$Q^{t+1} = Q^t (\bar{C} + \bar{R}) + CS. \quad (2.2)$$

Переключение синхронного *RS*-триггера в состояние $Q = 1$ происходит при $S = 1$ (или в состояние $Q = 0$ при $R = 1$) в момент прихода синхроимпульса *C*. При $C = 0$ информация с *S*- и *R*-входов на триггер не передается.

2.2. *T*-триггер

Триггер со счетным запуском (*T*-триггер) должен переключаться каждым импульсом, подаваемым на единственный счетный вход *T* (рис. 2.2, а).

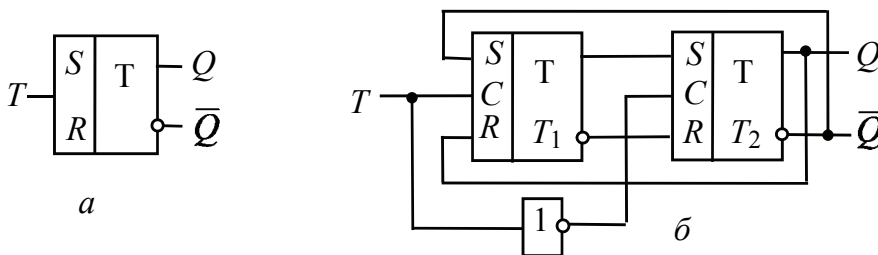


Рис. 2.2. *T*-триггер:

а – условное графическое обозначение; б – схема реализации *T*-триггера на базе двух синхронных *RS*-триггеров

Функционирование *T*-триггера определяется уравнением (2.3):

$$Q^{t+1} = Q^t \bar{T} + \overline{Q^t T}. \quad (2.3)$$

Он может быть реализован, например, на базе двух синхронных RS -триггеров (рис. 2.2, б). С появлением фронта тактового импульса триггер T_1 первой ступени переключается в состояние, противоположное состоянию триггера T_2 . Но это не вызывает изменение сигналов на выходах Q и \bar{Q} , так как за счет инвертора на тактовый вход C триггера T_2 в данный момент подан логический 0. Только на срезе счетного импульса на входе T_1 переключится триггер T_2 и произойдет изменение сигналов на выходах Q и \bar{Q} , а также на S и R -входах первой ступени.

2.3. D -триггер

Триггер задержки (D -триггер) может быть только синхронным, так как имеет один информационный D -вход, информация с которого переписывается на выход триггера только по тактовому сигналу, подаваемому на C -вход.

Условное изображение D -триггера приведено на рис. 2.3, а. Реализовать его можно на различных логических элементах, в том числе на основе синхронного RS -триггера, дополненного инвертором (рис. 2.3, б).

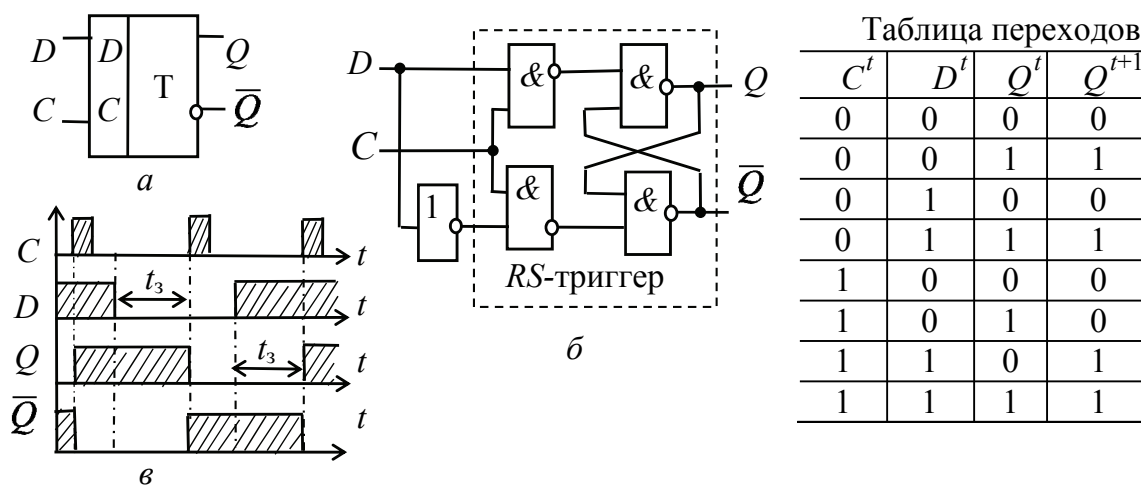


Рис. 2.3. D -триггер:

а – условное графическое обозначение; б – схема реализации;
в – временная диаграмма; г – таблица переходов

Из анализа таблицы переходов на рис. 2.3, г переключательной функции D -триггера (2.4)

$$Q^{t+1} = \bar{C}^t Q^t + C^t Q^t \quad (2.4)$$

следует, что при отсутствии синхроимпульса ($C = 0$) состояние триггера остается неизменным. При условии же $C = 1$ триггер передает на выход сигнал, поступивший на его вход D в предыдущем такте, т. е. выходной сигнал Q^{t+1} изменяется с *задержкой* на один период импульсов синхронизации.

Из анализа временной диаграммы D -триггера (рис. 2.3, в) также следует, что выходной сигнал Q триггера повторяет состояние D -входа с поступлением очередного тактового импульса на вход C с задержкой t_3 относительно сменившегося логического состояния на D -входе.

2.4. *JK-триггер*

JK -триггер обычно выполняют тактируемым. Он имеет информационные входы J и K , которые по своему воздействию на устройство аналогичны входам S и R синхронного RS -триггера: при $J = 1$ и $K = 0$ триггер по тактовому импульсу C устанавливается в состояние $Q = 1$; при $J = 0$ и $K = 1$ – переключается в состояние $Q = 0$, а при $J = 0$ и $K = 0$ – хранит ранее принятую информацию.

В отличие от синхронного RS -триггера одновременное присутствие логических единиц на информационных входах не является для JK -триггера запрещенной комбинацией; при $J = 1$ и $K = 1$ триггер работает в счетном режиме, т. е. переключается каждым тактовым импульсом на входе C .

На рис. 2.4, а изображена одна из функциональных схем JK -триггера. Она отличается от схемы T -триггера (см. рис. 2.2, б) двумя трехходовыми элементами И-НЕ Э1 и Э2 входной логики первой ступени JK -триггера. Переключающий вход C – динамический (рис. 2.4, б), переключение JK -триггера происходит в момент перепада синхроимпульса с уровня $C = 1$ на уровень $C = 0$, т. е. при срезе.

При $J = 0$ и $K = 0$ на выходе элементов Э1 и Э2 устанавливаются логические единицы, которые для триггеров с инверсными входами являются пассивными сигналами: триггер T_1 и, следовательно, JK -триггер в целом сохраняют прежнее состояние (см. рис. 2.4, а). Логическая 1 на одном из входов элемента И-НЕ не определяет 1 на его выходе и комбинация $J = 1$, $K = 1$ никак не влияет на входную логику первой ступени, поэтому схемы T - и JK -триггеров

(см. рис. 2.2, б и рис. 2.4, а) принципиально не отличаются: оба работают в счетном режиме.

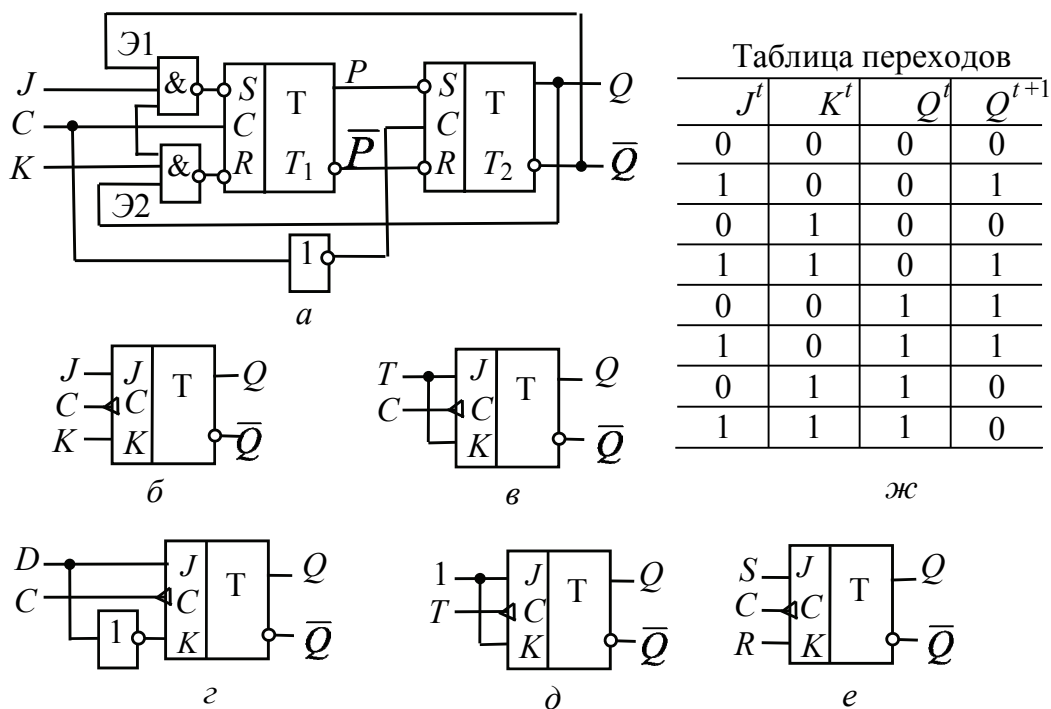


Рис. 2.4. *JK*-триггер:
 а – функциональная схема; б – *T*-триггер;
 в – синхронный *JK*-триггер; г – асинхронный *T*-триггер;
 д – *D*-триггер; е – синхронный *RS*-триггер; ж – таблица переходов

Только при комбинации сигналов $J = 1$, $C = 1$ и $\bar{Q} = 1$ на входе элемента Э1 триггер T_1 переключится в состояние $P = 1$. Аналогично логический 0 будет на выходе элемента Э2 при $K = 1$, $C = 1$ и $Q = 1$.

Таким образом, комбинация $J = 1$, $K = 0$ обуславливает по тактовому импульсу $C = 1$ переключение *JK*-триггера в целом в состояние $Q = 1$, а комбинация $J = 0$, $K = 1$ – в состояние $Q = 0$.

Из анализа таблицы переходов (рис. 2.4, ж) переключательной функции *JK*-триггера (2.5)

$$Q^{t+1} = \bar{K}^t Q^t + J^t \bar{Q}^t \quad (2.5)$$

следует, что состояние триггера определяется не только уровнями сигналов на информационных входах J и K , но и состоянием Q^t , в котором ранее находился *JK*-триггер. Так, при комбинации $J = 0$, $K = 0$ триггер сохраняет предыдущее состояние ($Q^{t+1} = Q^t$); комбинация $J = 1$, $K = 1$ приводит к тому, что тактовым импульсом триггер переключается в состояние, противоположное предыдущему: $Q^{t+1} = \bar{Q}^t$.

Комбинации $J = 1, K = 0$ и $J = 0, K = 1$ дают разрешение триггеру переключиться соответственно в состояния $Q = 1$ и $Q = 0$.

На основе JK -триггера (рис. 2.4, б) могут быть выполнены синхронный (рис. 2.4, в) и асинхронный (рис. 2.4, з) T -триггеры, D -триггер (рис. 2.4, д) и синхронный RS -триггер (рис. 2.4, е).

При проектировании сложных логических схем (микросхем) необходимы триггеры различных типов, которые можно было бы выполнить на основе одного универсального триггера и использовать его в разных режимах работы и модификациях. В интегральной схемотехнике наибольшее распространение получили D - и JK -триггеры.

Лабораторная работа № 2

Задание 1. Запустить среду разработки Multisim и собрать на рабочем поле среды Multisim схему для испытания *асинхронного RS-триггера* (рис. 2.5), установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

Схема (рис. 2.5) собрана на четырех логических элементах И-НЕ (NAND). На входы S и R элементов NAND1 и NAND2 через ключи 1 и 2 подаются логические сигналы 1 или 0 от источника прямоугольных импульсов E1 с амплитудой 5 В. К выходам Q и \bar{Q} элементов NAND3 и NAND4, т. е. к выходам триггера, как и к его входам S и R , подключены пробники X1, X2, X3 и X4 с пороговым напряжением 5 В.

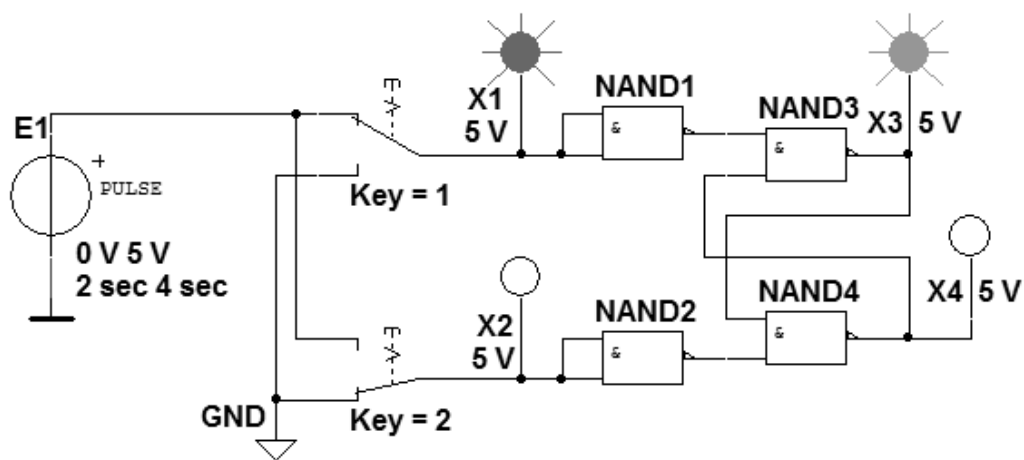


Рис. 2.5. Схема для испытания асинхронного RS -триггера

Воспользовавшись порядком засвечивания разноцветных пробников и задавая коды (00, 01, 10) состояния ключей 1 и 2 (входных сигналов), составить таблицу истинности RS -триггера. Например, сформировав с помощью ключей сигналы $S = 1$ и $R = 0$ и подав их на вход триггера, получите на его выходе сигналы $Q = 1$ и $\bar{Q} = 0$ (см. рис. 2.5). Убедитесь, что при запрещенном коде 11 входных сигналов на выходе RS -триггера могут засветиться оба пробника или оба не светятся.

Задание 2. Подключить к входам триггера логический генератор (генератор слова) **XWG1** (рис. 2.6), запрограммировав его первые три ячейки кодами 00, 10 и 01 и соединив входы и выходы триггера с входами логического анализатора **XLA2**.

В диалоговом окне генератора слова **XWG1** задать частоту $f_r = 10$ кГц и два цикла моделирования сигналов (в режиме **Burst**), а в окне анализатора **XLA2** – частоту $f_a = 0,1$ МГц таймера, уровень высокого напряжения $U_m = 5$ В, число импульсов **Clocks/div** = 8 таймеров, приходящихся на одно деление

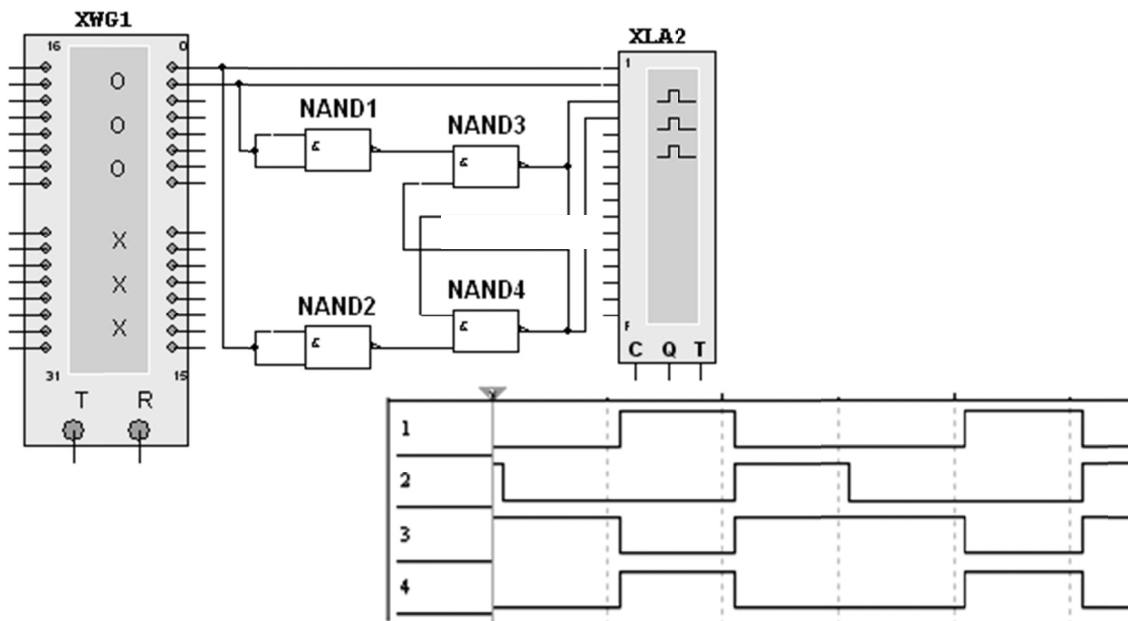


Рис. 2.6. Схема для испытания асинхронного RS -триггера

Получить на экране анализатора **XLA2** временную диаграмму состояний RS -триггера (см. рис. 2.6, внизу). Скопировать схему испытания и временную диаграмму состояния RS -триггера на страницу отчета.

Задание 3. Собрать на рабочем поле среды Multisim схему для испытания триггеров JK , T и D (рис. 2.7) и установить в диалоговых

окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

В схему (рис. 2.7) включены: генератор **XWG1** (частота $f_T = 500$ кГц); логический анализатор **XLA1**; триггеры в интегральном исполнении: универсальный **JK**, счетный **T** и задержки **D**.

На $\overline{1CLR}$ - и $1PK$ -входы триггеров подается постоянное напряжение 5 В (имитирующее сигнал 1) источника **VCC**, а на $1C$ -входы триггеров и на вход 20 анализатора **XLA1** поступают тактовые импульсы с амплитудой 5 В и частотой 500 кГц, сформированные генератором **E1**.

С выходов 1 и 2 генератора **XWG1** сигналы подаются на управляющие входы **1J** и **1K** **JK**-триггера, с выхода 3 – на вход **1D** **T**-триггера, а с выхода 4 – на вход **1D** **D**-триггера.

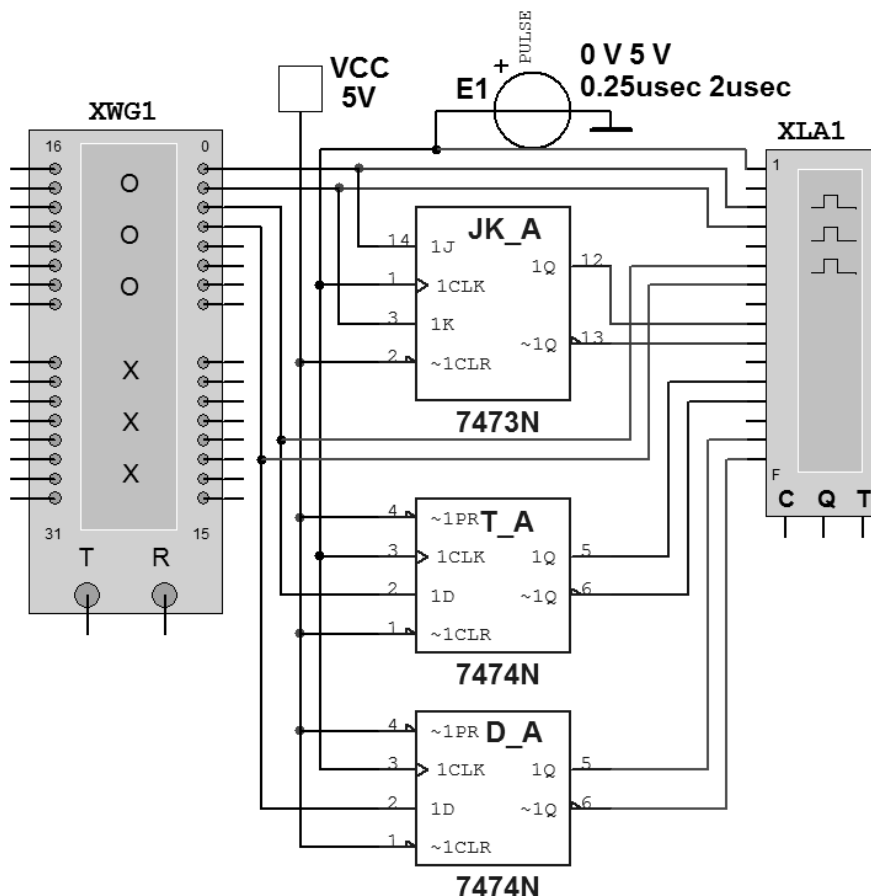


Рис. 2.7. Схема для испытания триггеров **JK**, **T** и **D**

Для формирования выходных сигналов генератор **XWG1** нужно запрограммировать, т. е. ввести в ячейки памяти кодовые комбинации из единиц и нулей согласно варианту (таблица).

Список заданий

| Вариант | Содержимое ячеек памяти генератора слова XWG1 |
|------------|--|
| 1, 6, 11 | 0000, 1010, 1111, 1001, 1001, 1101, 1100, 0000 |
| 2, 7, 12 | 0000, 1100, 1010, 1011, 1001, 1111, 1110, 0000 |
| 3, 8, 13 | 0000, 1010, 1011, 1001, 1001, 1111, 1101, 0000 |
| 4, 9, 14, | 0000, 1111, 1101, 1001, 1011, 1011, 1100, 0000 |
| 5, 10, 15, | 0000, 1011, 1101, 1001, 1100, 1111, 1010, 0000 |

В качестве примера введем в первые восемь ячеек памяти генератора четырехразрядные кодовые комбинации (см. рис. 2.8):

0000, 0101, 1010, 1111, 1001, 1001, 1111, 1100.

При моделировании генератор последовательно и циклично выводит содержимое каждой ячейки памяти (от начальной до конечной) на выходы 1, 2, 3 и 4, формируя на них следующие коды сигналов: 01011110, 00110010, 01010011 и 00111111 (см. сигналы на каналах 1, 2, 3 и 4 логического анализатора **XLA1** (рис. 2.8)). Перед моделированием выделите в окне генератора **XWG1** ячейку с адресом 0 начала счета и вывода сигналов.

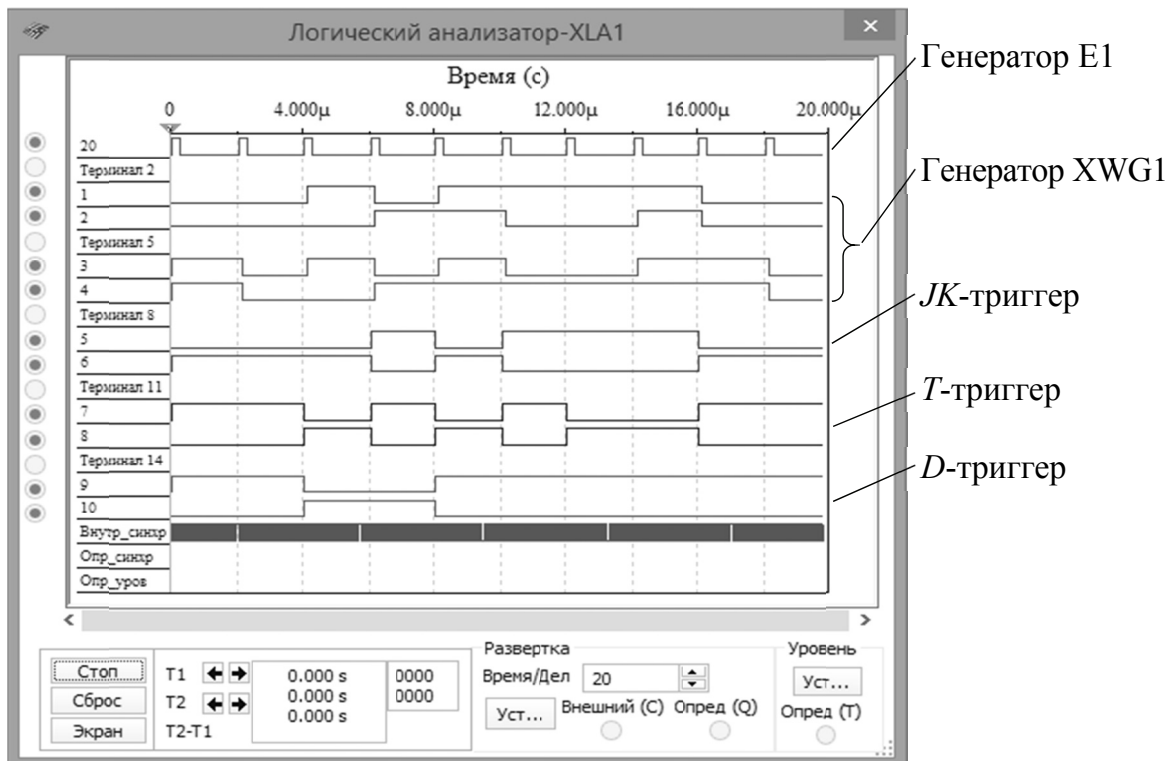


Рис. 2.8. Восемь ячеек памяти генератора с четырехразрядными кодовыми комбинациями

Провести моделирование работы триггеров в режиме Пошагово генератора **XWG1**, скопировать в отчет временные диаграммы, составить и заполнить таблицы истинности работы триггеров *JK*, *T* и *D* при заданном в таблице варианте входных кодовых комбинаций. В частности, описать состояния *JK*-триггера с приходом тактового сигнала $C = 1$, когда сигналы $J = 1$ и $K = 1$, а $Q = 0$ или $Q = 1$.

