

7. ПРАКТИКУМ

7.1. ИССЛЕДОВАНИЕ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ В СРЕДЕ СХЕМОТЕХНИЧЕСКОГО МОДЕЛИРОВАНИЯ MULTISIM

7.1.1. 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, разработанной группой Electronics Workbench (входящей в корпорацию National Instruments). Особенностью программной среды Multisim является наличие в ее библиотеке более 16 000 электронных компонентов, а также наличие 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 является программой 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. 7.1 и состоит из следующих элементов: строка меню, панель инструментов, панель разработки, окно редактирования, приборная панель.

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 также полностью настраивается, вплоть до контекстных меню разных объектов.

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

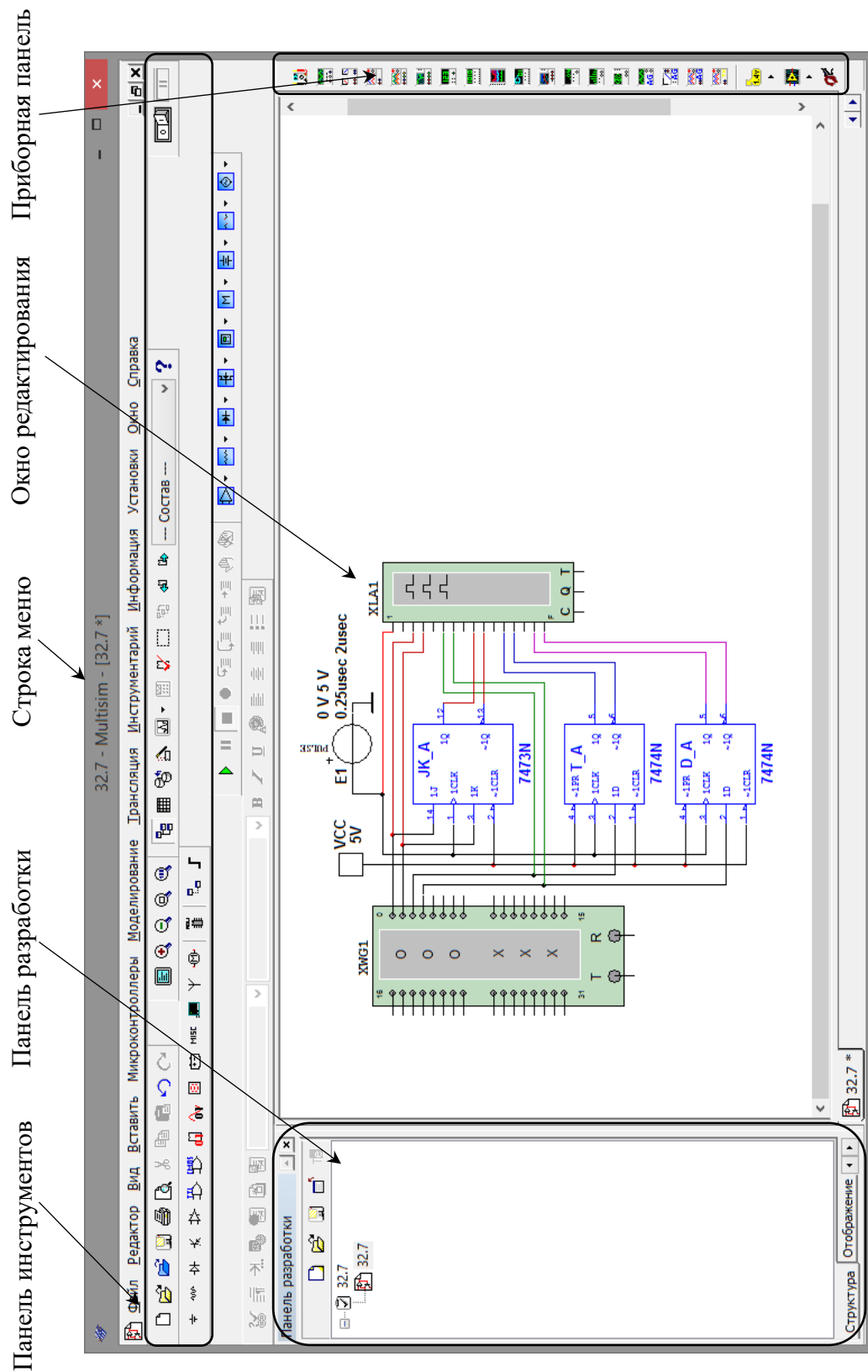


Рис. 7.1.1. Интерфейс программы Multisim

7.1.2. Контрольно-измерительные приборы Multisim

Осциллограф (Oscilloscope) позволяет проводить анализ сигналов во временной области. Внешний вид и лицевая панель осциллографа показаны на рис. 7.2.