Содержание отчета

1. Название и цель работы.
2. Перечень приборов, использованных в экспериментах, с их краткими характеристиками.
3. Изображения электрических схем для испытания триггеров *RS*, *JK*, *T* и *D* с помощью логических пробников и логического анализатора **XLA1**. Копии временных диаграмм и таблицы истинности, отображающие работу исследуемых триггеров.
4. Выводы по работе.

ИССЛЕДОВАНИЕ ИНТЕГРАЛЬНЫХ ПРЕОБРАЗОВАТЕЛЕЙ КОДОВ (ДЕШИФРАТОРА, ШИФРАТОРА) И КОММУТАЦИОННЫХ УЗЛОВ (ДЕМУЛЬТИПЛЕКСОРА И МУЛЬТИПЛЕКСОРА)

Кодом называют систему символов для представления информации в форме, удобной для обработки, хранения и передачи. В цифровой технике для записи кодовых символов, или просто кода, используют две цифры: 0 и 1. **Преобразователи кодов** служат для перевода одной формы бинарного числа (кодовой комбинации) в другую, например преобразование двоично-десятичного кода в семисегментный код индикатора. Входные и выходные коды преобразователей связаны между собой. Эту связь задают логическими функциями или в виде таблицы переключений. Рассмотрим наиболее распространенные в цифровой технике виды преобразователей кодов.

3.1. Дешифратор

Дешифратор (DC), или **декодер** – комбинационная схема с n входами и $m = 2^n$ выходами ($m > n$), преобразующая двоичный входной n -код (кодовое слово) в унитарный. На одном из m выходов дешифратора появляется логическая 1, а именно на том, номер которого соответствует поданному на вход двоичному коду.

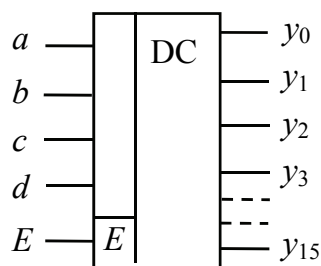


Рис. 3.1. Условное изображение дешифратора

На всех остальных выходах дешифратора выходные сигналы равны нулю. Дешифратор используют, когда нужно обращаться к различным цифровым устройствам по адресу, представленному двоичным кодом.

Условное изображение дешифратора 4×16 (читаемого «четыре шестнадцать») на схемах дано на рис. 3.1. Дешифратор содержит число выходов, равное числу комбинаций входных переменных: от $y_0 = \overline{a}\overline{b}\overline{c}\overline{d}$ до $y_{15} = abcd$ при $n = 4$ и $m = 2^n = 16$.

Применяются также неполные дешифраторы с меньшим числом выходов (10 или 12 при четырех переменных на входе, тогда ряд комбинаций на входе не используется).

Каждый выход полного дешифратора реализует конъюнкцию входных переменных (код адреса) или их инверсий: при наборе $\overline{a}\overline{b}\overline{c}\overline{d}$ (0000) $y_0 = 1$, при $\overline{a}bcd$ (0111) $y_7 = 1$, при $abcd$ (1111) $y_{15} = 1$ и т. д.

Дешифраторы часто имеют *разрешающий* (управляющий, стробирующий) вход E . При $E = 1$ дешифратор функционирует как обычно, при $E = 0$ на всех выходах устанавливается 0 независимо от поступающего кода адреса. Дешифраторы широко применяют во многих устройствах, в том числе в качестве преобразователей двоичного кода в десятичный.

3.2. Шифратор

Шифратор (CD), или *кодер*, выполняет функцию, обратную функции дешифратора. Условное изображение шифратора 16×4 (16 в 4) на схемах показано на рис. 3.2, а.

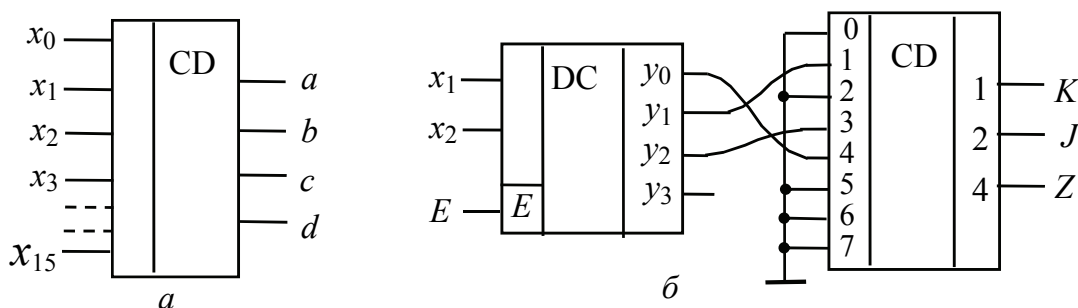


Рис. 3.2. Шифратор:

а – условное изображение; б – схема кодового преобразователя

Классический шифратор имеет n входов и m выходов ($m < n$), при подаче сигнала 1 на один из входов (и не более) на выходе кодера появляется двоичный код номера возбужденного выхода. Число входов и выходов такого шифратора связано соотношением $n = 2^m$.

Области использования шифраторов – отображение в виде двоичного кода номера нажатой кнопки или положения многопозиционного переключателя, а также номера устройства, подавшего сигнал на обслуживание в микропроцессорных системах. Шифраторы входят в состав микросхем контроллеров прерываний, например КР580ВН59.

Для решения многих конкретных задач необходимо синтезировать преобразователи различных кодов. В качестве примера на рис. 3.2, б представлена схема кодового преобразователя, состоящая из пары декодер DC – кодер CD, реализующая логику работы ($y = a + b\bar{c} + \bar{b}c$) некоторого трехцветного светофора K , J и Z , управляемого двухразрядным двоичным кодом X . При этом вначале дешифруется каждая комбинация исходного кода, в результате чего на соответствующем выходе декодера появляется логическая 1. Затем этот логический сигнал, значение которого определено номером выхода декодера, подается на кодер и на его выходах устанавливается преобразованный код.

Число входов дешифратора DC равно двум (x_1 и x_2), число выходов – трем (числу выходов преобразователя) y_0 , y_1 и y_2 . Соединения дешифратора и шифратора выполнены в соответствии с заданной логической функцией y . Часть выходов декодера и входов кодера не используется.

Эффективно стыкуются друг с другом декодер и кодер, построенные на элементах И-НЕ: первый имеет инверсные выходы, а второй – инверсные входы. Если некоторым входным комбинациям соответствует одна и та же выходная, то соответствующие выходы декодера объединяют на элементе ИЛИ и выход последнего подают на нужный вход кодера.

Проектирование кодовой преобразовательной схемы на паре декодер – кодер оказывается в среднем более выгодным и по числу корпусов, и по быстродействию, чем при проектировании из готовых базовых логических микросхем И-НЕ и ИЛИ-НЕ. Однако потребляемая мощность в этом случае может оказаться больше, чем у схемы из отдельных элементов. Затраты времени инженера на логическое проектирование по схеме декодер – кодер неизмеримо меньше, чем затраты на проектирование преобразователя из россыпи.

3.3. Мультиплексор

Мультиплексор (MS) – это функциональный узел, осуществляющий подключение (*коммутацию*) одного из нескольких входов к выходу y . На выход такого устройства передается логический уровень того информационного разряда, номер которого в двоичном коде задан на адресных входах x_1 и x_2 . Может использоваться для преобразования параллельного кода в последовательный. Условное изображение мультиплексора на четыре входа и возможный вариант его структурной схемы показаны на рис. 3.3, *а* и *б*.

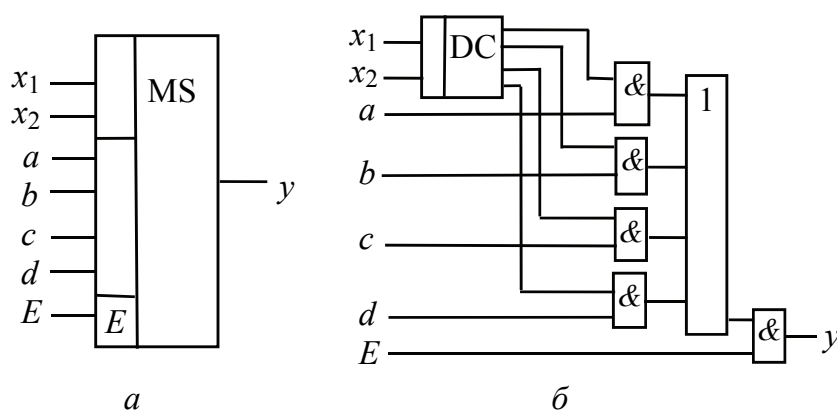


Рис. 3.3. Мультиплексор:
а – условное изображение; *б* – вариант структурной схемы

При $x_1 = 0$ и $x_2 = 0$, $y = a$; при $x_1 = 0$ и $x_2 = 1$, $y = b$; при $x_1 = 1$ и $x_2 = 0$, $y = c$ и при $x_1 = 1$ и $x_2 = 1$, $y = d$.

Функционирование мультиплексора описывается выражением (3.1):

$$y = a\bar{x}_1\bar{x}_2 + b\bar{x}_1x_2 + cx_1\bar{x}_2 + dx_1x_2. \quad (3.1)$$

Вход E – разрешающий: при $E = 1$ мультиплексор работает как обычно, при $E = 0$ выход узла находится в неактивном состоянии, мультиплексор заперт. Серийные узлы выпускаются с числом адресных входов $n = 2, 3$ и 4 при возможном числе 2^n коммутируемых входов. При необходимости коммутировать большее количество входов используют несколько мультиплексоров. Мультиплексоры находят широкое применение в устройствах отображения информации в различных устройствах управления.

Так как мультиплексор может пропустить на выход сигнал с любого информационного входа, адрес которого установлен

на соответствующих адресных входах, то на основе мультиплексов реализуют логические функции, подавая на информационные входы логические 1 или 0 в соответствии с таблицей переключений, а на адресные входы – аргументы функции.

3.4. Демультимплексор

Демультимплексор (DMS) выполняет функцию, обратную функции мультиплексора, т. е. производит коммутацию одного входного сигнала на 2^n выходов, где n – число адресных входов x_i . Демультимплексор имеет один информационный вход D и несколько выходов, причем вход подключается к выходу y_i , имеющему заданный адрес.

В качестве примера на рис. 3.4, б дано условное графическое обозначение демультимплексора, имеющего четыре выхода, закон функционирования которого задан рис. 3.4, а. Пользуясь законом функционирования, запишем переключательные функции для выхода устройства (3.2):

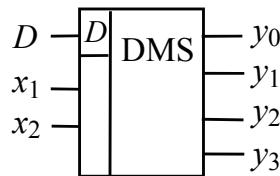
$$y_0 = D\bar{x}_1\bar{x}_2; y_1 = D\bar{x}_1x_2; y_2 = Dx_1\bar{x}_2; y_3 = Dx_1x_2. \quad (3.2)$$

Функциональная схема демультимплексора, реализующая эти выражения, приведена на рис. 3.4, в.

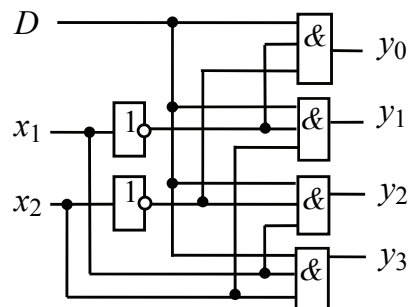
Закон функциональности

| D | x_1 | x_2 | y_3 | y_2 | y_1 | y_0 |
|-----|-------|-------|-------|-------|-------|-------|
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

а



б



в

Рис. 3.4. Демультимплексор:

а – закон функциональности; б – условное графическое изображение;
в – функциональная схема демультимплексора

Если общее число выходов разрабатываемого устройства превышает имеющиеся в выпускаемых интегральных микросхемах, то используют параллельное подключение нескольких схем. На рис. 3.5, а показано демультимплексорное дерево, построенное на мультиплексах с четырьмя выходами.

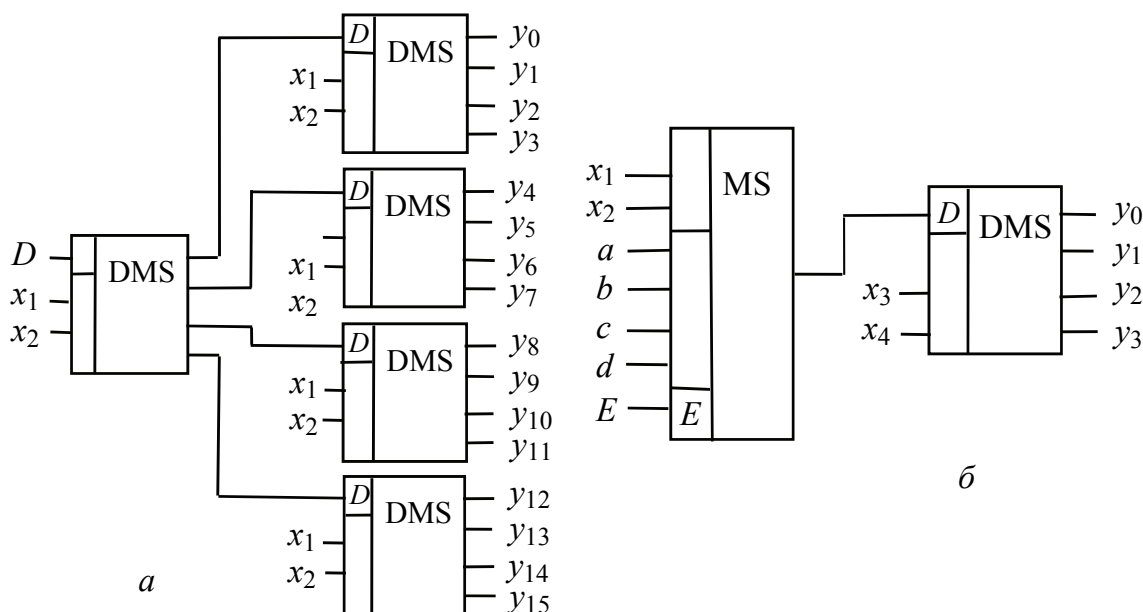


Рис. 3.5. Демультимплексорное дерево:
 а – демультимплексорное дерево, построенное на мультиплексорах с четырьмя выходами; б – комбинационное устройство

Объединяя мультиплексор с демультимплексором, получают комбинационное устройство, в котором по заданным адресам один из входов подключается к одному из его выходов (рис. 3.5, б).

Лабораторные работы № 3–4

Задание 1. Запустить среду разработки Multisim, собрать на рабочем поле этой среды схему для испытания *дешифратора ДС* (рис. 3.6), установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

Схема содержит интегральный *дешифратор ДС (decoder)* 3×8 , имеющий 3 информационных входа **A**, **B** и **C** (для кода 4–2–1), 8 выходов (Y_0, \dots, Y_7) и преобразующий позиционный 3-разрядный двоичный код в *унитарный* «1 из 8»: в выходной 8-разрядной кодовой комбинации только одна позиция занята единицей, а все остальные – нулевыми (см. рис. 3.7, справа). В зависимости от входного двоичного кода, например 001, на выходе **ДС** появляется сигнал 1 только на одной (второй, см. рис. 3.6) из 8 выходных линий, к которым подключены пробники **X0**, ..., **X7**.

Данный тип шифратора относится к шифраторам с разным уровнем входных и выходных сигналов: активные входные уровни соответствуют уровню логической 1, а активные выходные сигналы – уровню логического 0. Для получения активных выходных уровней, равных 1, к выходам дешифратора подключено восемь инверторов $C0, \dots, C7$.

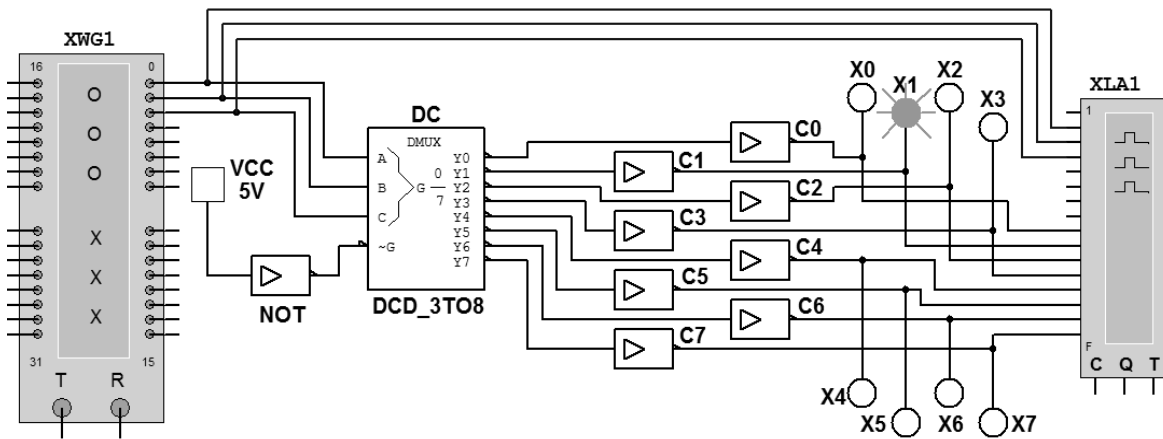


Рис. 3.6. Схема для испытаний дешифратора

Логический генератор слова **XWG1** ($f_r = 500$ кГц) с записанными логическими словами в ячейки памяти, которые эквивалентны десятичным числам от 0 до 7, представлен на рис. 3.7.

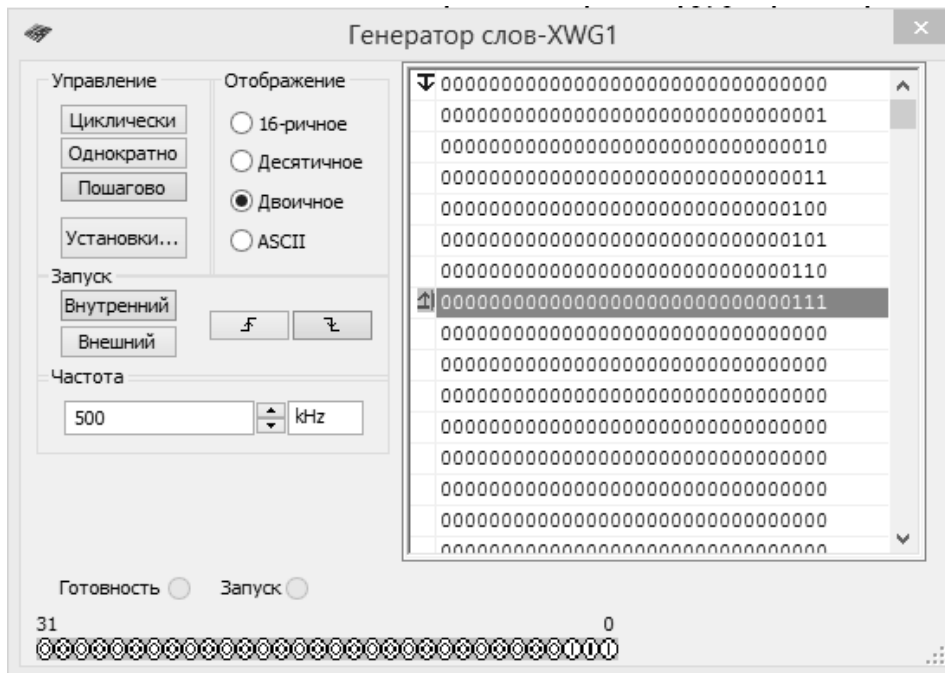


Рис. 3.7. Логический генератор

На экран логического анализатора **XLA1** (представлен на рис. 3.8) выводятся временные диаграммы как трех входных (**A**, **B**, **C**), так и восьми (**Y0**, **Y1**, ..., **Y7**) выходных сигналов при Пошаговом режиме генератора **XWG1**.

С выхода источника **VCC** напряжение 5 В подано на инвертор **NOT**. Логический 0 с инвертора подается на управляющий вход дешифратора **DC**: при $\bar{G} = 0$ дешифратор находится в активном состоянии.

Запустить программу моделирования дешифратора. Щелкая мышью по кнопке <Пошагово> генератора **XWG1**, последовательно подавать на вход дешифратора логические слова.

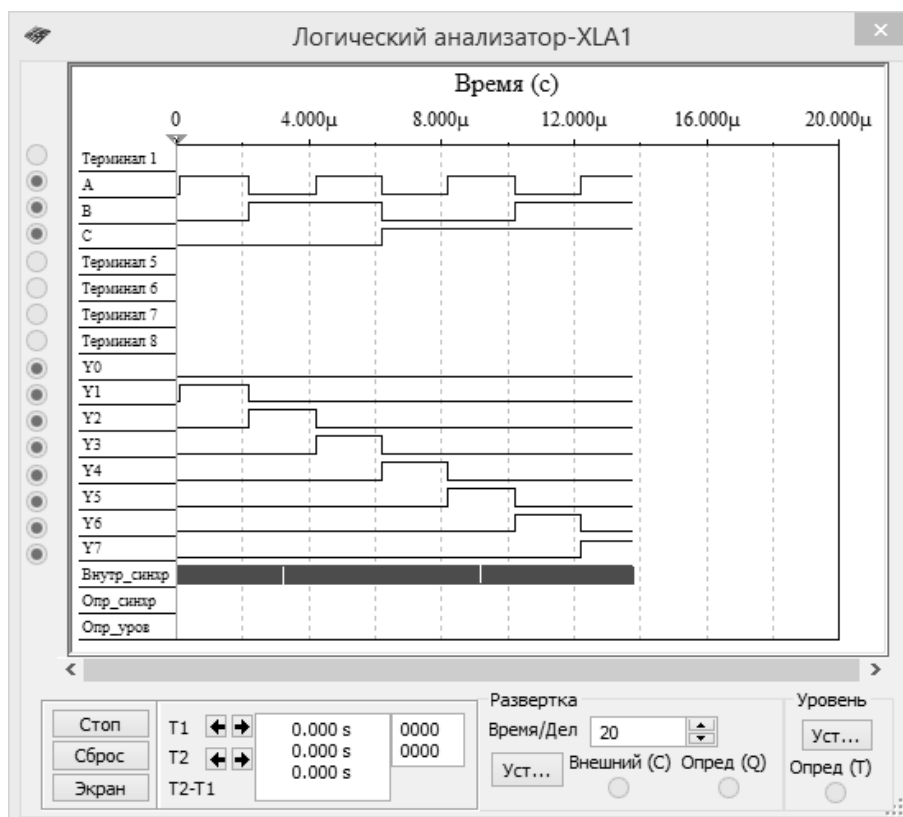


Рис. 3.8. Логический анализатор

Убедиться, что при подаче на вход дешифратора каждой новой двоичной кодовой комбинации засвечивается только один пробник, который «распознает» свой входной код.

Скопировать временные диаграммы входных и выходных сигналов дешифратора на страницу отчета. По результатам моделирования составить и заполнить табл. 3.1 переключений (функций $Y_i = (A_i B_i C_i; G_i)$) на выходах дешифратора **DC** 3×8 .

Переключение на выходах дешифратора DC 3×8

| | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|---|---|
| A | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| B | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| C | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| Y0 | | | | | | | | | |
| Y1 | | | | | | | | | |
| Y2 | | | | | | | | | |
| Y3 | | | | | | | | | |
| Y4 | | | | | | | | | |
| Y5 | | | | | | | | | |
| Y6 | | | | | | | | | |
| Y7 | | | | | | | | | |

Задание 2. Собрать на рабочем поле среды Multisim схему для испытания шифратора CD (рис. 3.9) и установить в диалоговых окнах компонентов их параметры или режимы работы.

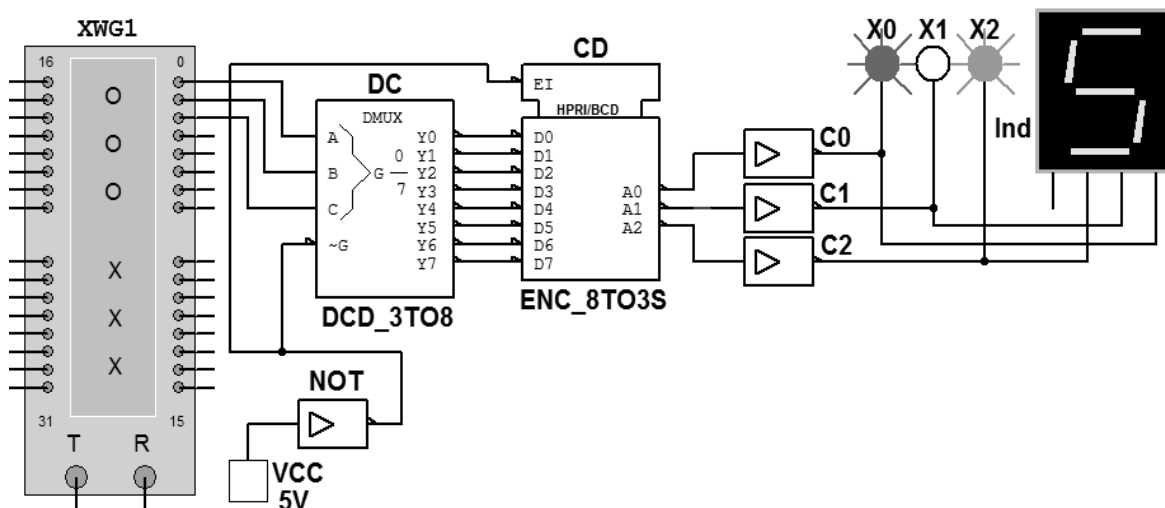


Рис. 3.9. Схема для испытаний шифратора

Скопировать схему на страницу отчета.

Запустить программу моделирования шифратора. Щелкая мышью на кнопке <Пошагово> генератора XWG1, последовательно подавать на вход дешифратора логические слова. Убедиться, что при подаче с выхода DC на вход шифратора CD 8-разрядной последовательности, в которой только одна позиция занята единицей, а остальные – нулями, на выходе шифратора формируются 3-разрядные двоичные коды A0A1A2, где A0 = A, A1 = B и A2 = C, соответствующие двоичным кодовым комбинациям на входе дешифратора DC.

По результатам моделирования (по засвечиванию логических пробников **X0**, **X1**, **X2** и показаниям индикатора **Ind**) составить и заполнить таблицу переключений на выходе шифратора **CD 8×3** (табл. 3.2).

Таблица 3.2

Переключение на выходах шифратора **DC 3×8**

| | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|---|---|
| A | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| B | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| C | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| Y0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Y1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Y2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Y3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Y4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Y5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Y6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Y7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| A0 | | | | | | | | | |
| A1 | | | | | | | | | |
| A2 | | | | | | | | | |

Преобразовать схему дешифратора **DC 3×8** и шифратора **CD 8×3** (см. рис. 3.10) в схему **DC 2×4** и шифратора **CD 4×2**, отсоединив провод **C**, подходящий к дешифратору, и провод **A2** с выхода шифратора, и составить таблицы переключений дешифратора **2×4** и шифратора **4×2**.

Задание 3. Собрать на рабочем поле среды Multisim схему для испытания *демультиплексора DMS* (рис. 3.10) и установить в диалоговых окнах компонентов их параметры или режимы работы.

Демультиплексор DMS (Логические микросхемы CMOS\74НС 4V - 74НС137D_4V) 1×8 (из 1 в 8) (рис. 3.10) имеет один информационный вход (с активными высоким **G1** и низким **G2** уровнями), три адресных **A**, **B**, **C** входа, разрешающий **GL**-вход с активным низким уровнем и восемь **Y0**, **Y1**, ..., **Y7** инверсных выходов, соединенных с входами логического анализатора **XLA1**. На вход анализатора также подаются сигналы с адресных входов **A**, **B**, **C**. С помощью ключей **A**, **B** и **C** можно сформировать восемь трехразрядных двоичных адресных слов. При последовательной подаче формируемых ключами адресных слов от 111 до 000 на экран анализатора **XLA1** при моделировании выводятся 8-разрядные кодовые последовательности с одним активным (низким) уровнем.

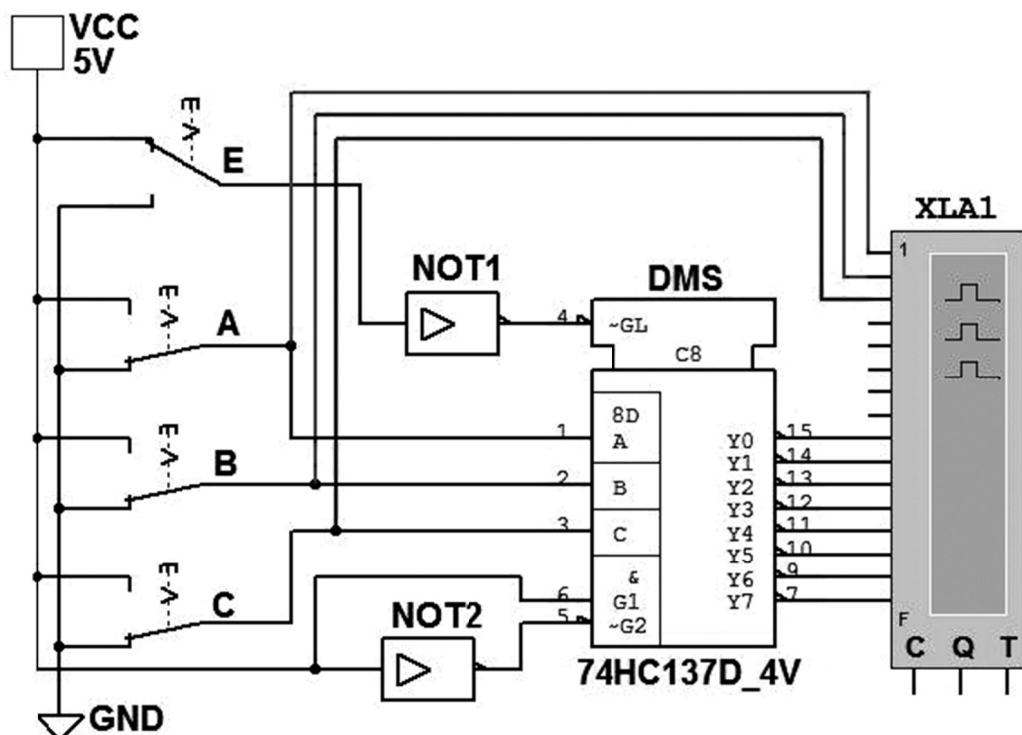


Рис. 3.10. Схема для испытаний демультиплексора

Для обеспечения медленного перемещения лучей на экране анализатора **XLA1** установить тактовую частоту его таймера $f_a = 500$ Гц и число импульсов, приходящихся на одно деление, **Время/Дел** = 80.

Задать код ключей 111 и щелкнуть мышью на кнопке **<Run/Stop>**. Кривые адресных и выходных логических сигналов медленно разворачиваются во времени на экране анализатора.

Повторять перечисленные выше операции для спадающих счетных комбинаций адресных сигналов (с 110 до 000) до тех пор, пока не будет записан процесс моделирования при адресном слове 000.

Убедиться, что для каждой комбинации адресных сигналов демультиплексор формирует логический 0 на одном из восьми выходов, номер которого соответствует определенному кодовому слову на входе, т. е. демультиплексор подобен коммутатору, посредством которого поток цифровой информации разделяется на 8 выходных потоков.

Скопировать схему (рис. 3.10) и временные диаграммы входных и выходных сигналов на страницу отчета.

Если адресные входы **A**, **B** и **C** принять в качестве информационных входов, а вход **G1** (**G2**) в качестве входа разрешения работы, то мультиплексор превратится в дешифратор.

Задание 4. Собрать на рабочем поле среды Multisim схему (рис. 3.11) для испытания *мультиплексора MS 8×1 (из 8 в 1)* (Логические микросхемы TTL\74STD\74151N) и установить в диалоговых окнах компонентов их параметры или режимы работы.

Мультиплексор MS с разрешающим входом G осуществляет передачу сигнала с каждого информационного входа D0, D1, ..., D7, заданного 3-разрядным кодом ABC – адресом выбираемого входа, на единственный выход Y. Разрядность (3) управляющего сигнала определяет количество входов ($2^3 = 8$), с которых мультиплексор может принимать информацию. Если предположить, что к входам D0, D1, ..., D7 мультиплексора MS присоединено 8 источников цифровых сигналов – генераторов последовательных двоичных слов, то байты от любого из них можно передавать на выход Y.

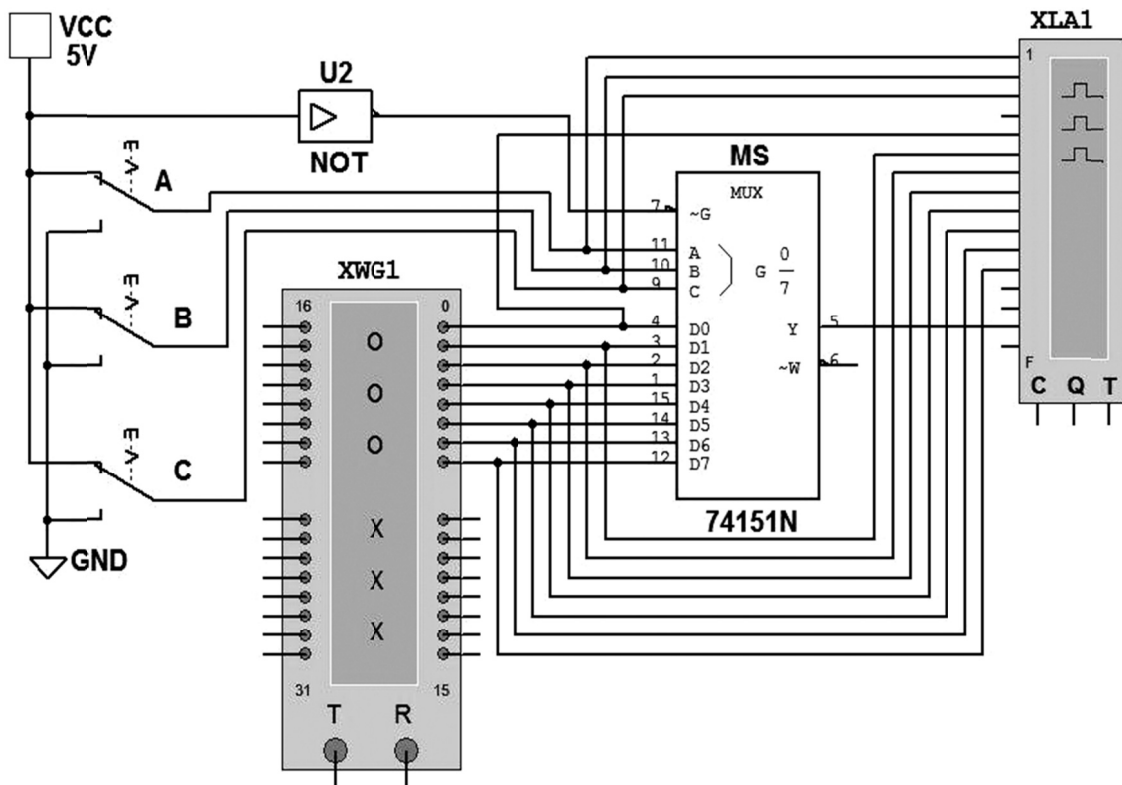


Рис. 3.11. Схема для испытаний мультиплексора

Скопировать схему (рис. 3.11) на страницу отчета.

Для иллюстрации работы мультиплексора MS запишем в ячейки памяти генератора XWG1 произвольные 8-разрядные кодовые слова (рис. 3.12), а с помощью ключей A, B, C сформируем управляющий сигнал **111**.

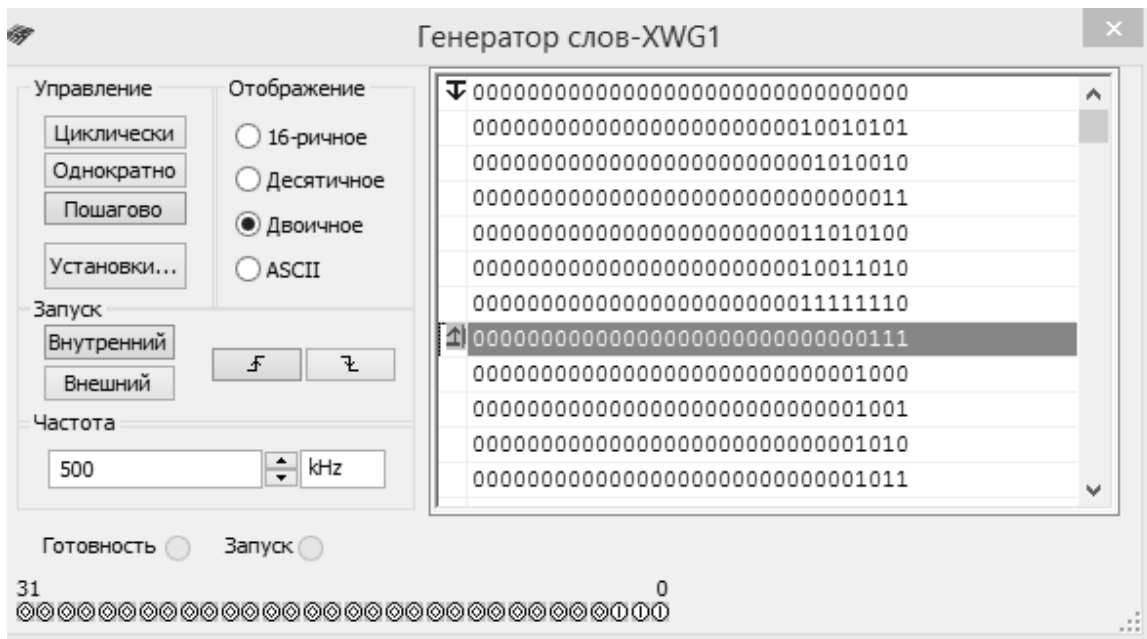


Рис. 3.12. Генератор XWG1 с произвольными 8-разрядными кодовыми словами

Последовательно щелкая мышью по кнопке <Пошагово> генератора XWG1, при $G = 1$ поступающие на вход D7 мультиплексора байты (сигнал 01001110) с восьмого разряда (на рис. 3.12, слева 8-й разряд показан стрелкой) логических слов генератора XWG1 передаются на выход Y и на вход анализатора (см. рис. 3.13).

Если ключ A установить в нижнее положение (сформировав тем самым адресный код 011), то со входа D3 на выход Y мультиплексора будут поступать байты четвертого разряда логических слов, записанных в ячейки памяти генератора XWG1, и т. д.

Записать в первые восемь ячеек памяти генератора XWG1 произвольные 8-разрядные кодовые слова, задать частоту $f_r = 500$ кГц и режим Пошагово его работы (см. рис. 3.13).

Задать частоту $f_a = 20$ МГц таймера логического анализатора XLA1 и количество импульсов таймера $\text{Clock/div} = 20$, приходящихся на одно деление.

Установить с помощью ключей A, B и C адресный код, например 100_2 (4_{10}), и запустить программу моделирования мультиплексора. Получить и скопировать временные диаграммы входных сигналов D0, D1, ..., D7 и выходного сигнала Y мультиплексора на страницу отчета.

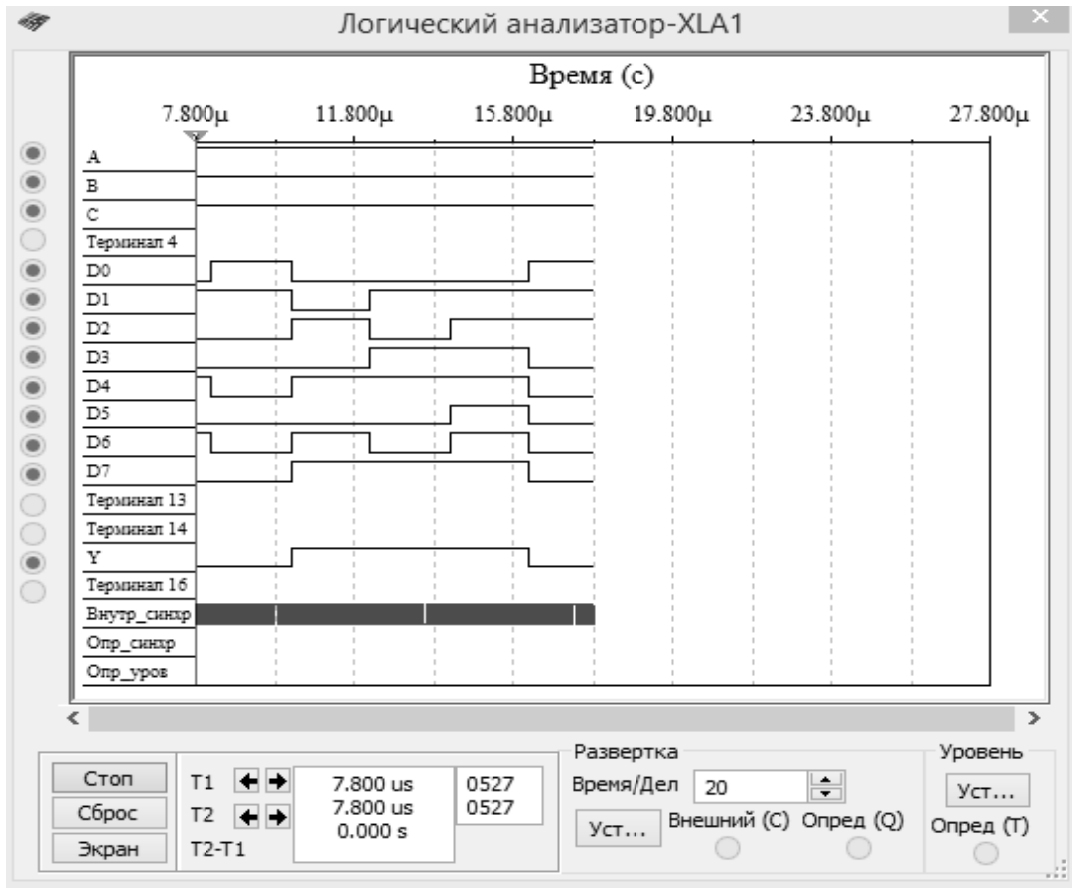


Рис. 3.13. Анализатор XLA1 с произвольными 8-разрядными кодовыми словами

Задание 5. Собрать на рабочем поле среды Multisim схему для испытания *демультиплексора DMS 1×16 (из 1 в 16)* (рис. 3.14) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему (рис. 3.14) в отчет.

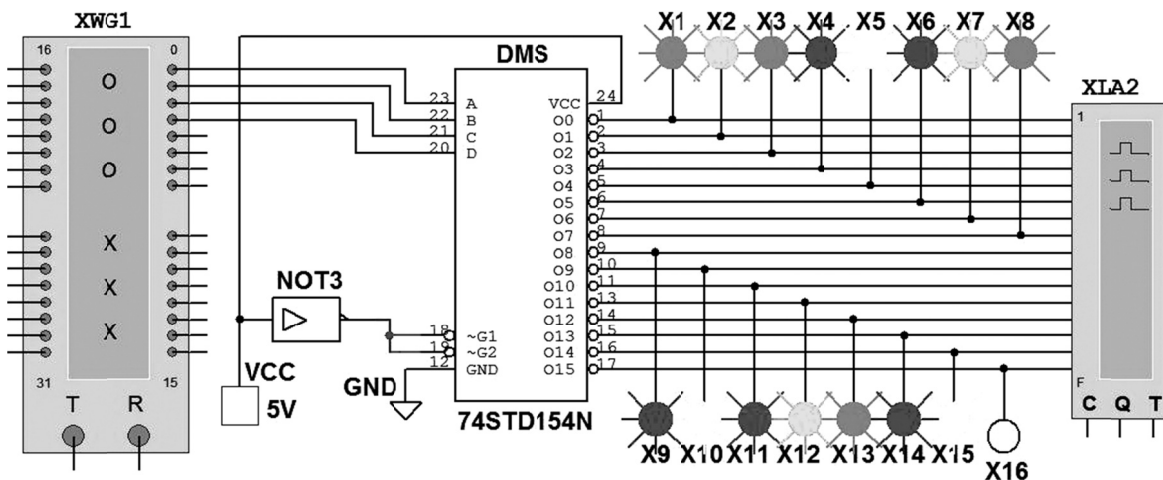


Рис. 3.14. Схема для испытаний демультиплексора

С целью автоматизации процесса моделирования к входу демультимплектора **DMS** подключен логический генератор **XWG1** с записанными в его ячейки памяти адресными кодами от 0000 до 1111, а для визуализации сигналов на выходах включены 16 логических пробников **X1**, **X2**, ..., **X16** и логический анализатор **XLA2**.

Запустить программу моделирования демультимплектора **DMS** 1×16. Последовательно подавать (щелкая мышью по кнопке <Пошагово> генератора **XWG1**) (рис. 3.15) на вход демультимплектора логические слова, начиная с комбинации 0000 адресного сигнала и заканчивая комбинацией 1111, наблюдать за изменениями выходных сигналов по показаниям индикаторов и в окне анализатора **XLA2** (рис. 3.16).

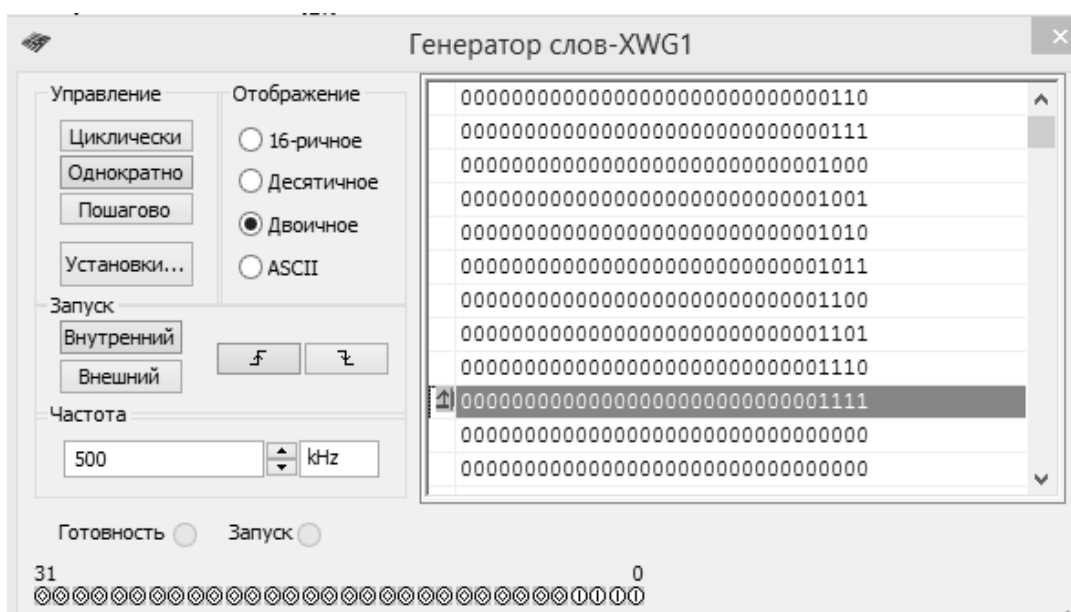


Рис. 3.15. Логический генератор с записанными в него ячейками памяти

В исследуемой модели демультимплектора соответствующий активный выход имеет низкий логический уровень (рис. 3.14), поэтому пробник на этом выходе не светится. Так, при подаче последней кодовой комбинации 1111 на вход демультимплектора не светится пробник **X16**, так как активным является выход **15** (см. рис. 3.14).

Скопировать на страницу отчета временные диаграммы выходных сигналов демультимплектора **DMS** 1×16.

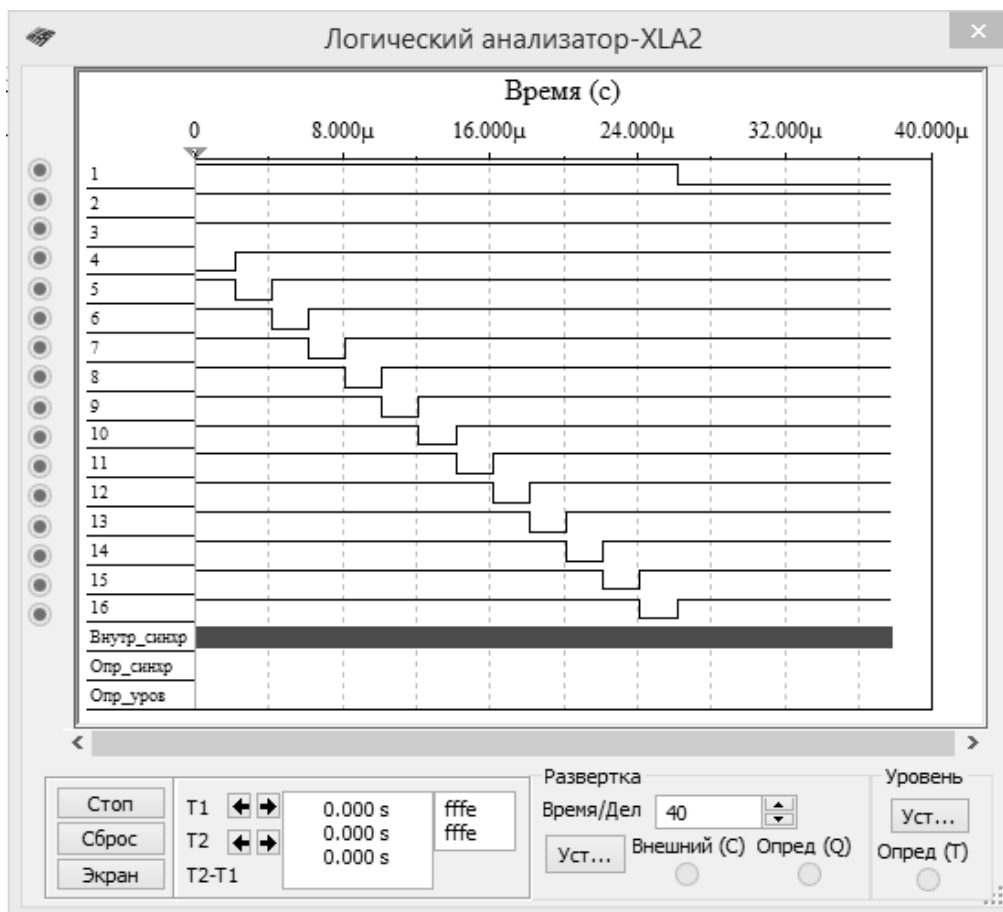


Рис. 3.16. Показания логического анализатора при исследовании демультиплексора

Содержание отчета

1. Название и цель работы.
2. Перечень приборов, использованных в экспериментах, с их краткими характеристиками.
3. Изображения электрических схем для испытания дешифратора, шифратора, демультиплексора и мультиплексора.
4. Копии временных диаграмм и таблицы переключений, отображающие работу исследуемых преобразователей кодов.
5. Выводы по работе.