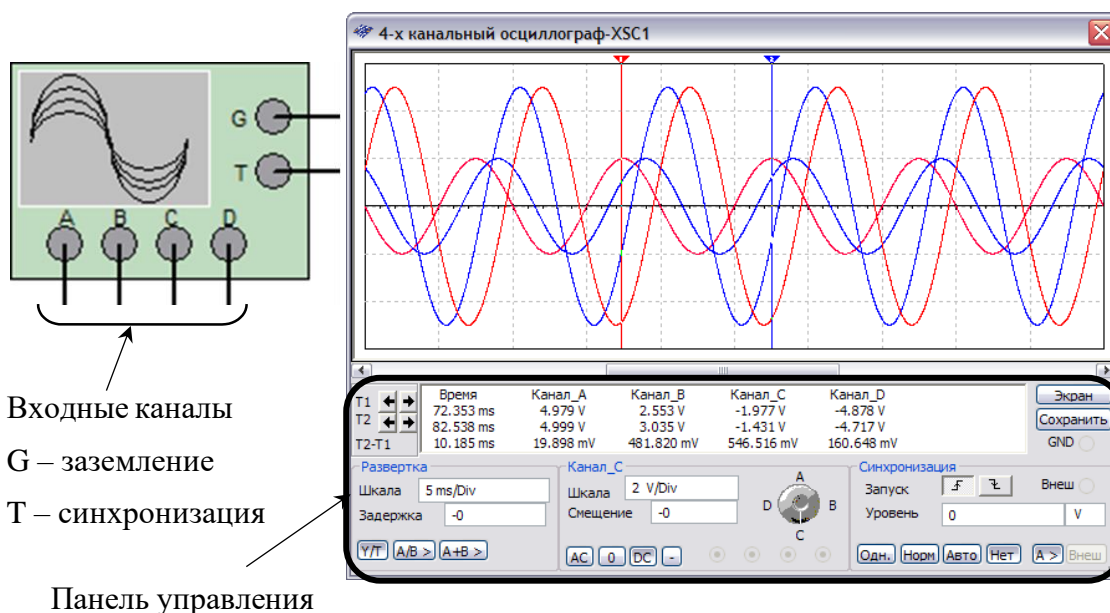


Рис. 7.2. Четырехканальный осциллограф

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

Для настройки отображения измеряемого сигнала используется панель управления осциллографа (рис. 7.3). Выбор режима работы по входу осуществляется нажатием кнопок «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} 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 >.

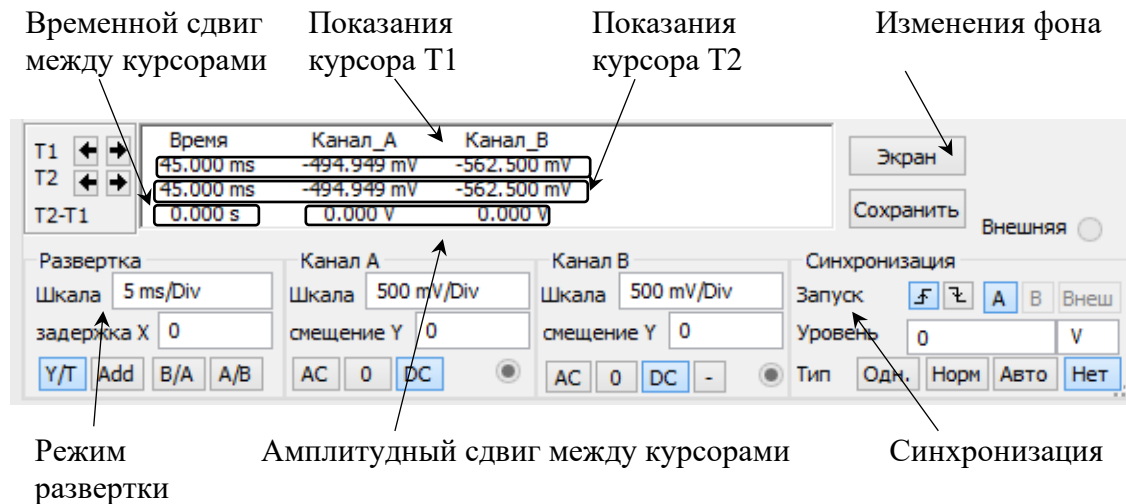


Рис. 7.3. Панель управления осциллографа

Генератор слов (Word Generator) предназначен для генерации до 8192 32-разрядных двоичных слов. Внешний вид и лицевая панель генератора показаны на рис. 7.4. 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).