ЗАПОМИНАЮЩИЕ УЗЛЫ. РЕГИСТРЫ

4.1. Классификация регистров

Регистр – это последовательностное устройство, предназначенное для записи, хранения и (или) сдвига информации, которая поступает и хранится в виде n -разрядных двоичных чисел. В общем случае регистр может выдавать информацию в последовательной или параллельной форме, *преобразовывать* прямой код числа в обратный (когда единицы заменяются нулями, а нули – единицами) и наоборот.

В зависимости от способа ввода и вывода разрядов числа различают регистры параллельные, последовательные и параллельно-последовательные. В *параллельном* регистре ввод и вывод всех разрядов кодового числа осуществляется одновременно, в *последовательном* – разряды числа вводятся и выводятся последовательно, а в *параллельно-последовательном* регистре ввод числа производится в параллельной форме, а вывод – в последовательной и наоборот. Преобразование параллельного кода в последовательный и наоборот – очень актуальная задача, так как передача цифровой информации в сетях передачи данных осуществляется в последовательном коде, а обработка ее в микропроцессорах вычислительных устройств – в параллельном.

Регистр, в котором можно осуществить сдвиг числа, называют *сдвигающим* (сдвиговым), причем сдвиг может быть или в одну сторону (в сторону младшего разряда – *прямой* (правый) *сдвиг*, или в сторону старшего разряда – *обратный* (левый) *сдвиг*), или в обе стороны (*реверсивный сдвигающий*) регистр. В этом смысле последовательный и параллельно-последовательный регистры относят к сдвиговым.

В качестве примера на рис. 4.1 приведено изображение четырехразрядного регистра (микросхема серии К155). При $V_2 = 0$ разряды числа вводят последовательно в регистр через вход V_1 ; синхроимпульсы, поступающие на вход C_1 , обеспечивают сдвиг вправо разрядов числа; регистр работает как сдвигающий. В микросхеме

(рис. 4.1) предусмотрен также параллельный ввод всех разрядов числа по синхроимпульсу на входе C_2 с входов D_1, \dots, D_4 при $V_2 = 1$. В данном случае регистр работает как параллельный.

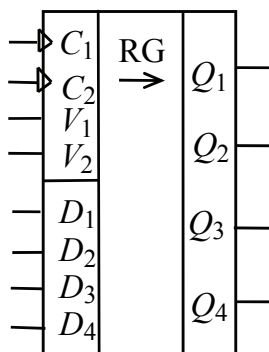


Рис. 4.1. Четырехразрядный регистр (микросхема серии К155)

Если выводы последнего триггера сдвигающего регистра соединить с входами первого, то получится *кольцевой* регистр сдвига, называемый *кольцевым счетчиком*. Его коэффициент пересчета равен числу разрядов n : единица, записанная в один из разрядов, периодически появляется на выходе счетчика после того, как пройдут n сдвигающих синхроимпульсов.

4.2. Параллельный регистр на RS -триггерах

Любой регистр состоит из связанных между собой триггеров с динамическим или статическим управлением и логических элементов, причем количество триггеров равно количеству разрядов в записываемом числе. Синтез регистра сводится к выбору типа триггеров и логических элементов И, НЕ, ИЛИ для реализации заданных операций.

Рассмотрим работу параллельного регистра на RS -триггерах (рис. 4.2). Ввод (запись) числа осуществляется в два такта. Во избежание ошибочной записи числа $x_1x_2\dots x_n$ в первом такте все триггеры регистра обнуляются. Для этого на шину «0» подается логический 0. Во втором такте по сигналу 1 на шине «П» («Прием») через конъюнкторы одновременно записывается в соответствующие разряды регистра двоичное число $x_1x_2\dots x_n$. Вывод (считывание) числа $y_1y_2\dots y_n$ в прямом коде происходит по сигналу 1 на шине «В_{пр}», а в обратном – по сигналу 1 на шине «В_{обр}».

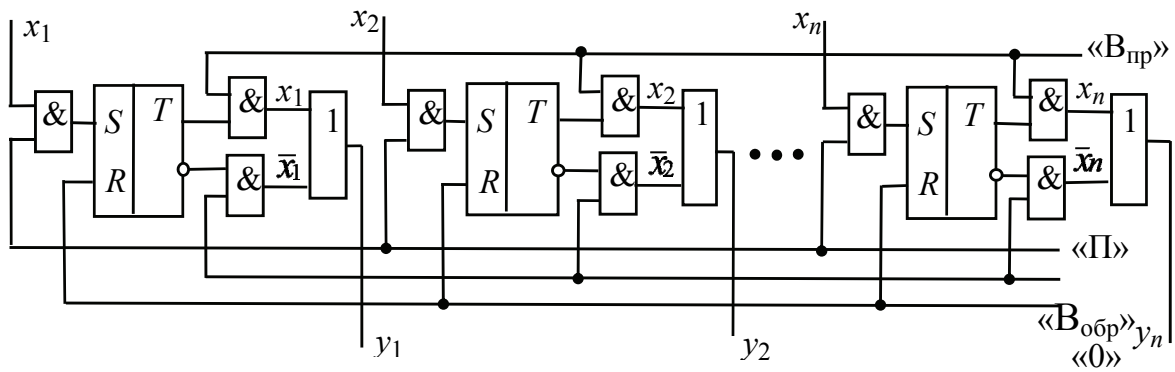


Рис. 4.2. Параллельный регистр на RS-триггерах

Объединив в одной микросхеме несколько регистров и добавив на входе дешифратор DCW, а на выходе мультиплексор MS, получают *регистровую (сверхоперативную) память* (рис. 4.3). Входы D_i четырех или восьми регистров, как правило 4-разрядных, подключают к общей входной шине данных DIN .

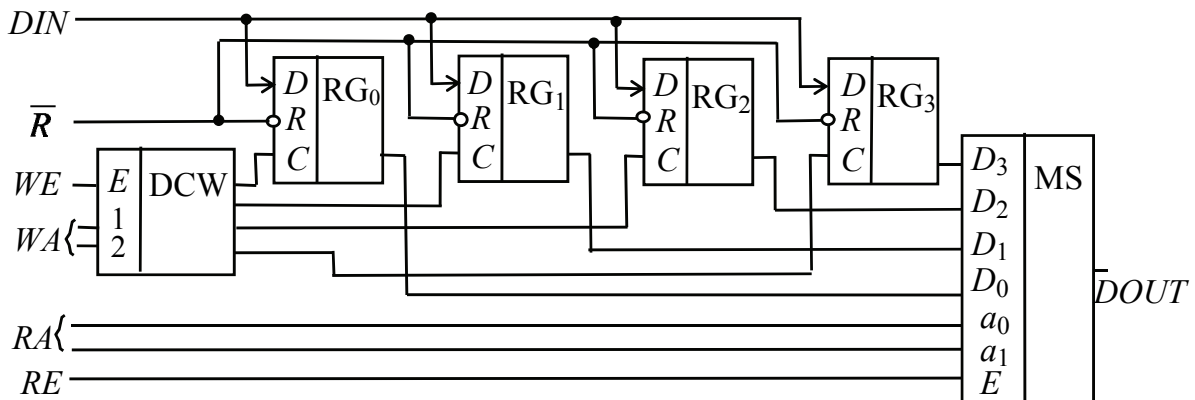


Рис. 4.3. Регистровая (сверхоперативная) память

Вход загрузки требуемого регистра выбирается дешифратором записи DCW на основании поступающего на его вход адреса записи WA , т. е. кода номера загружаемого регистра. *Запись* данных, присутствующих на шине DIN , происходит в момент поступления сигнала разрешения записи WE .

Выходы регистров мультиплексором MS подключаются к выходной шине $DOUT$. Номер регистра, с которого происходит *чтение*, определяется посредством кода адреса чтения RA . Разрешение выдачи данных в шину $DOUT$ осуществляется по сигналу RE . Поскольку дешифрация адреса записи и адреса чтения производится двумя независимыми узлами, имеющими автономные адресные входы

WA и RA , в регистровую память можно одновременно записывать бинарное число в один из регистров и считывать число из другого.

Микросхемы регистровой памяти легко наращиваются по разрядности и допускают наращивание по числу регистров. Они разработаны для построения блоков *регистров общего назначения* (РОН), предназначенных для временного хранения исходных данных и промежуточных результатов расчета в микропроцессорах.

Лабораторная работа № 5

Задание 1. Запустить среду разработки Multisim, собрать на рабочем поле среды Multisim схему для испытания *универсального регистра сдвига* (рис. 4.4), установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

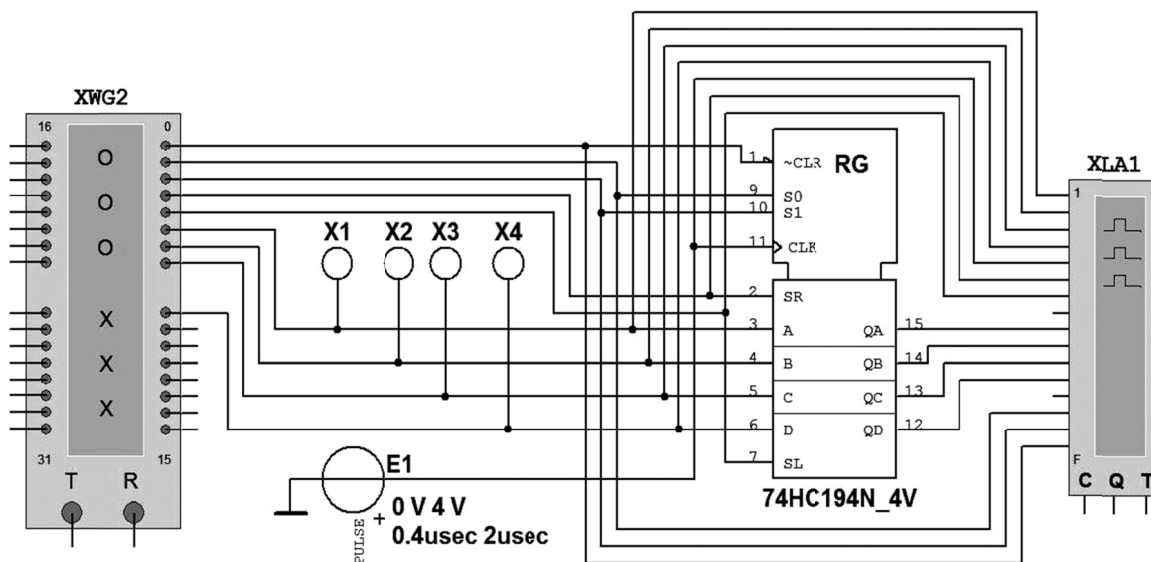


Рис. 4.4. Схема для испытания универсального регистра сдвига

Универсальный 4-разрядный регистр сдвига 74HC194N_4V (отечественные аналоги-микросхемы К230ИР2, КМ155ИР1, К176ИР3) способен сдвигать информацию и вправо, и влево, возможна как параллельная, так и последовательная запись данных. Регистр имеет параллельные входы (A, B, C, D), параллельные выходы (QA, QB, QC, QD), последовательные входы (SR, SL), цепь прямой очистки регистра по входу \overline{CLR} и управляющие входы (S0 и S1) – входы задания режима:

$S0 = 1, S1 = 1$ – запись данных в регистр по входам A, B, C, D;
 $S0 = 1, S1 = 0$ – сдвиг данных влево в направлении от QA к QD;
 $S0 = 0, S1 = 1$ – сдвиг данных вправо в направлении от QD к QA;
 $S0 = 0, S1 = 0$ – входы регистра недоступны (блокировка).

Задание 2. Составить план исследования параллельного регистра сдвига, заполнив ячейки памяти генератора слова XWG1 на основе правил функционирования регистра 74HC194_4V, отраженных в таблице.

План исследования параллельного регистра сдвига

| Входы | | | | | | | | | | Выходы | | | |
|------------------|------------------|-------|----|-----------------------|----|-------------------|---|---|---|-----------------|-----------------|-----------------|-----------------|
| Сброс | Старт | Режим | | Последовательный вход | | Параллельный вход | | | | | | | |
| \overline{CLR} | \overline{CLX} | S0 | S1 | SR | SL | A | B | C | D | QA | QB | QC | QD |
| 0 | × | × | × | × | × | × | × | × | × | 0 | 0 | 0 | 0 |
| 1 | 0 | × | × | × | × | × | × | × | × | QA ₀ | QB ₀ | QC ₀ | QD ₀ |
| 1 | ↑ | 1 | 1 | × | × | A | B | C | D | A | B | C | D |
| 1 | ↑ | 1 | 0 | 1 | × | × | × | × | × | 1 | QA _n | QB _n | QC _n |
| 1 | ↑ | 1 | 0 | 0 | × | × | × | × | × | 0 | QA _n | QB _n | QC _n |
| 1 | ↑ | 0 | 1 | × | 1 | × | × | × | × | QB _n | QC _n | QD _n | 1 |
| 1 | ↑ | 0 | 1 | × | 0 | × | × | × | × | QB _n | QC _n | QD _n | 0 |
| 1 | × | 0 | 0 | × | × | × | × | × | × | QA ₀ | QB ₀ | QC ₀ | QD ₀ |

Примечание. 0 – низкий уровень; 1 – высокий уровень; × – любое состояние; ↑ – положительный перепад (с низкого уровня на высокий); QA₀, QB₀, QC₀, QD₀ – стационарные уровни A, B, C, D до установки указанных состояний на входах; QA_n, QB_n, QC_n, QD_n – соответственно уровни A, B, C, D перед началом прохождения фронта самого последнего тактового импульса.

Запустить программу моделирования параллельного регистра, скопировать в отчет программу и временные диаграммы сигналов на входах и выходах регистра.

Ввод (запись) и вывод (считывание) информации производится параллельным кодом. Ввод обеспечивается тактовым импульсом, с приходом очередного тактового импульса записанная информация обновляется. Считывание информации происходит в прямом коде в интервале между синхроимпульсами при нахождении триггеров в режиме хранения.

Руководствуясь схемой соединения генератора XWG1 с регистром (см. рис. 4.4), при записи чисел в ячейки памяти генератора

в младший разряд 9-разрядных чисел нужно заносить значение сигнала CLR: логический 0 для очистки регистра или логическая 1 – разрешение записи числа, сдвига данных и др.; в следующие два разряда – значения (1 или 0) сигналов S0 и S1, определяющих режим работы регистра; в два следующих – вводить значения сигналов SR и SL, определяющих направление сдвига записанной информации в направлении от QA к QB, QC, а затем к QD после каждого положительного перепада импульса на тактовом входе CLK или наоборот от QD к QA. В старшие разряды нужно занести произвольные (или по указанию преподавателя) значения 4-разрядных чисел DCBA, которые передаются на соответствующие выходы.

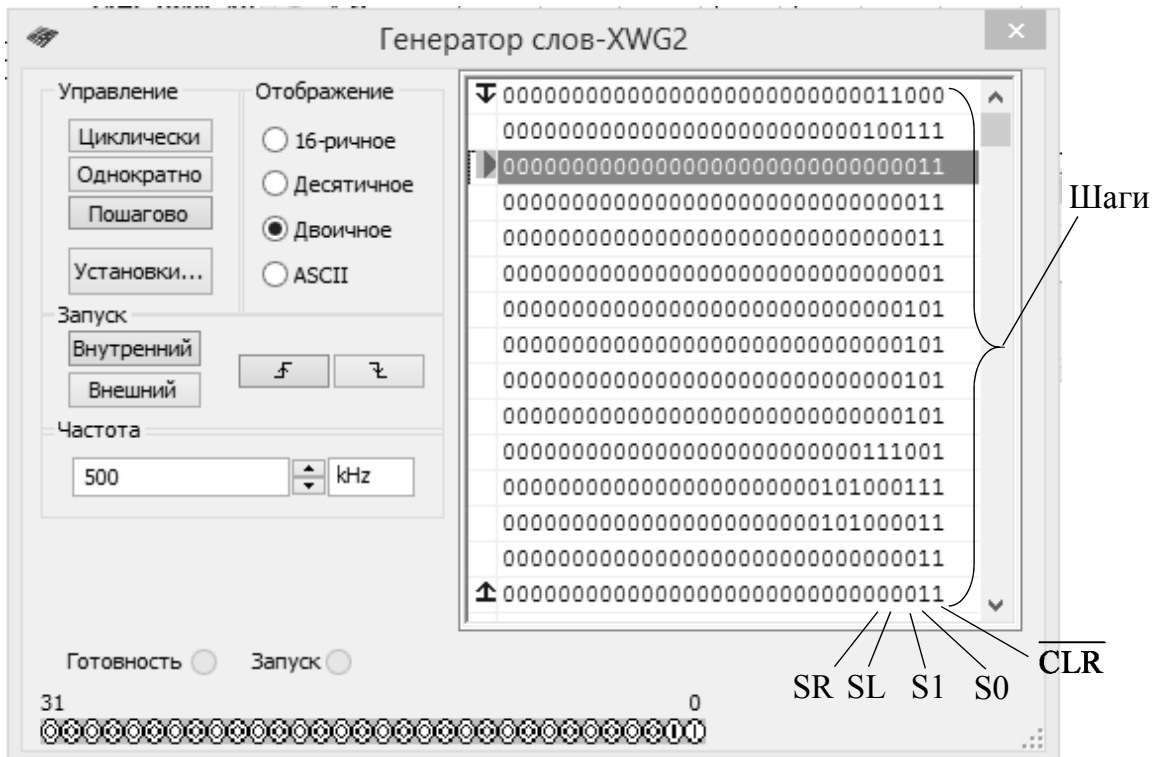


Рис. 4.5. Запись 9-разрядных кодовых комбинаций в 15 ячеек памяти генератора

В качестве примера на рис. 4.5 приведена запись 9-разрядных кодовых комбинаций в 15 ячеек памяти генератора XWG1, а на рис. 4.6 – реализация программы моделирования параллельного регистра в виде временных диаграмм сигналов (выводимых в окне анализатора XLA1) на его входах и выходах при Шаговом режиме работы генератора XWG1.

На первом шаге (первом такте работы генератора и регистра) при подаче сигнала $\overline{CLR} = 0$ (см. первую строку таблицы и рис. 4.6) на всех входах и выходах регистра установились нулевые значения. На втором шаге при $\overline{CLR} = 1$, $SR = 0$, $SL = 0$ (разрешение записи числа в регистр), $S0 = 1$ и $S1 = 1$ (запрещение сдвига данных во время синхронной параллельной записи числа в регистр, см. третью строку таблицы) происходит загрузка 4-разрядного двоичного числа $DCBA = 0001$ в регистр.

При задании направления сдвига данных влево ($S0 = 1$ и $S1 = 0$, такт или шаг 3) сигнал 0001 выводится на выходы: $QD = 0$, $QC = 0$, $QB = 0$ и $QA = 1$. С приходом очередного тактового импульса (шаги 4, 5 и 6) происходит перезапись (сдвиг) содержимого триггера каждого разряда в соседний разряд (от разряда А к разряду D) без изменения порядка следования единиц и нулей. По окончании шестого тактового импульса на выходе устанавливается число 1000 (см. рис. 4.6). Если выполнить еще один шаг при $S0 = 1$ и $S1 = 0$, то занесенная в регистр информация будет полностью из него выведена. Если при работе регистра в режиме сдвига влево (см. шаги 3, ..., 6 на рис. 4.6) в ячейки памяти генератора внести $SL = 1$, то сигнал 1 будет формироваться на выходе QA и сдвигаться влево от QA к QD при каждом тактовом импульсе. В результате после шестого импульса на выходе установится сигнал 1111.

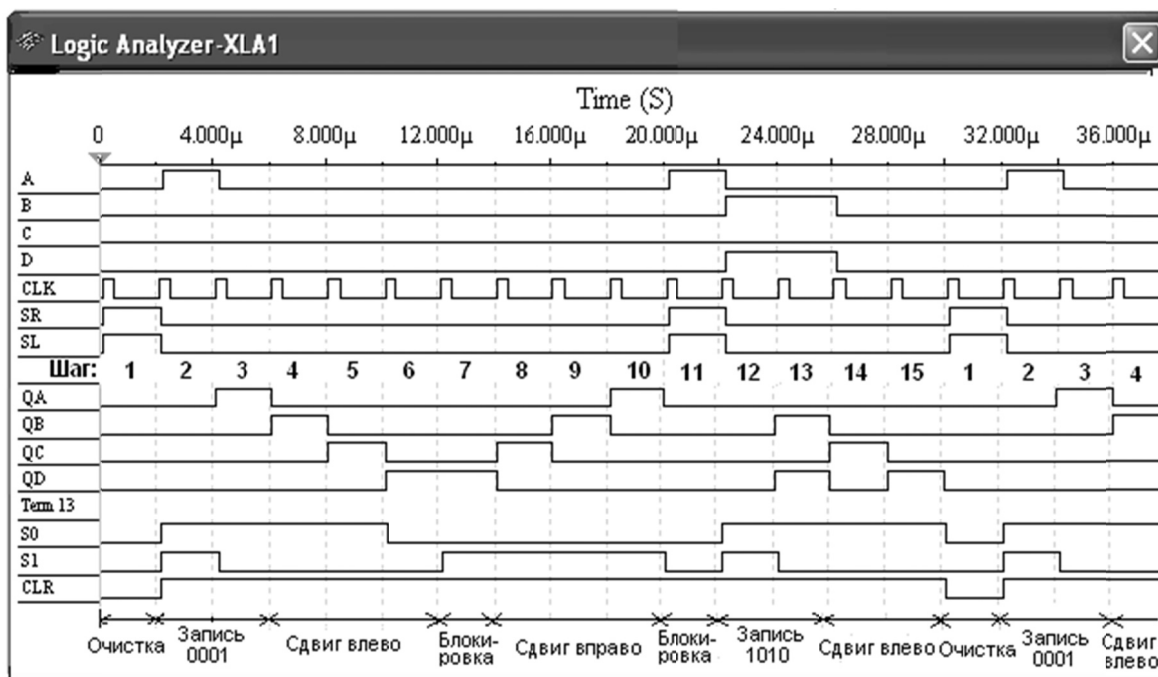


Рис. 4.6. Моделирование параллельного регистра в виде временных диаграмм сигналов

Режим блокировки реализуется при подаче на оба управляющих входа сигналов низкого уровня, т. е. $S0 = S1 = 0$ при $\overline{CLR} = 1$ (см. шаг 7 и шаг 11 на рис. 4.6). В режиме блокировки данные в регистре не сдвигаются ни вправо, ни влево, а остаются на своих прежних позициях. При установке сигналов $S0 = 0$ и $S1 = 1$ с приходом 8, 9 и 10 тактовых импульсов происходит сдвиг сигнала 1000 вправо и его полный вывод из регистра. Если при работе регистра в режиме сдвига вправо (см. шаги 8, 9, 10 на рис. 4.6) в ячейки памяти генератора внести $SR = 1$, то сигнал 1 будет формироваться на выходе QD и сдвигаться вправо от QD к QA при каждом тактовом импульсе. И, как следствие, после десятого импульса на выходе установится сигнал 1111.

При установке $S0 = S1 = 0$ с приходом одиннадцатого импульса происходит блокировка выходов, на следующем шаге выполняется параллельная запись числа $DCBA = 1010$ в регистр, далее сдвиг данных влево и т. д.

Задание 3. Запустить среду разработки Multisim и собрать на рабочем поле среды Multisim схему для испытания *последовательного регистра сдвига* (рис. 4.7), установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему в отчет.

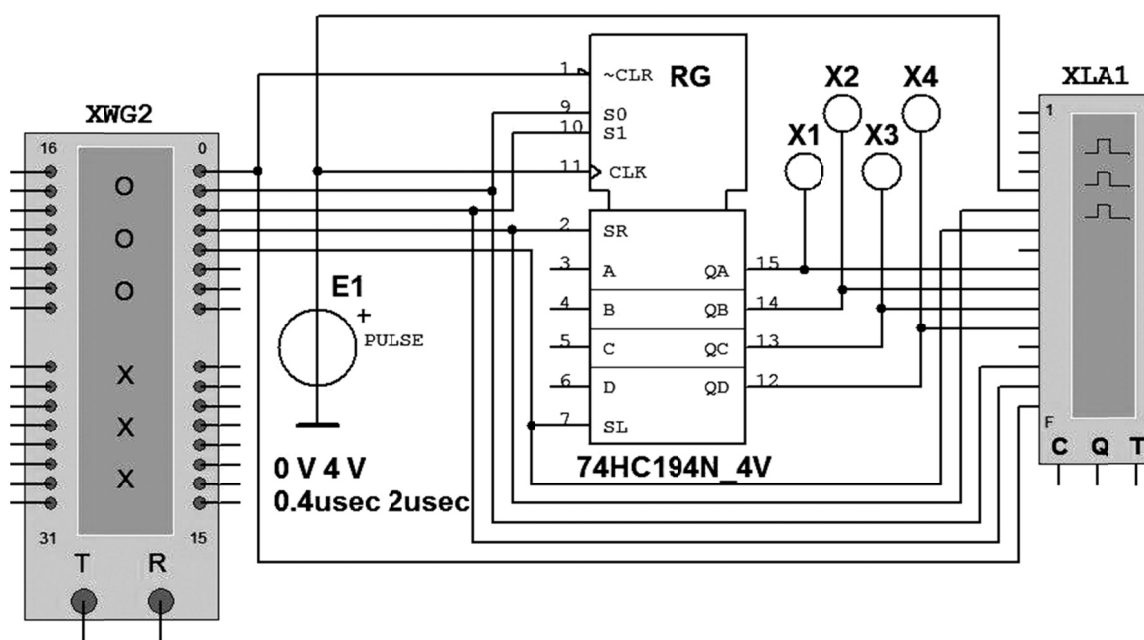


Рис. 4.7. Схема для испытания последовательного регистра сдвига

Чтобы микросхема 74HC194N_4V работала в качестве *последовательного регистра сдвига влево*, нужно подать на управляющий вход S0 высокий уровень напряжения, а на вход S1 – низкий уровень, т. е. установить $S0 = 1$ и $S1 = 0$, и подавать в последовательной

форме на вход SR данные, например 1, 0, 1 и 0, которые записываются в разряд A и передаются на выход QA (рис. 4.8). Регистр последовательно сдвигает влево эти сигналы от QA к QD, на выходе QD они теряются (см. шаги 3, ..., 9 на рис. 4.8).

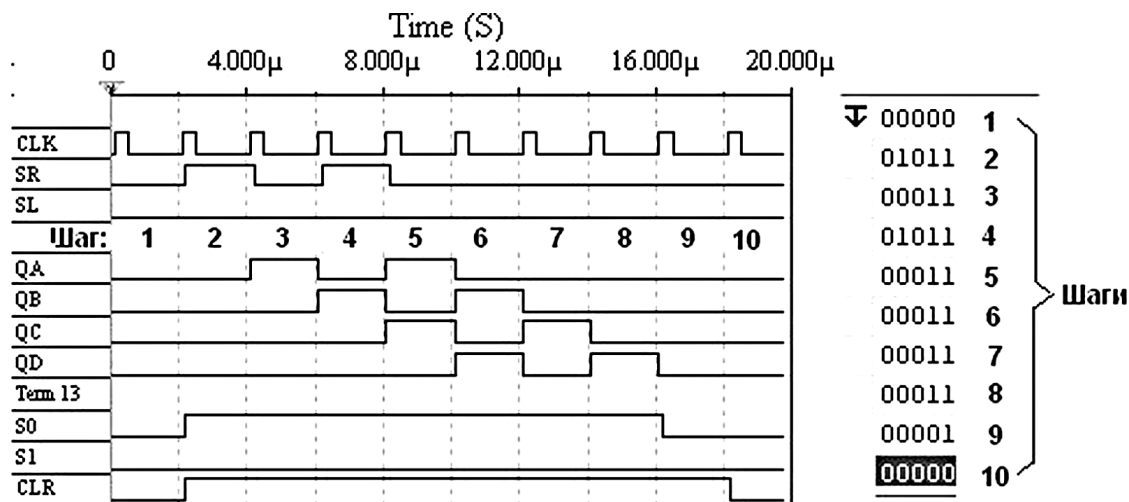


Рис. 4.8. Моделирование последовательного регистра сдвига влево в виде временных диаграмм сигналов

При установке $S1 = 0$ и $S1 = 1$ и подаче на вход SL данных в последовательной форме, например 1, 0, 0 и 1, которые записываются в разряд D (и передаются на выход QD), микросхема работает в режиме *последовательного регистра сдвига вправо* (без кольцевого перемещения данных): сигналы 1, 0, 0 и 1 сдвигаются по направлению к разряду A, на выходе QA они теряются (см. шаги 3, ..., 9 на рис. 4.9).

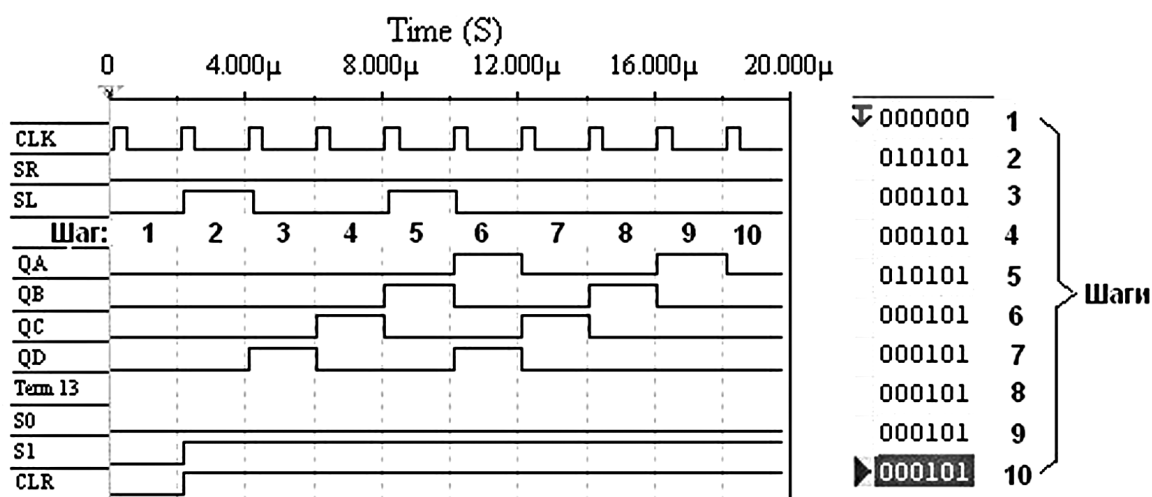


Рис. 4.9. Моделирование последовательного регистра сдвига вправо в виде временных диаграмм сигналов

Задание 4. Составить план исследования последовательного регистра 74НС194_4V, заполнив ячейки памяти генератора XWG1 произвольными (или по заданию преподавателя) 4-разрядными кодовыми комбинациями, вводимыми последовательно сперва в регистр А, а затем в регистр D.

Запустить программу моделирования последовательного регистра, скопировать в отчет временные диаграммы сигналов на входах и выходах регистра при сдвиге данных влево (см. рис. 4.8) и вправо (см. рис. 4.9).

Содержание отчета

1. Название и цель работы.
2. Перечень приборов, использованных в экспериментах, с их краткими характеристиками.
3. Изображения электрических схем для испытания параллельного и последовательного регистров.
4. Копии временных диаграмм, отображающих работу исследуемых регистров.
5. Выводы по работе.

ЗАПОМИНАЮЩИЕ УЗЛЫ. СЧЕТЧИКИ

5.1. Классификация счетчиков

Счетчик предназначен для счета поступающих на его вход импульсов, в интервале между которыми он должен хранить информацию об их количестве. Поэтому счетчик состоит из запоминающих ячеек – триггеров обычно *D*- или *JK*-типа. Между собой ячейки счетчика соединяют таким образом, чтобы каждому числу импульсов соответствовали состояния 1 или 0 определенных ячеек. При этом совокупность единиц и нулей на выходах *n* ячеек, называемых *разрядами* счетчика, представляет собой *n*-разрядное двоичное число, которое однозначно определяет количество прошедших через входы импульсов.

Каждый разряд счетчика может находиться в двух состояниях. Число устойчивых состояний, которое может принимать данный счетчик, называют **коэффициентом пересчета** $K_{сч}$.

Если с каждым входным импульсом «записанное» в счетчике число увеличивается, то такой счетчик является *суммирующим*, если же оно уменьшается – то *вычитающим*. Счетчик, работающий как на сложение, так и на вычитание, называют *реверсивным*.

Счетчики, у которых под воздействием входного импульса переключение соответствующих разрядов происходит последовательно друг за другом, называют *асинхронными*, а когда переключение происходит одновременно – *синхронными*. Максимальное число *n*, которое может быть записано в счетчике, равно $(2^n - 1)$, где *n* – число разрядов счетчика.

По способу кодирования последовательных состояний различают *двоичные счетчики* с коэффициентами пересчета (обнуления) $K_{сч} = 2^n$, у которых порядок смены состояний триггеров соответствует последовательности двоичных чисел, и *недвоичные*, у которых $K_{сч} < 2^n$ (например, десятичные с коэффициентом $K_{сч} = 10$ или делители частоты с коэффициентом деления $K_{сч} \neq 2^n$).

Счетчики входят в состав разнообразных цифровых устройств: электронных часов, делителей частоты, распределителей импульсов, вычислительных и управляющих устройств. Выпускаемые промышленностью интегральные счетчики представляют собой схемы средней интеграции (например, микросхемы серий К155, К176 и др.); среди них много разрядные бинарные счетчики на сложение и реверсивные счетчики с установочными входами R и S для всех разрядов, с постоянными и произвольными коэффициентами пересчета.

5.2. Счетчик с непосредственными связями

Условное изображение трехразрядного суммирующего счетчика показано на рис. 5.1, *а*, на котором символом R обозначен вход общего сброса, символами Q_1 , Q_2 и Q_3 – выходы счетчика, CR – выход переноса единицы. Суммирующий вход счетчика обозначается $+1$, вычитающий – (-1) . Это счетные входы. У асинхронных счетчиков эти входы помечены специальными символами: \triangleright или \triangleleft , указывающими полярность перепада входного сигнала: $1/0$ или $0/1$, при которой происходит переключение триггеров счетчика.

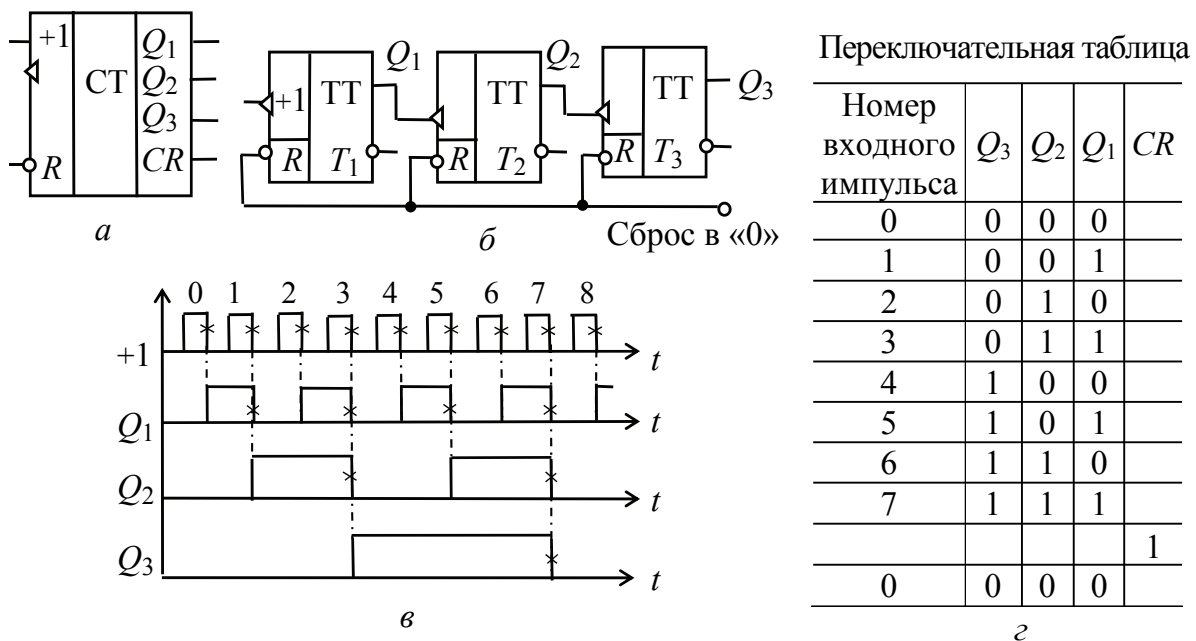


Рис. 5.1. Трехразрядный суммирующий счетчик: *а* – условное графическое изображение; *б* – схема счетчика с непосредственными связями; *в* – временные диаграммы; *г* – переключательная таблица

Для переключения триггеров в счетчиках используют следующие связи: непосредственную, тракт последовательного переноса, тракт параллельного переноса. Схема счетчика с непосредственными связями показана на рис. 5.1, б. Первый триггер счетчика T_1 образует младший разряд. Он пересчитывает входные импульсы по модулю 2, а состояние его выхода воспринимается следующим T_2 триггером как входные сигналы и снова пересчитываются на 2 и т. д.

Полное представление о состояниях счетчика (рис. 5.1, б) в зависимости от числа поданных на вход импульсов дают переключательная таблица (рис. 5.1, г) и временные диаграммы (рис. 5.1, в), на которых изображены последовательность входных импульсов (на входе +1), а также состояния триггеров – первого (Q_1), второго (Q_2) и третьего (Q_3). Фронты импульсов на диаграммах показаны идеальными: потенциал, соответствующий логическому 0, считается равным нулю, переключаящиеся перепады для наглядности помечены крестиками.

Рассмотрим воздействие на счетчик, к примеру, шестого (обозначенного на диаграмме цифрой 5) импульса. По его спаду триггер T_1 устанавливается в 0, перепад 1/0 на его выходе Q_1 переключает в 1 триггер T_2 , а триггер T_3 остается в прежнем (единичном) состоянии, так как перепад 0/1 на выходе Q_2 не является для него переключаящим.

Из диаграммы видно, что частота импульсов на выходе каждого триггера вдвое меньше частоты импульсов на его входе. В момент, предшествующий переключению очередного разряда, все предыдущие разряды счетчика находятся в состоянии 1. Восьмой импульс для трехразрядного счетчика (рис. 5.1, г) является импульсом переполнения: им все триггеры устанавливаются в 0 (счетчик «обнуляется»).

Если в счетчике используются триггеры, переключаящиеся перепадом 0/1, то вход последующего триггера нужно соединить с инверсным выходом предыдущего, на котором формируется этот перепад, когда по основному выходу триггер переключается из 1 в 0.

Схема *вычитающего* счетчика приведена на рис. 5.2. Согласно ей, по входам S в разряды счетчика заносят двоичное число, из которого нужно вычесть число, представляемое количеством входных импульсов. Пусть, например, в счетчик (рис. 5.2) занесено число $5_{10} = 101_2$. Первым входным импульсом триггер T_1 переключится из 1 в 0 (по основному выходу); при этом на инверсном выходе \bar{Q}_1 возникает перепад 0/1, которым триггер T_2 переключиться не может; в счетчике останется число $100_2 = 4_{10}$.

Второй входной импульс устанавливает триггер T_1 в состояние 1, на выходе \bar{Q}_1 появляется перепад 1/0, который переключает T_2 в состояние 1, а формирующийся при этом на \bar{Q}_2 перепад 1/0 переключает T_3 в состояние 0. В счетчике остается число $011_2 = 3_{10}$. Аналогично можно рассмотреть действие последующих входных импульсов.

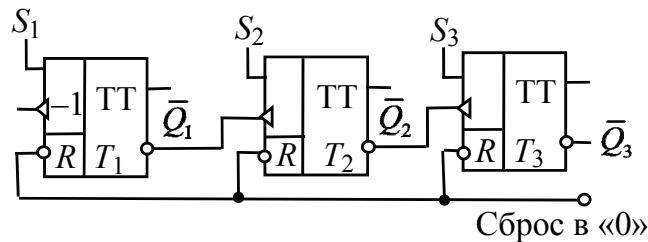


Рис. 5.2. Схема вычитающего счетчика

В счетчике с непосредственной связью переключение триггеров, вызванное срезом входного сигнала, происходит один за другим, *последовательно*, и задержка распространения n -разрядного счетчика, оцениваемая задержкой самого худшего случая, – сменой всех 1 на все 0, – в n раз больше задержки одного T -триггера. Если разрядов много, то большая задержка может оказаться серьезным недостатком такого счетчика. Из-за невозможности выполнить смену состояния всего счетчика в единый момент времени, счетчики с непосредственной связью бывают только *асинхронными*, т. е. сигналом, переключающим их, является сам входной сигнал.

5.3. Суммирующий синхронный счетчик

В *синхронном* счетчике переключающиеся разряды переходят в новое состояние одновременно (синхронно). Для того чтобы на входы всех разрядов каждый счетный импульс поступал одновременно, а переключение разрядов происходило в нужной последовательности, в схему добавляют логические цепи, которые обеспечивают переключение одних разрядов, а другие удерживают от переключения.

В схеме (рис. 5.3) четырехразрядного синхронного счетчика на JK -триггерах на тактовые входы C всех триггеров счетные импульсы поступают одновременно со входа T . Информационные входы J и K каждого триггера объединены. Триггер T_1 переключается каж-

мым счетным импульсом, так как на его входы J и K постоянно подается 1. Остальные триггеры переключаются счетными импульсами при следующих условиях: T_2 – при $Q_1 = 1$; T_3 – при $Q_1 = 1$ и $Q_2 = 1$; T_4 – при $Q_1 = 1$, $Q_2 = 1$ и $Q_3 = 1$.

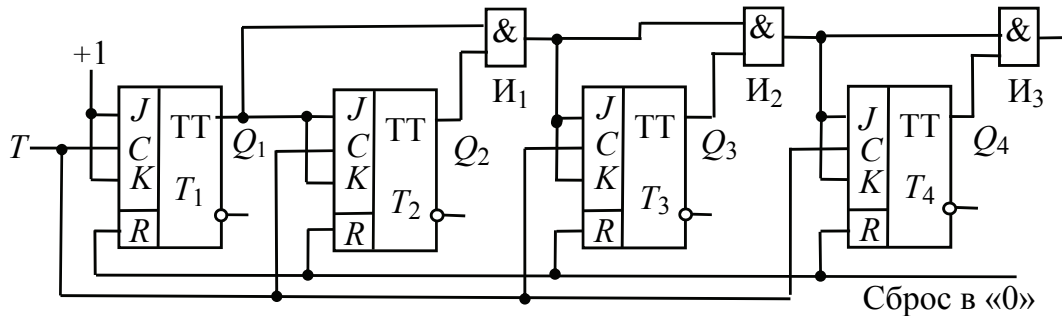


Рис. 5.3. Четырехразрядный синхронный счетчик на JK -триггерах

Для обеспечения указанных условий переключения триггеров в схему (рис. 5.3) добавлены конъюнкторы $И_1$, $И_2$ и $И_3$. На информационный вход каждого из триггеров T_2 , T_3 и T_4 подается конъюнкция сигналов с основных выходов предыдущих триггеров. Разрешающая переключение единица поступит на вход соответствующего триггера, если все предыдущие находятся в состоянии 1, и по счетному сигналу он переключается.

5.4. Реверсивный синхронный счетчик

Реверсивный счетчик, фрагмент которого изображен на рис. 5.4, работает как на сложение, так и на вычитание.

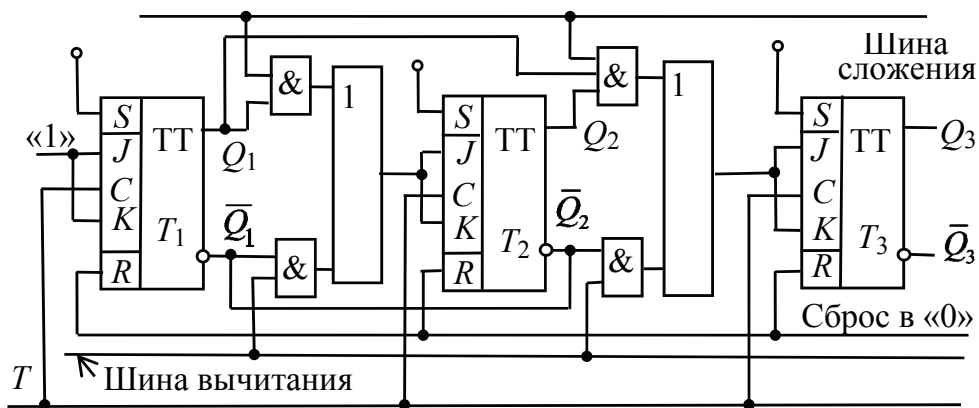


Рис. 5.4. Фрагмент реверсивного счетчика

Для перехода от сложения к вычитанию и обратно изменяют подключение входа последующего триггера к выходам предыдущего.

На объединенные входы J и K каждого триггера подается через дизъюнктеры конъюнкция сигналов с выходов предыдущих триггеров: основные выходы предыдущих триггеров присоединяются через конъюнктеры верхнего ряда (при сложении), а инверсные выходы – через конъюнктеры нижнего ряда (при вычитании). При сложении подают 1 на шину сложения, которой вводятся в действие конъюнктеры верхнего ряда; при этом на шине вычитания присутствует 0, вследствие чего конъюнктеры нижнего ряда выключены. Вычитание осуществляется при подаче 1 на шину вычитания и 0 – на шину сложения. Счетные импульсы поступают на вход T .

Как отмечалось в п. 5.2, каждый триггер переключается по тактовому входу C при $J = K = 1$, что имеет место, когда на выходах всех предыдущих триггеров (на основных – при сложении, на инверсных – при вычитании) будут единицы. Функционирование счетчика при сложении и вычитании описано в п. 5.2 и в 5.3.

5.5. Десятичный счетчик

Наибольшее распространение среди недвоичных счетчиков, у которых коэффициент пересчета $K_{сч} < 2^n$, имеют десятичные счетчики, у которых $K_{сч} = 10$. При проектировании недвоичного счетчика вначале определяют количество его разрядов n так, чтобы 2^n было бóльшим ближайшим к $K_{сч}$ числом. Затем тем или иным способом (например, принудительной установкой некоторых разрядов счетчика в 1) исключают избыточные состояния счетчика, число которых равно $2^n - K_{сч}$.

Так, для получения $K_{сч} = 10$ одноразрядный счетчик должен содержать четыре триггера, а избыточные состояния $2^n - K_{сч} = 16 - 10 = 6$ исключают тем или иным способом. При проектировании десятичного счетчика чаще используют двоично-десятичное кодирование чисел. В этой системе, например, число 375 записывается как 0011 0111 0101, где сохранены позиции десятичных разрядов: $0011_2 = 3_{10}$, $0111_2 = 7_{10}$, $0101_2 = 5_{10}$.

В связи с этим десятичный счетчик должен состоять из последовательно соединенных декад, информация о каждом из девяти импульсов накапливается в декаде, а десятым импульсом она обну-

ляется, и единица переносится в следующую декаду. Каждая декада работает в натуральном двоичном коде с весами двоичных разрядов, начиная со старшего, соответственно равными 8, 4, 2, 1, т. е. декада работает в коде 8-4-2-1. Если к выводам декад подключить индикаторы, то они будут показывать записанные числа в декадах в привычном десятичном коде. Десятичные счетчики выполняются и с другими весами разрядов, например, в коде 4-2-2-1.

На рис. 5.5 изображена функциональная схема десятичного счетчика с параллельным переносом на *JK*-триггерах с встроенными логическими элементами, реализующая переключательные функции:

$$J_1 = 1; J_2 = Q_1 \bar{Q}_4; J_3 = Q_1 \bar{Q}_2; J_4 = Q_1 Q_2 Q_3;$$

$$K_1 = 1; K_2 = Q_1; K_3 = Q_1 Q_2; K_4 = Q_1.$$

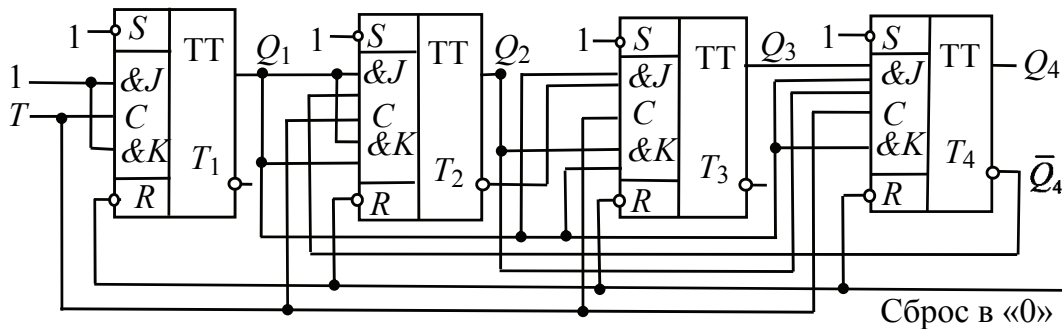


Рис. 5.5. Функциональная схема десятичного счетчика с параллельным переносом на *JK*-триггерах

Рассмотрим работу схемы. Пусть по тактовому входу T на триггер T_1 поступило семь импульсов и показание счетчика 0111. При этом на входах K триггеров T_1 , T_2 и T_3 будут логические единицы. Восьмой импульс вызовет переключение всех триггеров счетчика, т. е. в нем будет записан код 1000. Девятый импульс вызовет переключение только первого триггера, так как остальные триггеры заблокированы по входу J уровнями логического нуля с триггеров T_1 , T_2 и T_3 соответственно. Показание счетчика будет 1001. Десятый входной импульс вызовет переключение триггеров T_1 и T_4 , так как триггеры T_2 и T_3 заблокированы по входу J уровнями 0 с выходов соответствующих триггеров. Счетчик зафиксирует двоичный код 0000, т. е. установится в исходное состояние.

Уменьшение числа устойчивых состояний в счетчике прямого счета достигнуто за счет введения обратных связей, посредством

которых сигнал с какого-либо старшего разряда поступает в младшие, обеспечивая при этом изменение естественной последовательности двоичных чисел при подсчете входных импульсов. Этим способом можно строить счетчики с заданным коэффициентом пересчета.