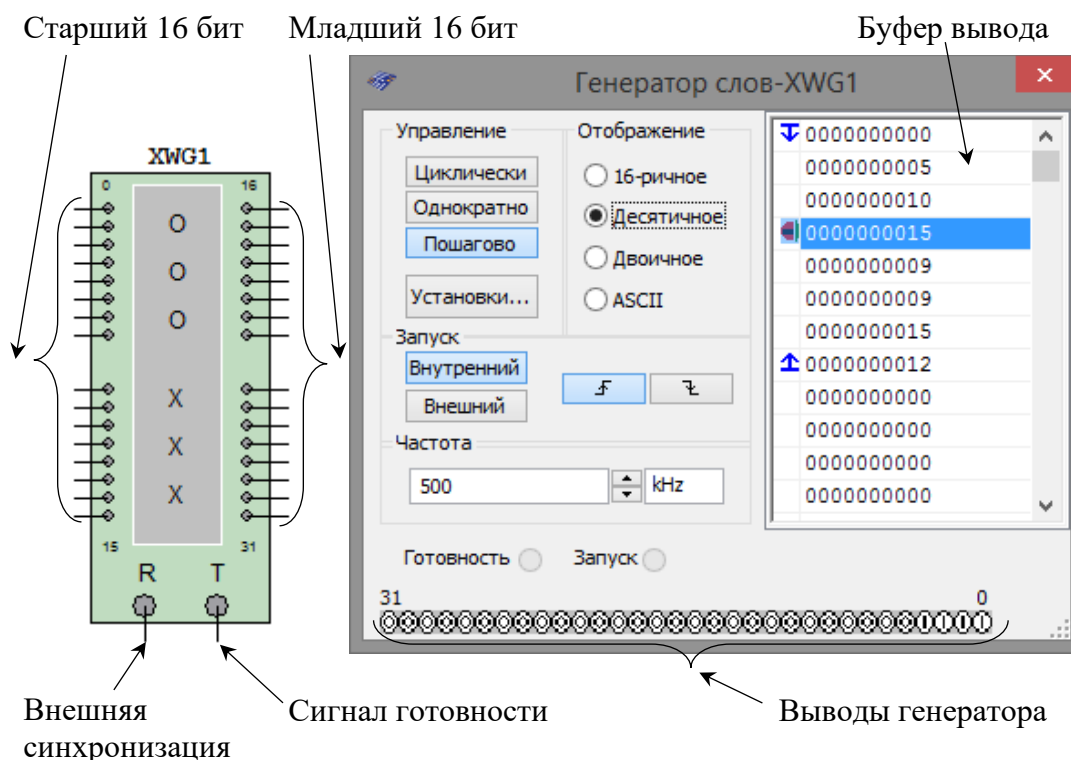


Рис. 7.4. Генератор слов

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. 7.5):

- 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.

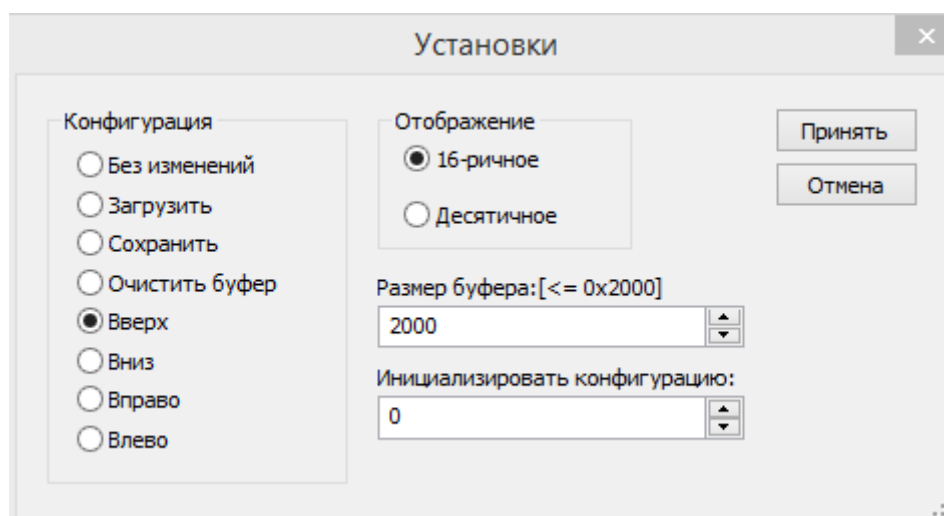


Рис. 7.5. Окно свойств буфера

Логический анализатор (Logic Analyzer) – 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 логического анализатора показаны на рис. 7.6.

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. 7.7).