Лабораторная работа № 6

Задание 1. Запустить среду разработки Multisim, собрать на рабочем поле среды Multisim схему для испытания *синхронного двоичного счетчика* (рис. 5.6), установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

В библиотеке программной среды Multisim имеются 4-, 8- и 12-разрядные счетчики различных типов. Среди них: счетчики асинхронные (SN7493, SN74393), синхронные (74NC161, SN74163), реверсивные (SN74191), двоично-десятичные (SN7493, SN74160) и др.

В схему (рис. 5.6) включен синхронный двоичный 4-разрядный счетчик 74HC161, к входу $\overline{\text{CLK}}$ которого подключен источник тактовых импульсов E1, а к выходам QA, QB, QC и QD – шестнадцатеричный 7-сегментный индикатор DCD_HEX и дешифратор DC 4×10. Выход дешифратора соединен с входами логического анализатора XLA1.

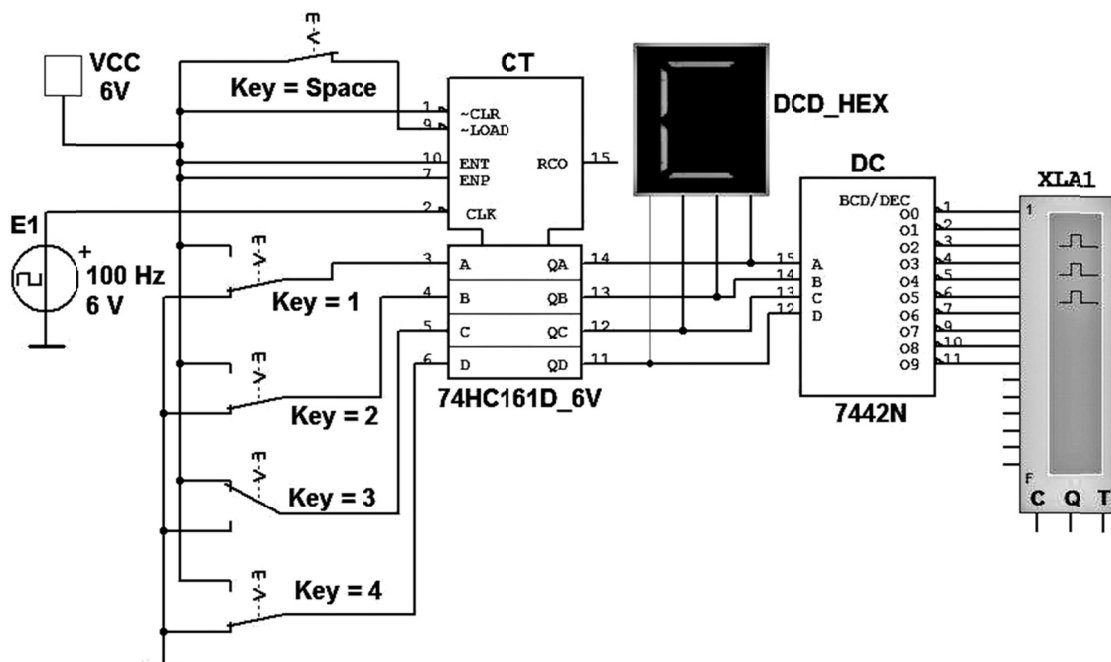


Рис. 5.6. Схема для испытания синхронного двоичного счетчика

Ко входам **A**, **B**, **C** и **D** счетчика **СТ** подключены источник постоянного напряжения **VCC**, переключатели **1**, ..., **4** для формирования входных двоичных кодов и ключ **Space** для изменения режима работы счетчика. В синхронном счетчике заданные с помощью ключей уровни сигналов подаются на входы всех триггеров, как и тактовые импульсы, которые подаются на счетные входы \overline{CLK} всех разрядов счетчика.

При замкнутом ключе **Space** число поданных от генератора **E1** на вход счетчика импульсов высвечивается на индикаторе **DCD_HEX** в десятичном коде, от 0 до 15, после чего счетчик обнуляется и вновь начинается счет. При этом на одном из выходов дешифратора **DC** формируется сигнал низкого уровня (логический 0), номер которого соответствует коду входного числа: от 0000 до 1001 (9_{10}).

При разомкнутом ключе **Space** сформированное с помощью переключателей на входе счетчика 4-разрядное двоичное число высвечивается на индикаторе в десятичном коде, а на экране анализатора на одном из выходов, соответствующем входному коду счетчика, формируется логический 0.

Задание 2. Замкнуть ключ **Space**, запустить программу моделирования суммирующего счетчика и наблюдать за показаниями индикатора. Убедиться, что на экране анализатора **XLA1** логические нули перестают формироваться после прихода одиннадцатого тактового импульса и появляются вновь только с приходом семнадцатого импульса (рис. 5.7).

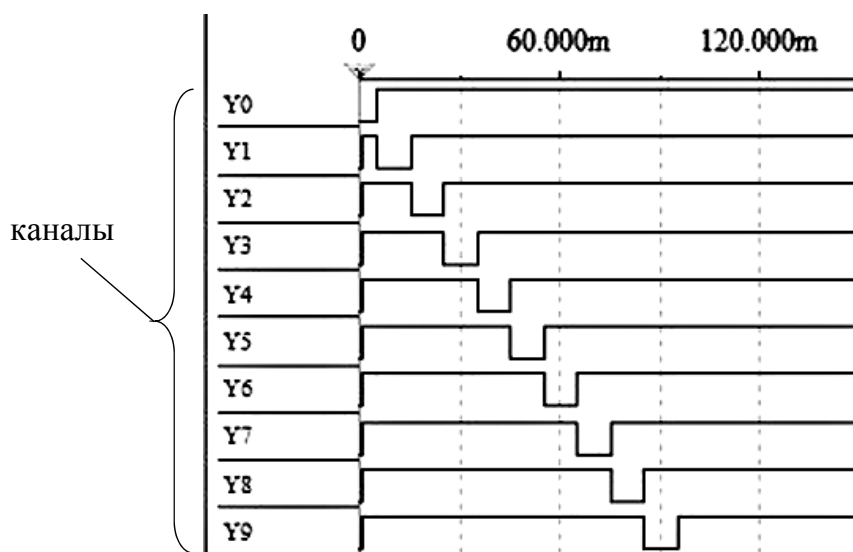


Рис. 5.7. Результаты моделирования синхронного суммирующего счетчика

Разомкнуть ключ **Space**. Установить в диалоговом окне анализатора **XLA1** напряжение $V = 5$ В, частоту таймера $f_a = 2$ кГц, число импульсов, приходящихся на одно деление, **Clocks/div** = 60. (При таком режиме лучи медленно перемещаются на экране анализатора.) С помощью активных клавиш 1, 2, 3 и 4 клавиатуры сформировать произвольные (или по указанию преподавателя) двоичные входные числа (коды), например 1001, 0011, 0000, 1110, и подавать их на входы **D**, **C**, **B** и **A** счетчика. Убедиться, что при подаче числа 1110_2 (14_{10}) ни на одном выходе дешифратора 4×10 не сформировался низкий уровень сигнала.

Скопировать на страницу отчета результаты моделирования синхронного суммирующего счетчика (см. рис. 5.7).

Задание 3. Собрать на рабочем поле среды Multisim схему для испытания *реверсивного двоичного счетчика* (рис. 5.8) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

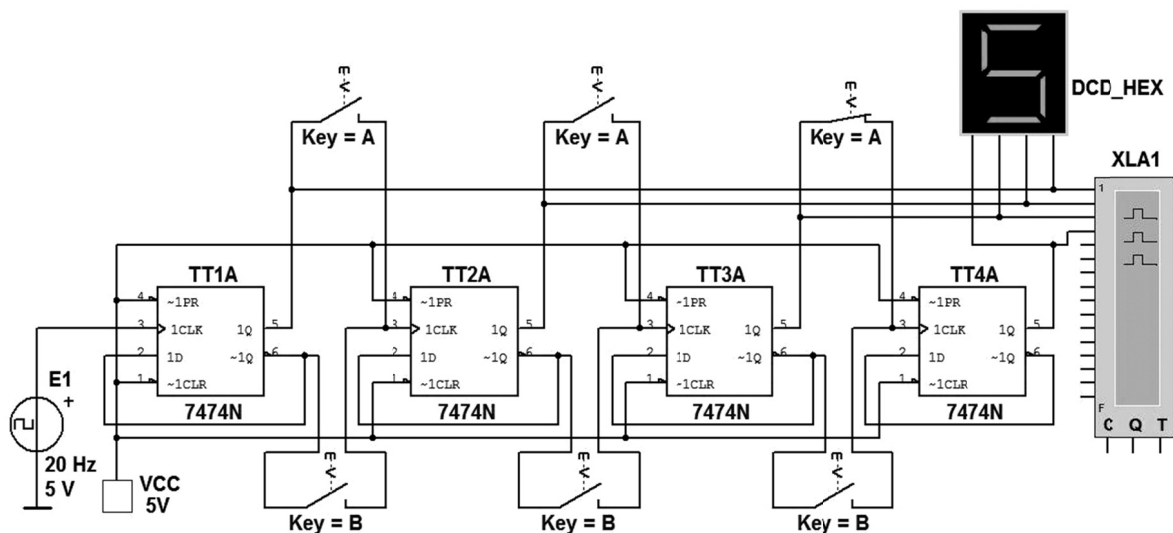


Рис. 5.8. Схема для испытания реверсивного двоичного счетчика

В схеме реверсивного двоичного счетчика (рис. 5.8) с помощью групп ключей **A** и **B** осуществляется замыкание или размыкание выходов высокого либо низкого уровня предыдущего триггера со входами $1PR$ или $1CLR$ триггера следующего разряда, причем при замкнутых ключах **A** и разомкнутых **B** (режим суммирования) с каждым тактовым импульсом увеличивается результат счета, а при замкнутых ключах **B** и разомкнутых **A** (режим вычитания) – результат счета уменьшается.

Установить в диалоговом окне анализатора **XLA1** напряжение $V = 5 \text{ В}$, частоту таймера $f_a = 2 \text{ кГц}$, число импульсов, приходящихся на одно деление, **Clocks/div** = 60. Разомкнуть ключи **В** и замкнуть ключи **А**. Запустить программу моделирования счетчика.

При высвечивании числа 15 на 7-сегментном индикаторе щелкнуть мышью по кнопке **Stop** (остановки моделирования) и скопировать окно анализатора с результатами моделирования на страницу отчета (см. верхнюю часть рис. 5.9).

Разомкнуть ключи **А** и замкнуть ключи **В**. Щелкнуть мышью по кнопке **Stop** (продолжить моделирование), остановить моделирование при высвечивании числа 0 на индикаторе и скопировать окно анализатора с результатами моделирования в отчет (см. нижнюю часть рис. 5.9).

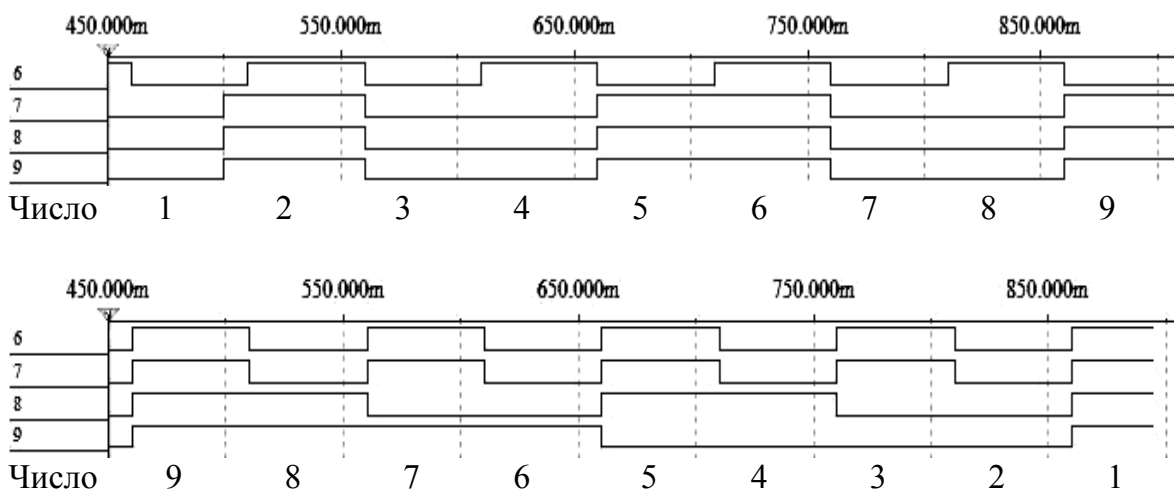


Рис. 5.9. Результаты моделирования реверсивного двоичного счетчика

Задание 4. Собрать на рабочем поле среды Multisim схему для испытания *десятичного счетчика* (рис. 5.10) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

Результаты моделирования 4-разрядного двоичного счетчика (см. рис. 5.8 и рис. 5.9) показали, что с его помощью можно сосчитать до 15.

Следовательно, для создания счетчика натуральных десятичных чисел в двоичном коде для одной декады нужно в схему двоичного 4-разрядного счетчика ввести обратные связи с логическими элементами, посредством которых сигнал с какого-то старшего разряда поступает в младшие и т. п., обеспечивая в целом выработку счетчиком двоичного эквивалента счетной декады.

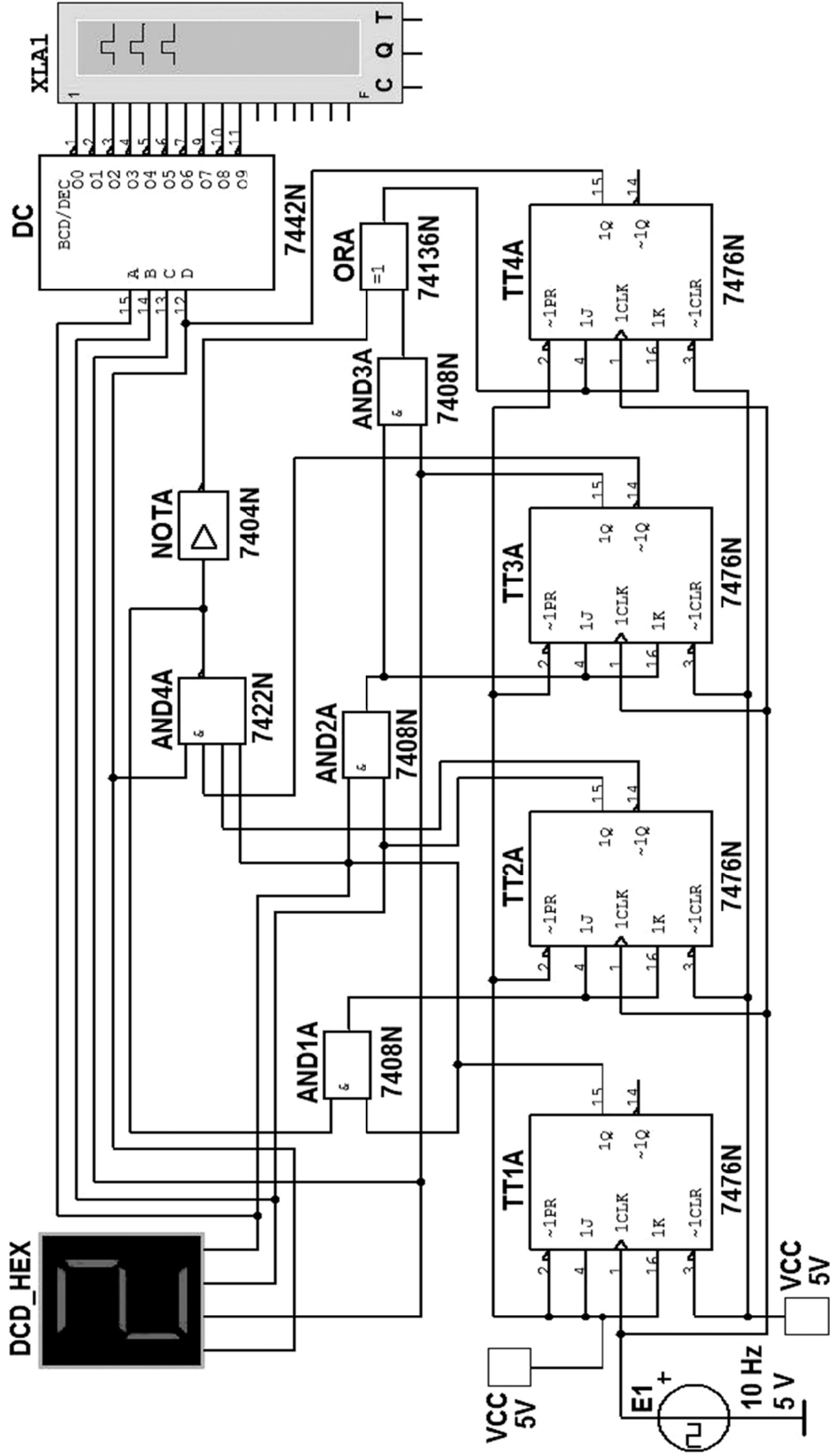


Рис. 5.10. Схема для испытания десятичного счетчика

В функциональной схеме десятичного счетчика (5.10), собранной на триггерах *JK*-типа, на каждый одиннадцатый тактовый импульс результат счета сбрасывается в нуль и далее результат счета увеличивается. Возврат счетчика при поступлении одиннадцатого тактового импульса в начальное состояние обеспечивается дополнительной комбинационной схемой со встроенными логическими элементами И (**AND**), ИЛИ (**OR**) и НЕ (**NOT**).

Счетчик работает так же, как синхронный двоичный счетчик до поступления седьмого импульса, а далее, благодаря обратной связи, нарушается изменение естественной последовательности двоичных чисел на входах и выходах триггеров при подсчете тактовых импульсов.

Запустить программу моделирования десятичного счетчика и скопировать окно анализатора с результатами моделирования на страницу отчета.

Содержание отчета

1. Название и цель работы.
2. Перечень приборов, использованных в экспериментах, с их краткими характеристиками.
3. Изображения электрических схем для испытания суммирующего, реверсивного и десятичного счетчиков.
4. Копии временных диаграмм, отображающие работу исследуемых счетчиков.
5. Выводы по работе.

ЦИФРО-АНАЛОГОВЫЙ ПРЕОБРАЗОВАТЕЛЬ

6.1. Структура резистивных матриц ЦАП

При построении устройств, связывающих цифровое устройство с объектами, использующими информацию в непрерывно изменяющейся форме, требуется преобразование информации из аналоговой формы в цифровую и из цифровой в аналоговую. Устройство, осуществляющее автоматическое преобразование непрерывно изменяющихся во времени аналоговых значений физической величины (напряжения, тока) в эквивалентные значения числовых кодов, называют *аналого-цифровым преобразователем* (АЦП). Устройство, осуществляющее автоматическое преобразование входных значений, представленных числовыми кодами, в эквивалентные им значения какой-нибудь физической величины (напряжения, тока и др.), называют *цифро-аналоговым преобразователем* (ЦАП).

Итак, цифро-аналоговый преобразователь предназначен для прямого преобразования входного двоичного кода, например $A_i(a_2a_1a_0)$, в аналоговый эквивалент. Выходная аналоговая величина, обычно напряжение $u_{\text{вых}}$, иногда нормированное $u_{\text{вых.н}} = u_{\text{вых}}/u_{\text{вых.мах}}$, соответствует кодовой комбинации A_i , поступившей на вход, и воспроизводится для дискретных моментов времени (рис. 6.1, а). Сменяющиеся входные цифровые коды обуславливают сменяющееся ступенчатое напряжение на выходе (L – идеальная передаточная характеристика ЦАП).

Существует два широко распространенных способа цифро-аналогового преобразования с использованием:

- резистивной матрицы с весовыми двоично-взвешенными сопротивлениями;
- резистивной матрицы с двумя номиналами сопротивлений, которую обычно называют матрицей $R-2R$.

ЦАП с весовыми двоично-взвешенными сопротивлениями (рис. 6.1, б) состоит из n переключателей S_i (по одному на каждый разряд), управляемых двоичным кодом A_i ; матрицы двоично-

взвешенных резисторов с сопротивлениями $2^{n-1}R$; источника опорного напряжения u_0 и выходного операционного усилителя ОУ, с помощью которого суммируются токи, протекающие через резисторы с двоично-взвешенными сопротивлениями, для получения аналогового выходного напряжения $u_{\text{ВЫХ}}$.

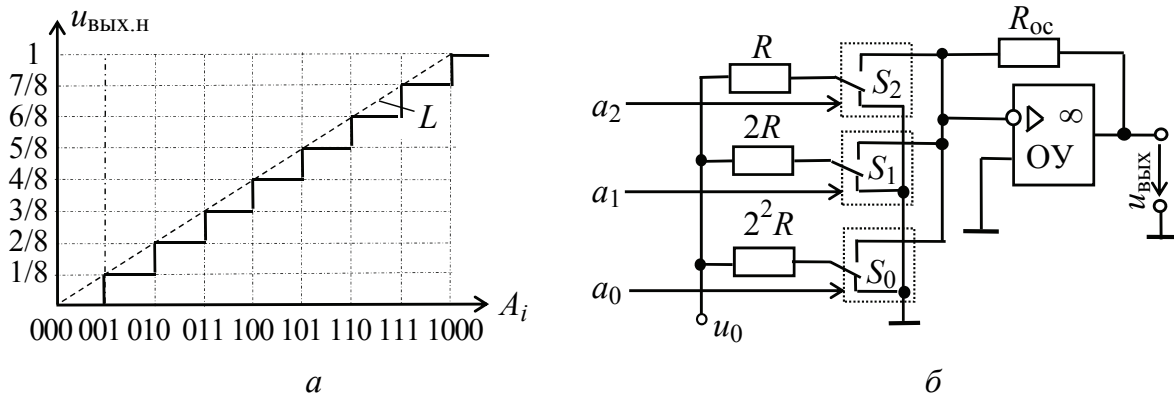


Рис. 6.1. ЦАП с весовыми двоично-взвешенными сопротивлениями: *a* – кодовые комбинации A_i , поступившие на вход ЦАП; *б* – структурная схема ЦАП с весовыми двоично-взвешенными сопротивлениями

Каждый i -тый разряд управляет переключателем S_i , который подключается к источнику опорного напряжения u_0 , когда $a_i = 1$, или к общей шине, когда $a_i = 0$. Сопротивления резисторов $2^{n-1}R$ (n – номер разряда входного кода), соединенных с ключами, таковы, что обеспечивают пропорциональность в них тока двоичному весу соответствующего разряда входного кода. Следовательно, ток на входе ОУ и выходное напряжение ЦАП будут (6.1)

$$i = \frac{a_{n-1}u_0}{R} + \frac{a_{n-2}u_0}{2R} + \dots + \frac{a_1u_0}{2^{n-1}R} + \frac{a_0u_0}{2^n R}; \quad (6.1)$$

$$u_{\text{ВЫХ}} = -R_{\text{OC}}i = -u_0 \frac{R_{\text{OC}}}{2^n R} \sum_{i=0}^{n-1} a_i 2^i.$$

Напряжение на выходе ЦАП пропорционально «весу» присутствующего на входах кода, а максимальное значение имеет место, когда все разряды примут значение 1, т. е (6.2):

$$u_{\text{max}} = \left| u_0 \frac{(2^n - 1)R_{\text{OC}}}{2^n R} \right|, \quad (6.2)$$

оно всегда меньше опорного напряжения на шаг квантования $u_0 R_{\text{OC}} / (2^n R)$.

Номиналы сопротивлений резисторов в младшем и старшем разрядах отличаются в 2^{n-1} раз и должны быть выдержаны с высокой точностью. Например, для 12-разрядного ЦАП использование в старшем разряде резистора с сопротивлением 10 кОм потребует включения в младший разряд преобразователя резистора с сопротивлением порядка 20 МОм. Широкий набор номиналов резисторов и требования их высокой точности, в особенности при значительном числе разрядов n входного кода, создают трудности при реализации ЦАП посредством интегральной технологии.

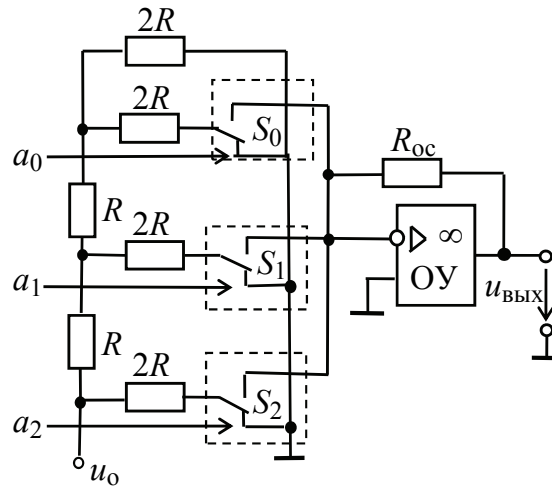


Рис. 6.2. ЦАП с матрицей $R-2R$

Во второй схеме ЦАП с матрицей $R-2R$ используют резисторы с двумя номиналами сопротивлений, причем резисторы с сопротивлением R включены в каждый разряд (см. рис. 6.2 при $n = 3$). Однако в этой схеме увеличиваются значения паразитных емкостей.

Принцип функционирования схемы основан на свойстве резистивного делителя $R-2R$ сохранять постоянное сопротивление нагрузки для источника опорного напряжения при замыкании ключей. Вследствие этого на выводах резистора R , начиная со старшего $n - 1$ разряда, опорное напряжение последовательно делится пополам, как и входящий в каждый узел матрицы ток. При этом напряжение (6.3) на выходе преобразователя с матрицей $R-2R$ следующее:

$$\begin{aligned}
 u_{\text{ВЫХ}} &= -u_0 \frac{R_{\text{oc}}}{R} (a_{n-1} 2^{-1} + a_{n-2} 2^{-2} + \dots + a_1 2^{-(n-1)} + a_0 2^{-n}) = \\
 &= -u_0 \frac{R_{\text{oc}}}{2^n R} \sum_{i=0}^{n-1} a_i 2^i.
 \end{aligned} \tag{6.3}$$

Таким образом, выходное напряжение ЦАП пропорционально сумме напряжений со своими весами, обусловленными переключателями, подключенными к источнику опорного напряжения u_0 .

Недостатком ЦАП с матрицей $R-2R$ является сильное влияние на точность преобразования нестабильности сопротивлений переключателей в замкнутом состоянии, что снижает временную и температурную стабильность характеристик ЦАП. Этот недостаток в значительной степени удается устранить в схемах код – напряжение, выполненных на базе полупроводниковой технологии с использованием тонкопленочных резисторов на кристалле и переключателей на КМДП-транзисторах, в которых нелинейность от $\pm 0,8\%$ до $\pm 0,003\%$ от опорного напряжения u_0 , время установления тока от 5 мкс до десятых долей микросекунд и менее, часто выходной диапазон напряжения ± 5 В. Опорное напряжение в схемах ЦАП может выбираться разной полярности или двуполярным.

6.2. Основные параметры ЦАП

Основными параметрами ЦАП являются число разрядов $n = 8, \dots, 24$ и *абсолютная разрешающая способность* – среднее значение минимального изменения сигнала на выходе ЦАП, обусловленное увеличением или уменьшением его кода на единицу. Теоретически ЦАП, преобразующий n -разрядные двоичные коды, должен обеспечить 2^n различных значений выходного сигнала с разрешающей способностью $1/(2^n - 1)$. При числе разрядов $n = 8$ количество независимых квантов (ступеней) выходного напряжения ЦАП равно $2^8 - 1 = 255$, при $n = 12$, $2^{12} - 1 = 4095$ и т. д.

Абсолютное значение минимального кванта напряжения определяется как предельным принимаемым числом $2^n - 1$, так и максимальным выходным напряжением ЦАП, по-другому называемым напряжением шкалы или опорным напряжением u_0 . Значение абсолютной разрешающей способности ЦАП, часто обозначаемое ЗМР (значение младшего разряда), при $n = 8$ и опорном напряжении $u_0 = 5$ В следующее:

$$\text{ЗМР} = \frac{u_0}{2^8 - 1} = \frac{5}{255} \approx 0,0196 \text{ В} = 19,6 \text{ мВ.}$$

Отличие реального значения разрешающей способности от теоретического обусловлено погрешностями и шумами входящих в ЦАП узлов. Точность ЦАП определяется значением абсолютной погрешности δ_a и нелинейностью преобразователя δ_n . *Абсолютная погрешность* δ_a характеризуется отклонением максимального значения выходного напряжения u_{\max} от расчетного, соответствующего конечной точке характеристики идеального преобразователя, и измеряется обычно в единицах ЗМР.

Нелинейность преобразователя δ_n характеризует отклонение действительной характеристики от линейной (от прямой линии L , см. рис. 6.1, *a*), проведенной через центры ступенек или через нуль и точку максимального значения выходного сигнала.

Из динамических параметров наиболее важным является максимальная частота преобразования f_{\max} (десятки и сотни кГц) – наибольшая частота дискретизации, при которой параметры ЦАП соответствуют заданным значениям.

Работа ЦАП часто сопровождается специфическими переходными импульсами в выходном сигнале, возникающими из-за разности времени открывания и закрывания аналоговых переключателей в ЦАП. Особенно значительно выбросы проявляются, если входной код 01...111 сменяется кодом 10...000, а переключатель старшего разряда ЦАП открывается позже, чем закрываются переключатели младших разрядов. Вследствие определенной идеализации при моделировании библиотечных ЦАП среды Multisim не всегда удается определить отмеченные выше параметры.

Библиотечные интегральные схемы ЦАП среды Multisim требуют для своей работы подключения только постоянного эталонного напряжения, заземления и входных сигналов.

Лабораторная работа № 7

Задание 1. Собрать на рабочем поле среды Multisim схему для испытания интегрального *цифро-аналогового преобразователя* (рис. 6.3) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

В схеме (рис. 6.3) использован библиотечный (**Mixed**) 8-разрядный цифро-аналоговый преобразователь **DAC**, на входы которого подаются сформированные с помощью переключателей **0, ..., 7** двоичные коды от 00000000 до 11111111_2 (FF_{16} или 255_{10}). Выходное напряжение ЦАП можно измерить с помощью вольтметра **V1** или осциллографа **XSC1**, воспользовавшись визирными линиями, расположенными на его экране.

Задание 2. Получить на экране осциллографа **XSC1** ступенчатое выходное напряжение ЦАП (рис. 6.4). Для этого нужно вначале замкнуть переключатель **0**, т. е. подать напряжение **5 В** на вход **D0** ЦАП, и запустить программу моделирования.

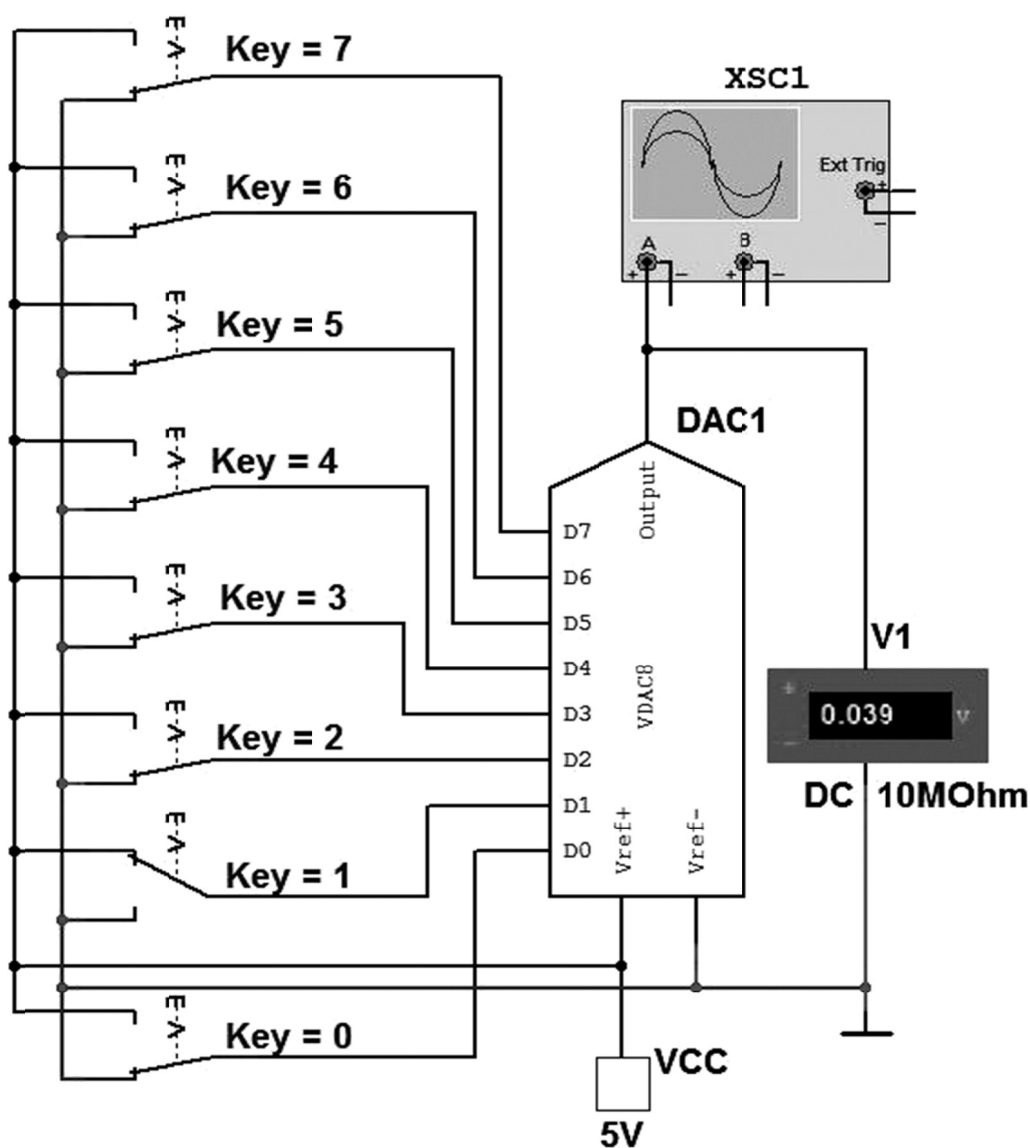


Рис. 6.3. Схема для испытания интегрального цифро-аналогового преобразователя

На выходе цифро-аналогового преобразователя формируется напряжение, равное ЗМР. Затем во время остановок моделирования нужно замыкать поочередно переключатели 1, 2, ..., 7, подавая входные десятичные комбинации 3, 7, 15, 31, 63, 127, 255 на входы D0, ..., D7 ЦАП.

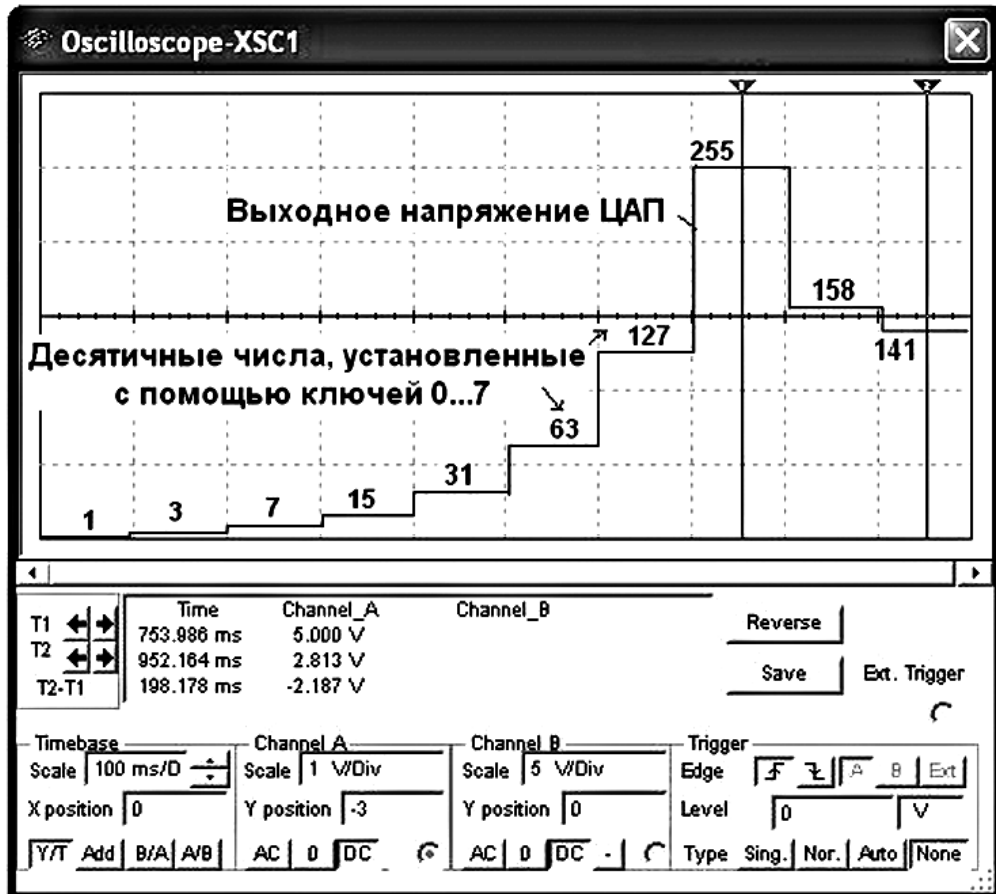


Рис. 6.4. Ступенчатое выходное напряжение ЦАП на экране осциллографа

Повторить эксперимент, подавая на входы ЦАП сформированные с помощью переключателей шестнадцатеричные коды от 0 до FF (255₁₀) через шаг 10₁₆ (16₁₀) и занося в таблицу показания вольтметра V1 (значения выходного напряжения $u_{\text{вых}}$ ЦАП) при напряжении источника VCC $u_0 = 5$ В. Найти частичные и усредненные значения ступени, частичные и усредненные значения ЗМР. Построить график $u_{\text{вых}}(N)$, выбрав соответствующие масштабы для напряжений и входных десятичных чисел N , откладываемых по осям координат.

Результаты эксперимента

| № п/п | Входной десятичный код N | Выходное напряжение, $u_{\text{вых}}$, В | Напряжение ступени $u_{\text{вых}2} - u_{\text{вых}1}$, В | Значение младшего разряда $MЗР = (u_{\text{вых}2} - u_{\text{вых}1})/16$, В |
|----------|----------------------------------|---|--|--|
| 1 | 0 | 0 | 0 | – |
| 2 | 15 | | | |
| 3 | 31 | | | |
| 4 | 47 | | | |
| 5 | 63 | | | |
| 6 | 79 | | | |
| 7 | 95 | | | |
| 8 | 111 | | | |
| 9 | 127 | | | |
| 10 | 143 | | | |
| 11 | 159 | | | |
| 12 | 175 | | | |
| 13 | 191 | | | |
| 14 | 207 | | | |
| 15 | 223 | | | |
| 16 | 239 | | | |
| 17 | 255 | | | |

Задание 3. Собрать на рабочем поле среды Multisim схему для испытания *цифро-аналогового преобразователя* (рис. 6.4, а) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

Провести моделирование ЦАП, запрограммировав генератор **XWG1** (частота генерации сигналов $f_r = 1$ кГц) на возрастание и убывание шестнадцатеричных чисел от 0 до FF (255_{10}) при шаге 10_{16} (16_{10}).

Составить таблицу и занести в нее выходные напряжения ЦАП и величину ступеней, которые выводятся в нижнем окне осциллографа **XSC2**.

Измерение напряжений проводить с помощью визирных линий осциллографа, устанавливая их на двух соседних ступенях (см. рис. 6.6) при различных кодовых комбинациях на выходе генератора **XWG1** и напряжении $u_0 = 5$ В источника **VCC**.

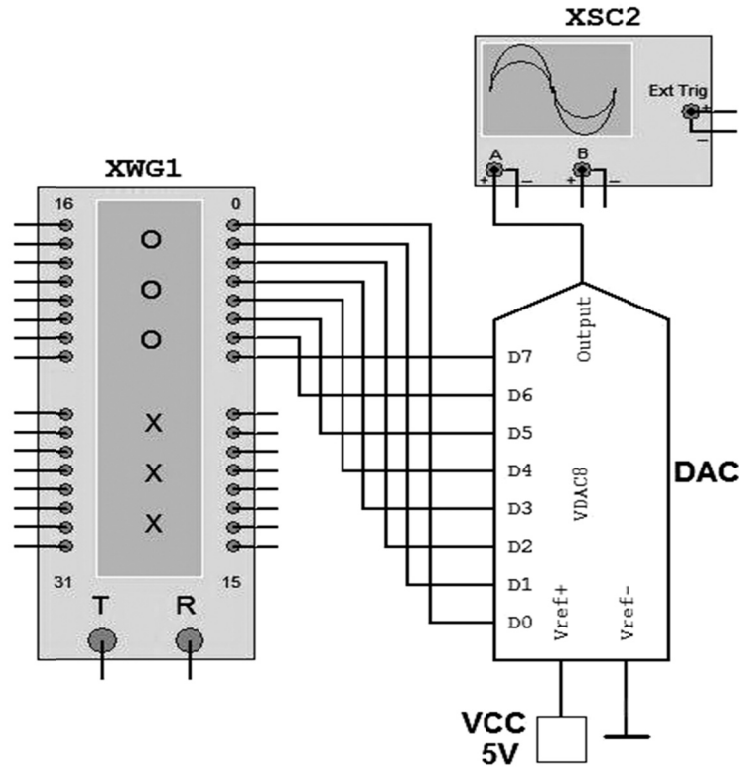


Рис. 6.5. Схема для испытания цифро-аналогового преобразователя

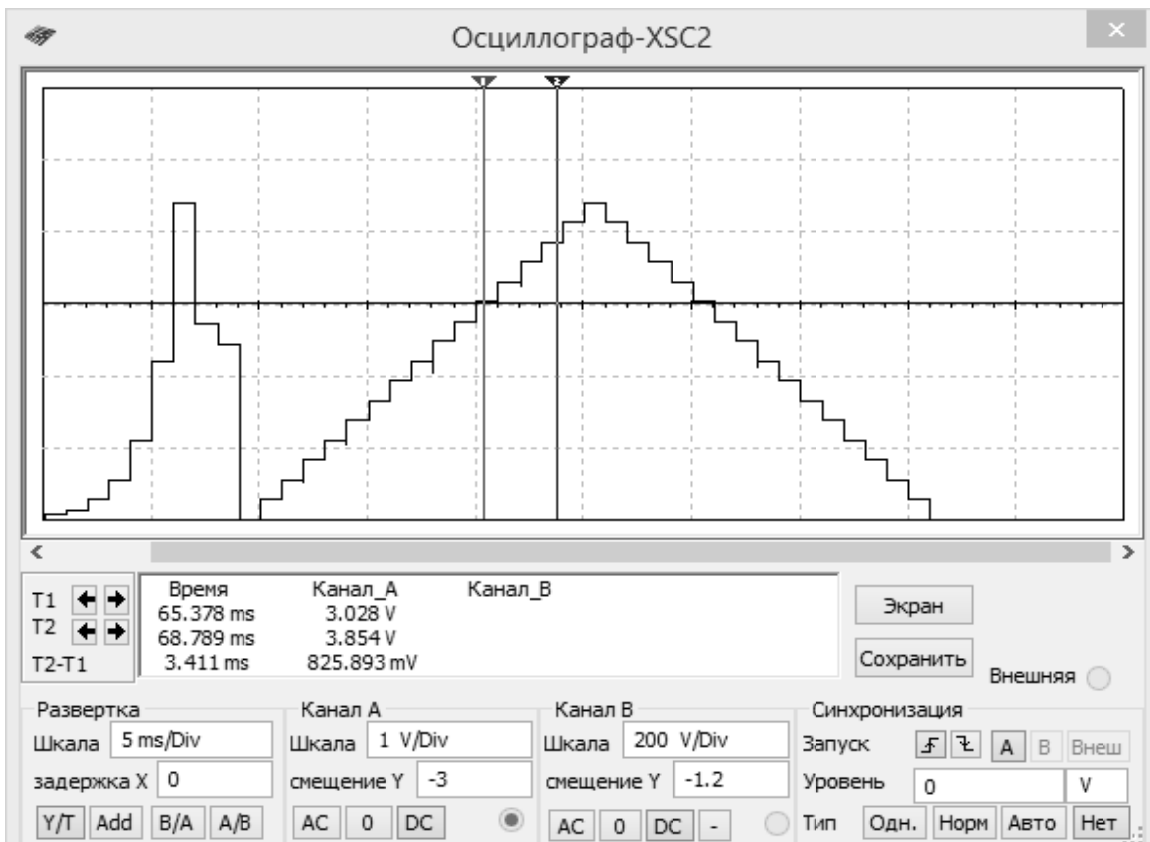


Рис. 6.6. Показания осциллографа

Так, при входных десятичных числах 175 и 191 и напряжении $u_0 = 5$ В выходные напряжения ЦАП соответственно равны 3,437 В и 3,750 В, а напряжение ступени – 312,5 мВ. При этом ЗМР = $= 312,5/16 = 19,53$ В. Найти и сравнить усредненное значение ЗМР с расчетным значением.

Установить напряжение $u_0 = 10$ В источника VСС и повторить моделирование ЦАП при опорном напряжении 10 В. Построить графики $u_{\text{вых}}(N)$ при $u_0 = 5$ В и $u_0 = 10$ В на одном рисунке, выбрав соответствующие масштабы для напряжений и входных десятичных чисел N , откладываемых по осям координат.

Содержание отчета

1. Название и цель работы.
2. Перечень приборов, использованных в экспериментах, с их краткими характеристиками.
3. Изображения электрических схем для испытания цифро-аналогового преобразователя.
4. Копии графиков выходного напряжения исследуемого ЦАП, отображающих его работу.
5. Графики $u_{\text{вых}}(N)$ при различных значениях опорного напряжения.
6. Выводы по работе.

АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ

7.1. Структурная схема АЦП последовательного действия

Аналого-цифровой преобразователь (АЦП) – устройство, предназначенное для преобразования аналоговых величин в их цифровой эквивалент в различных системах исчисления. Входным сигналом АЦП в течение некоторого промежутка времени Δt является постоянное напряжение, равное отсчету $u_{\text{вх}}(k\Delta t)$ входной аналоговой функции $u_{\text{вх}}$. За это время на выходе АЦП формируется цифровой (обычно двоичный) код, соответствующий дискретному отсчету напряжения $u_{\text{вх}}(k\Delta t)$. Количественная связь для любого момента времени определяется соотношением (7.1):

$$A_i = \frac{u_{\text{вх}}(k\Delta t)}{\Delta u \pm \delta_i}, \quad (7.1)$$

где Δu – шаг квантования входного аналогового напряжения $u_{\text{вх}}$; δ_i – погрешность преобразования напряжения $u_{\text{вх}}(k\Delta t)$ на данном шаге.

Физический процесс аналого-цифрового преобразования состоит из дискретизации по времени аналогового сигнала, квантования по уровню и кодирования. Процесс *дискретизации* аналогового сигнала длительностью $t_{\text{вх}}$ выполняется в соответствии с теоремой Котельникова, определяющей необходимый шаг дискретизации $\Delta t \leq 1/(2f_{\text{max}})$, где f_{max} – максимальная частота спектра входного сигнала, и число шагов $M = t_{\text{вх}}/\Delta t$.

Процесс *квантования по уровню* дискретизированной функции $u_{\text{вх}}(k\Delta t)$ заключается в отображении бесконечного множества ее значений на некоторое множество конечных значений $u_{\text{д}}(k)$, равное числу уровней квантования $N = u_{\text{вх max}}/\Delta u$. Процесс квантования по уровню (округление каждого значения $u_{\text{вх}}(k\Delta t)$ до ближайшего уровня $u_{\text{д}}(k)$) приводит к возникновению ошибки (шума) квантования, максимальное значение которой $\pm 1/2\Delta u$ определяется разрядностью

используемого выходного кода. При увеличении разрядности выходного кода ошибка квантования может быть уменьшена до сколь угодно малой величины, но не может быть сведена к нулю выбором параметров устройства, так как она присуща данному алгоритму.

Процесс *кодирования* заключается в замене найденных квантованных $N + 1$ значений входного сигнала $u_d(k)$ некоторыми цифровыми кодами.

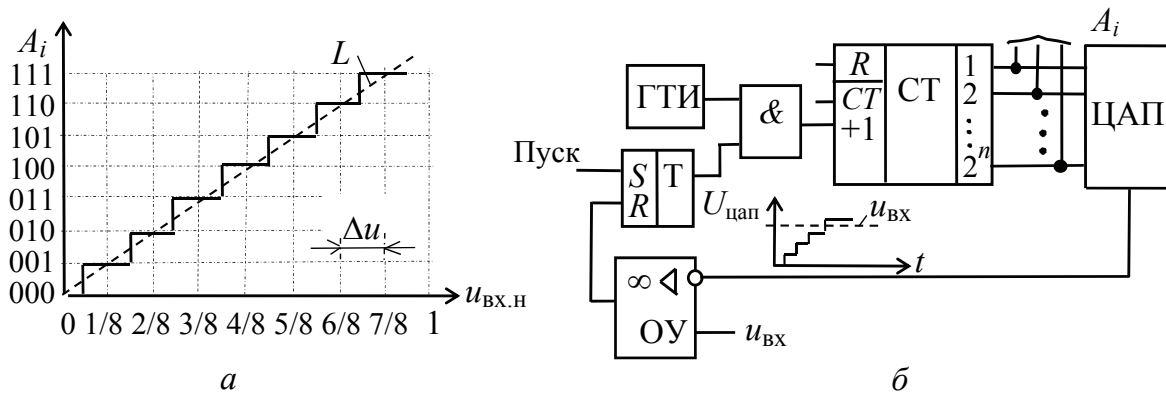


Рис. 7.1. АЦП последовательного счета с ЦАП в цепи обратной связи:
 а – характеристика идеального АЦП в нормированных единицах входного напряжения; б – структурная схема АЦП последовательного счета с ЦАП в цепи обратной связи

На рис. 7.1, а приведена характеристика идеального АЦП в нормированных единицах входного напряжения $u_{вх.н} = u_{вх} / u_{вх \max}$. Кроме ошибки квантования, при оценке точности АЦП учитывают дополнительные погрешности: *инструментальную* (погрешность смещения нуля, вызывающая смещение пунктирной прямой L влево или вправо от начала координат см. рис. 7.1, а), и *апертурную*, возникающую из-за несоответствия значения входного сигнала $u_d(k)$ преобразованному цифровому коду A_i . Несоответствие возникает, если входной сигнал в течение интервала дискретизации Δt изменяется более чем на значение шага квантования Δu .

7.2. Основные параметры АЦП

К основным параметрам АЦП относят:

– число разрядов выходного кода $n = 8, \dots, 16$, отображающего исходную аналоговую величину, которое может формироваться

на выходе АЦП. При использовании двоичного кода $n = \log_2(N + 1)$, где $N + 1$ – максимальное число кодовых комбинаций (уровней квантования) на выходе АЦП;

– диапазон изменения входного напряжения $u_{\text{вх.мах}}$. Отметим, что АЦП может обрабатывать входную информацию в виде однополярного аналогового напряжения с пределами $0 \dots u_{\text{вх.мах}}$ и двухполярного $\pm u_{\text{вх.мах}}/2$;

– абсолютную разрешающую способность $\text{ЗМР} = \Delta u$ (значение младшего разряда) – среднее значение минимального изменения входного сигнала $u_{\text{вх}}$, обуславливающего увеличение или уменьшение выходного кода на единицу. Значение ЗМР определяется разрядностью выходного кода и диапазоном входного напряжения;

– абсолютную погрешность δ_i преобразования в конечной точке шкалы – отклонение реального максимального значения входного сигнала $u_{\text{вх.мах}}$ от максимального значения идеальной характеристики L АЦП (см. рис. 7.1, а). Обычно δ_i измеряется в ЗМР;

– максимальную частоту преобразования (десятки и сотни килогерц);

– время преобразования входного сигнала: $t_{\text{пр.мах}} \leq (1/2)\Delta t$.

Состав АЦП в отличие от ЦАП может изменяться в значительной степени в зависимости от выбранного метода преобразования и способа его реализации. Наибольшее распространение получили три основных метода: последовательного счета, поразрядного кодирования и считывания.

Метод последовательного счета основан на уравнивании входной величины суммой одинаковых по величине эталонов (суммой шагов квантования). Момент уравнивания определяется с помощью одного компаратора, а количество эталонов, уравнивающих входную величину, подсчитывается с помощью счетчика.

Метод поразрядного кодирования (уравнивания) предусматривает наличие нескольких эталонов (часто реализованных в виде уравнивающего сдвигающего регистра), обычно пропорциональных по величине степеням числа 2, и сравнение этих эталонов с аналоговой величиной. Сравнение начинается с эталона старшего разряда. В зависимости от результата этого сравнения формируется значение старшего разряда выходного кода. Если эталон больше входной величины, то в старшем разряде ставится 0 и далее производится уравнивание входной величины следующим по значению эталоном. Если эталон равен или меньше входной величины,

то в старшем разряде выходного кода ставится 1 и в дальнейшем производится уравнивание разности между входной величиной и первым эталоном.

Наибольшим быстродействием обладают преобразователи, построенные по методу считывания. *Метод считывания* подразумевает наличие $2^n - 1$ эталонов при n -разрядном двоичном коде. Входная аналоговая величина одновременно сравнивается со всеми эталонами. В результате преобразования получается параллельный код в виде логических сигналов на выходах $2^n - 1$ компараторов.

7.3. Варианты реализации АЦП последовательного счета

В качестве примера рассмотрим структурную схему АЦП последовательного счета с ЦАП в цепи обратной связи (рис. 7.1, б) и вариант ее реализации (рис. 7.2). По сигналу «Пуск» на вход обнуленного счетчика СТ начинают подаваться импульсы генератора тактовой частоты ГТИ (см. рис. 7.1, б). По мере поступления этих импульсов растет входной код ЦАП и ступенчато повышается напряжение $u_{\text{цап}}$ на его выходе, причем уровень ступени соответствует шагу квантования Δu входного напряжения $u_{\text{вх}}$ АЦП.

Процесс преобразования заканчивается, когда напряжение $u_{\text{цап}}$ станет чуть больше входного напряжения $u_{\text{вх}}$ АЦП, поданного на вход ОУ, на котором собран компаратор. При этом работа счетчика прекращается, а на его выходе устанавливается код A_i , являющийся цифровым эквивалентом напряжения $u_{\text{вх}}$.

Согласно рассмотренной структурной схеме АЦП на рис. 7.2 приведен вариант реализации модели 4-разрядного АЦП последовательного счета с ЦАП, состоящего из операционного усилителя **OPAMP1** и резистивной матрицы **R1**, ..., **R4** со взвешенными сопротивлениями. Переключатели **Key1**, ..., **Key4** в схеме (при разомкнутом ключе **Space**) служат для проверки работы счетчика СТ, а осциллограф **XSC1** – для снятия осциллограмм напряжения с выхода ЦАП и входа компаратора.

При запуске моделирования АЦП сформированные генератором **E1** импульсы подаются на вход счетчика СТ, число которых последовательно высвечивается на 7-сегментном индикаторе. Выходные поразрядные сигналы со счетчика поступают также на входы

логического анализатора **XLF1** и входы резистивной матрицы **R1**, ..., **R4**, а суммарное напряжение с матрицы – на вход ОУ. Ступенчатое напряжение $u_{\text{цап}}$ с выхода **OPAMP1** (рис. 7.3) подается на вход компаратора, собранного на операционном усилителе **OPAMP2**. На этот же вход подано постоянное напряжение $u_{\text{вх}}$ с генератора **E7** через делитель **R6-R7**. В момент сравнения указанных напряжений, компаратор срабатывает, на элемент И (**AND**) подается логический 0 и прекращается работа счетчика, а на индикаторе высвечивается цифровой код (число шагов квантования), соответствующий уровню $u_{\text{д}}(k)$.

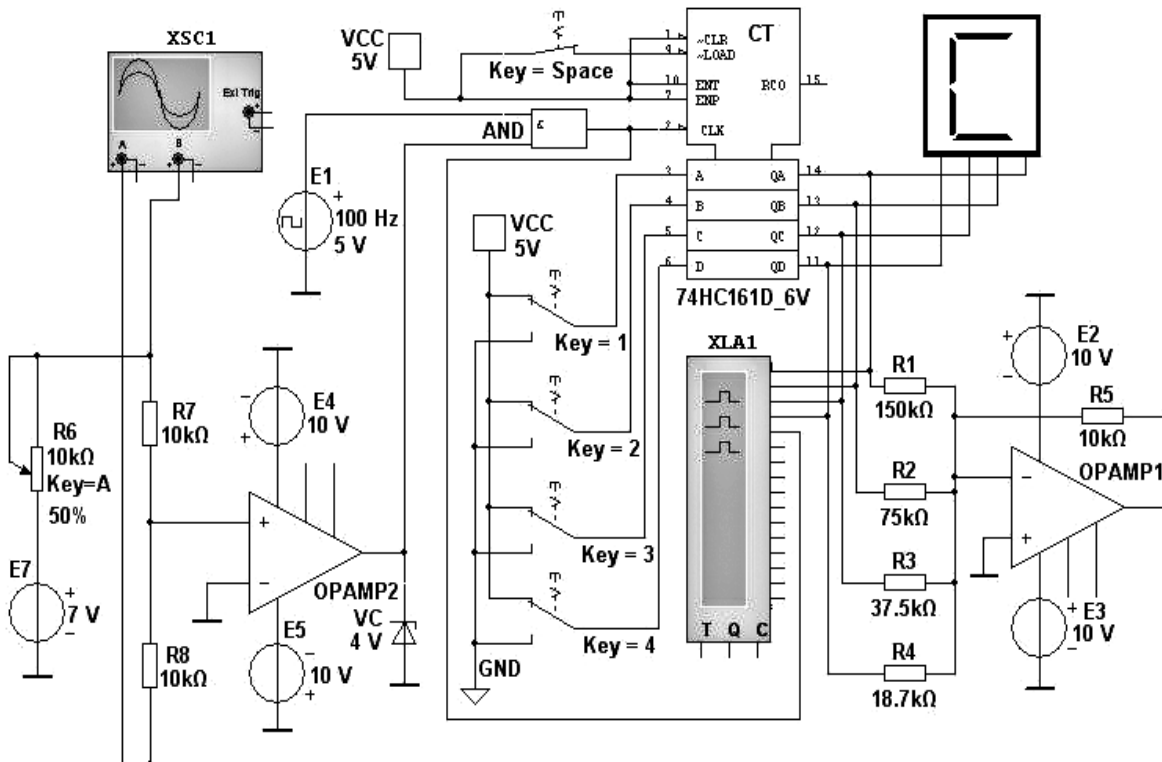


Рис. 7.2. Реализация АЦП последовательного счета с ЦАП в цепи обратной связи

Анализ временных диаграмм сигналов с выхода счетчика и осциллограмм напряжений со входов компаратора (см. рис. 7.3) показывает, что счетчик прервал счет с приходом двенадцатого тактового импульса, поэтому на семи-сегментном индикаторе высветилось число C_{16} (12_2) (см. рис. 7.2).

Погрешность преобразования зависит от шага квантования (высоты ступени напряжения $u_{\text{цап}}$), погрешности в формировании ступенчатого напряжения $u_{\text{цап}}$ и ошибки компаратора в определе-

нии равенства $u_{вх}$ и $u_{цап}$. Время преобразования непостоянно и зависит от уровня напряжения $u_{вх}$. При заданном числе разрядов АЦП время преобразования определяется числом периодов счетных импульсов.

По структуре построения ИМС АЦП подразделяют на АЦП с применением ЦАП и без них. К БИС АЦП без ЦАП, например ИМС КР572ПВ2, К107ПВ2 и другие, относят АЦП последовательного счета с двойным интегрированием (на первом такте – входного напряжения, на втором – эталонного напряжения с преобразованием результатов интегрирования во временной интервал и в эквивалентный цифровой код) для сглаживания импульсных помех, повышения точности и помехозащищенности данного типа АЦП.

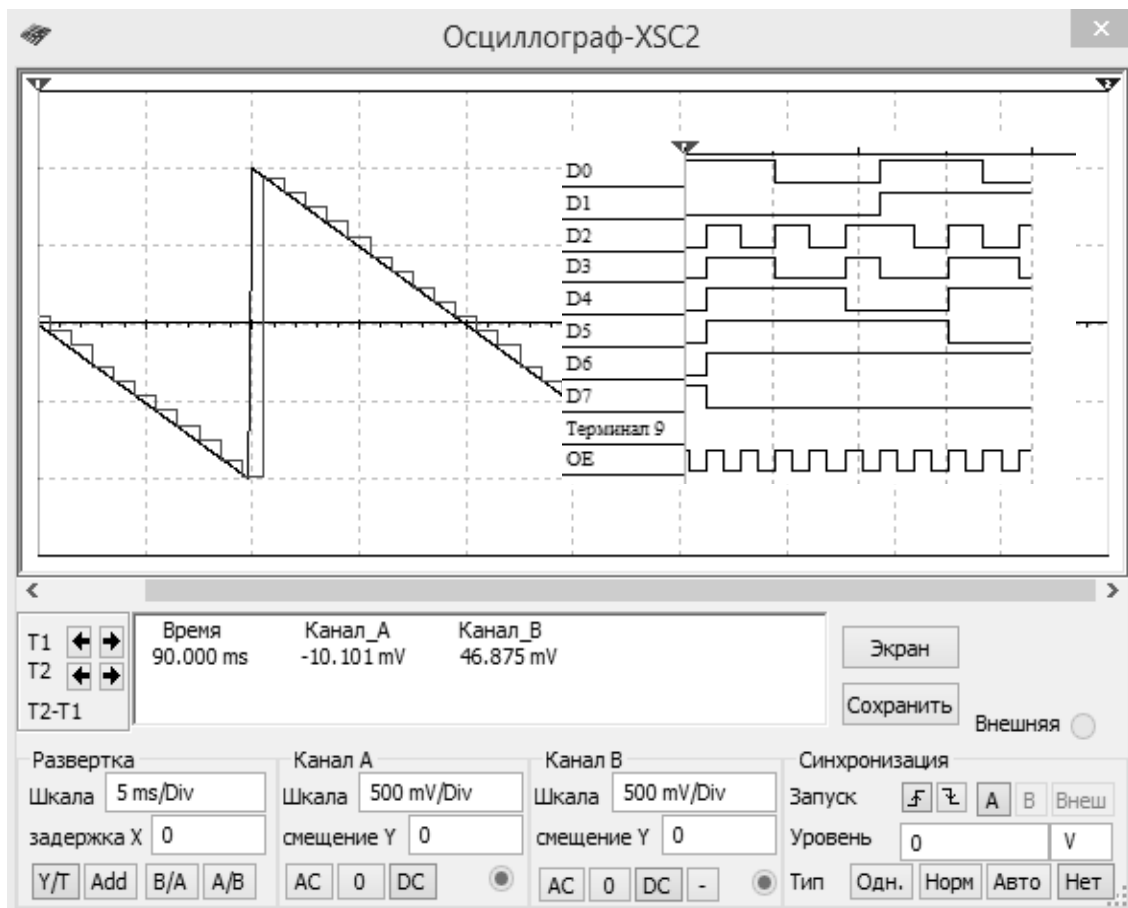


Рис. 7.3. Анализ временных диаграмм сигналов с выхода счетчика и осциллограмм напряжений

Платы АЦП/ЦАП, например модели LTC российской компании ЗАО «Л-КАРД», широко применяют в цифровых измерительных приборах, в системах и устройствах обработки и отображения

информации, автоматических системах контроля и управления, устройствах ввода – вывода информации ЭВМ и т. д.

Основные направления развития АЦП – повышение быстродействия главных узлов, в частности, компараторов до 5–10 нс, повышение их точности до 0,05–0,005%, увеличение разрядности преобразователей до 24, использование микропроцессоров в преобразователях.

Лабораторная работа № 8

Задание 1. Собрать на рабочем поле среды Multisim схему для испытания *аналого-цифрового преобразователя с ЦАП* (рис. 7.4) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

В схему (рис. 7.4) включены собственно библиотечный 8-разрядный АЦП (ADC); источники опорного напряжения **E1** и **E2** (подключены к входам **Vref+** и **Vref-** АЦП); генератор **E4** для синхронизации работы (подключен к входу **SOC**) и разрешения (вход **OE**) на выдачу двоичной информации на выходы **D0**, ..., **D7** АЦП, с которыми соединены входы логического анализатора **XLA1** и пробники **X0**, ..., **X7**; функциональный генератор **XFG1** в качестве источника входного сигнала $u_{ВХ}$ (подключен к входу **Vin**); ЦАП (**DAC**) и осциллограф **XSC1**.

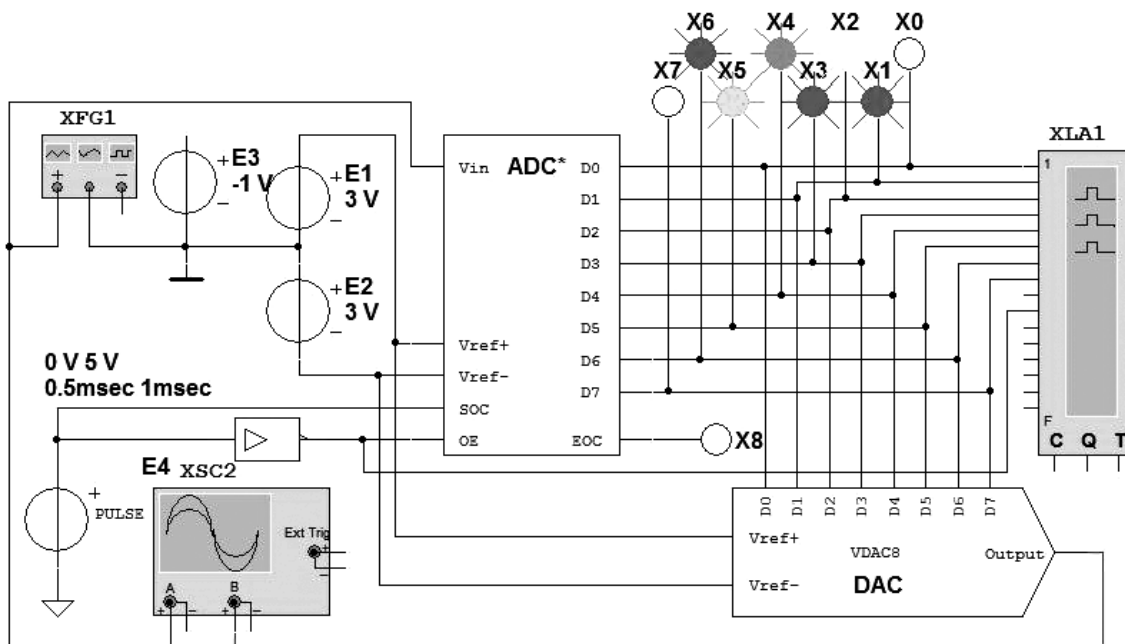


Рис. 7.4. Схема для испытания интегрального аналого-цифрового преобразователя

Выход **ЕОС** служит для передачи двоичной информации АЦП, например, на ЭВМ.

Задание 2. Исследовать точность преобразования АЦП уровней входного напряжения $u_{\text{вх}}$ в цифровой код с помощью пробников **X0**, ..., **X7**, логического анализатора **XLA1**, а также ЦАП и осциллографа **XSC1**.

С этой целью:

– временно удалить провод 1 (см. рис. 7.4) и подключить вход **Vin** АЦП к положительному полюсу источника постоянного напряжения **ЕЗ**;

– составить таблицу, аналогичную данной далее, в первый столбец которой записать уровни напряжения $u_{\text{вх}} = 0,1; 0,2; 0,5; 1,0; 1,5; 2,0; 2,4; -0,5; -1,0; -2,0$ В, поочередно задаваемые в диалоговом окне генератора **ЕЗ**;

– установить в диалоговых окнах генераторов **Е1** и **Е2** ЭДС $E_1 = 2,5$ В и ЭДС $E_2 = -2,5$ В;

– запустить программу моделирования АЦП и заносить в поля составленной таблицы значения напряжения $u_{\text{вых(цап)}}$ с выхода ЦАП, измеряемые на экране осциллографа с помощью визирной линии; двоичный эквивалент $D_{(2)}$ преобразуемого напряжения, определяемый по свечению пробников **X7**, ..., **X0**; шестнадцатеричный код $D_{(16)}$, считываемый с дисплея анализатора **XLA1**;

– получаемые с выхода АЦП десятичные инверсные сигналы $D_{(10)\text{инв}}$ пересчитать на неинверсные $D_{(10)}$ по выражению $D_{(10)} = D_{(10)\text{инв}} - 128$ и занести в соответствующие столбцы таблицы;

– расчетные десятичные эквиваленты $D_{(10)\text{расч}}$ двоичного кода $D_{(2)}$ на выходе АЦП при заданном значении входного напряжения $u_{\text{вх}}$ определить по формуле (7.2):

$$D_{(10)\text{расч}} = \frac{256u_{\text{вх}}}{E_1 + |E_2|} \quad (7.2)$$

и занести во второй справа столбец таблицы;

– рассчитать погрешности измерения напряжения по выражению (7.3):

$$\Delta U\% = \frac{100(u_{\text{вых(цап)}} - u_{\text{вх}})}{u_{\text{вх}}} \quad (7.3)$$

и занести в правый столбец таблицы.

В качестве примера в таблице приведены данные измерений при моделирования АЦП при $E_1 = 3$ В и $E_2 = -3$ В, которые близки к расчетным значениям. Так, при $E_1 = |E_2| = 3$ В и $u_{вх} = E_3 = 1$ В расчетный десятичный эквивалент $D_{(10)расч} = 256 \cdot 1/6 \approx 42,67$ при измеренном $D_{(2)} = 10101010$ и $D_{(10)} = 42$. При этом погрешность измерения составила 3,56%.

Результаты эксперимента

| $u_{вх}$, В | $u_{вых(цап)}$, В | $D_{(2)}$ | $D_{(16)}$ | $D_{(10)инв}$ | $D_{(10)}$ | $D_{(10)расч}$ | $\Delta U\%$ |
|-----------------|-----------------------|-----------|------------|---------------|------------|----------------|--------------|
| 0,1 | 0,09375 | 10000100 | 84 | 132 | 4 | 4,27 | 6,25 |
| 0,5 | 0,5156 | 10010101 | 95 | 149 | 21 | 21,33 | 3,12 |
| 1,0 | 0,9644 | 10101010 | AA | 170 | 42 | 42,67 | 3,56 |
| 2,0 | 2,017 | 11010101 | D5 | 213 | 85 | 85,34 | 0,85 |
| 2,5 | 2,484 | 11101010 | EA | 234 | 106 | 106,67 | 0,64 |
| 2,9 | 2,906 | 11111011 | FB | 251 | 123 | 123,74 | 0,21 |
| -1,0 | -0,9844 | 01010101 | 55 | 85 | -43 | -42,67 | 3,56 |
| 0,1 | 0,09375 | 10000100 | 84 | 132 | 4 | 4,27 | 6,25 |

Содержание отчета

1. Название и цель работы.
2. Перечень приборов, использованных в экспериментах, с их краткими характеристиками.
3. Изображение электрической схемы для испытания аналого-цифрового преобразователя.
4. Копии осциллограмм и временных диаграмм сигналов с разных узлов схемы, отображающие работу исследуемого АЦП.
5. Таблица с результатами измерений и расчетов входных отсчетов входного напряжения и выходных кодов АЦП.
6. Выводы по работе.

СОДЕРЖАНИЕ

| | |
|---|----|
| ПРЕДИСЛОВИЕ..... | 3 |
| ТЕМА 1. ИССЛЕДОВАНИЕ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ В СРЕДЕ СХЕМОТЕХНИЧЕСКОГО МОДЕЛИРОВАНИЯ MULTISIM | 4 |
| 1.1. Среда схемотехнического моделирования Multisim..... | 4 |
| 1.1.1. Контрольно-измерительные приборы Multisim | 6 |
| 1.1.2. Компоненты Multisim | 12 |
| 1.1.3. Создание схем в Multisim | 13 |
| 1.2. Определение комбинационных и последовательных устройств..... | 15 |
| 1.3. Основные элементы алгебры логики..... | 15 |
| 1.4. Базовые логические элементы | 17 |
| 1.5. Представление логических функций математическими выражениями | 18 |
| 1.6. Переход от логической функции к логической схеме | 19 |
| Лабораторная работа № 1 | 20 |
| ТЕМА 2. ЗАПОМИНАЮЩИЕ ЭЛЕМЕНТЫ. ТРИГГЕРЫ..... | 26 |
| 2.1. Асинхронный и синхронный <i>RS</i> -триггеры | 26 |
| 2.2. <i>T</i> -триггер | 28 |
| 2.3. <i>D</i> -триггер..... | 29 |
| 2.4. <i>JK</i> -триггер | 30 |
| Лабораторная работа № 2 | 32 |
| ТЕМА 3. ИССЛЕДОВАНИЕ ИНТЕГРАЛЬНЫХ ПРЕОБРАЗОВАТЕЛЕЙ КОДОВ (ДЕШИФРАТОРА, ШИФРАТОРА) И КОММУТАЦИОННЫХ УЗЛОВ (ДЕМУЛЬТИПЛЕКСОРА И МУЛЬТИПЛЕКСОРА) | 37 |
| 3.1. Дешифратор | 37 |
| 3.2. Шифратор..... | 38 |
| 3.3. Мультиплексор | 40 |
| 3.4. Демультимплексор..... | 41 |
| Лабораторные работы № 3–4 | 42 |

| | |
|--|----|
| ТЕМА 4. ЗАПОМИНАЮЩИЕ УЗЛЫ. РЕГИСТРЫ..... | 53 |
| 4.1. Классификация регистров | 53 |
| 4.2. Параллельный регистр на <i>RS</i> -триггерах..... | 54 |
| Лабораторная работа № 5 | 56 |
| ТЕМА 5. ЗАПОМИНАЮЩИЕ УЗЛЫ. СЧЕТЧИКИ | 63 |
| 5.1. Классификация счетчиков | 63 |
| 5.2. Счетчик с непосредственными связями..... | 64 |
| 5.3. Суммирующий синхронный счетчик | 66 |
| 5.4. Реверсивный синхронный счетчик | 67 |
| 5.5. Десятичный счетчик..... | 68 |
| Лабораторная работа № 6 | 70 |
| ТЕМА 6. ЦИФРО-АНАЛОГОВЫЙ ПРЕОБРАЗОВАТЕЛЬ | 76 |
| 6.1. Структура резистивных матриц ЦАП | 76 |
| 6.2. Основные параметры ЦАП..... | 79 |
| Лабораторная работа № 7 | 80 |
| ТЕМА 7. АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ..... | 86 |
| 7.1. Структурная схема АЦП последовательного действия.... | 86 |
| 7.2. Основные параметры АЦП..... | 87 |
| 7.3. Варианты реализации АЦП последовательного счета | 89 |
| Лабораторная работа № 8 | 92 |

Учебное издание

*Кобайло Александр Серафимович
Жияк Надежда Александровна
Миронов Игорь Александрович*

**СХЕМОТЕХНИКА
ЦИФРОВЫХ ВЫЧИСЛИТЕЛЬНЫХ
МАШИН**

Лабораторный практикум

Учебно-методическое пособие

Редактор *Е. И. Гоман*
Компьютерная верстка *Е. В. Ильченко*
Корректор *Е. И. Гоман*

Подписано в печать 06.09.2018. Формат 60×84¹/₁₆.
Бумага офсетная. Гарнитура Таймс. Печать ризографическая.
Усл. печ. л. 5,6. Уч.-изд. л. 5,8.
Тираж 150 экз. Заказ .

Издатель и полиграфическое исполнение:
УО «Белорусский государственный технологический университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/227 от 20.03.2014.
Ул. Свердлова, 13а, 220006, г. Минск.