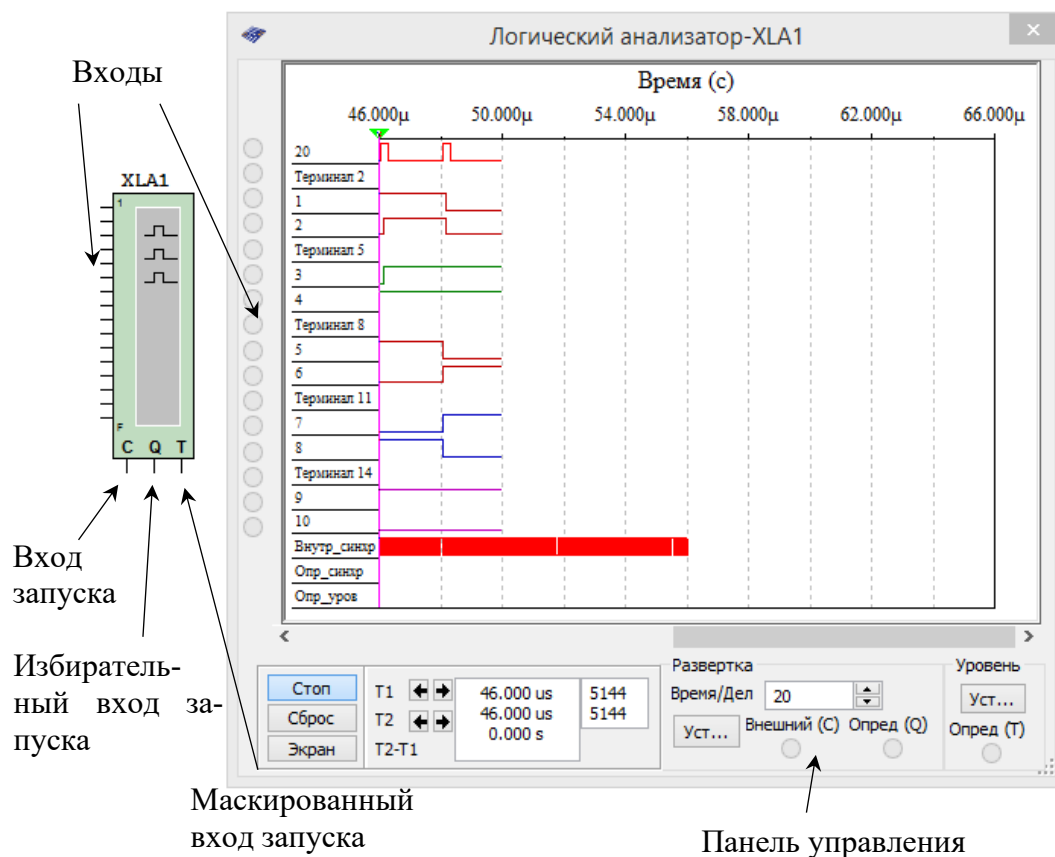


Рис. 7.6. Логический анализатор

При нажатии кнопки «Установки» в группе «Развертка» (тактовый генератор) открывается диалоговое окно настройки параметров тактирования входных сигналов (рис. 7.8).

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

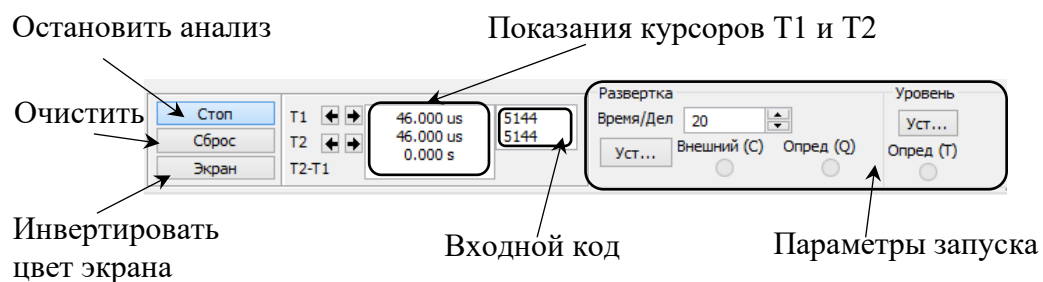


Рис. 7.7. Панель управления логического анализатора

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 осуществляется фильтрация логических уровней и синхронизация входных каналов.

Установки синхронизации

Источник
☐ Внешний ☒ Внутренний

Тактовая частота
10 MHz

Определитель
x

Дискретизация

До порога	100
После порога	1000
Порог(V)	4 V

Принять
Отмена

Рис. 7.8. Настройка параметров синхронизации

7.1.3. 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. 7.9, а) и реальными (real) (рис. 7.9, б).

Реальные компоненты являются полными аналогами компонентов, выпускаемых или выпущенных радиоэлектронной промышленностью.

Виртуальные компоненты являются математическими моделями семейств (Family) компонентов (резисторы, конденсаторы и т. д.) с любыми произвольными параметрами, присущими данной категории.

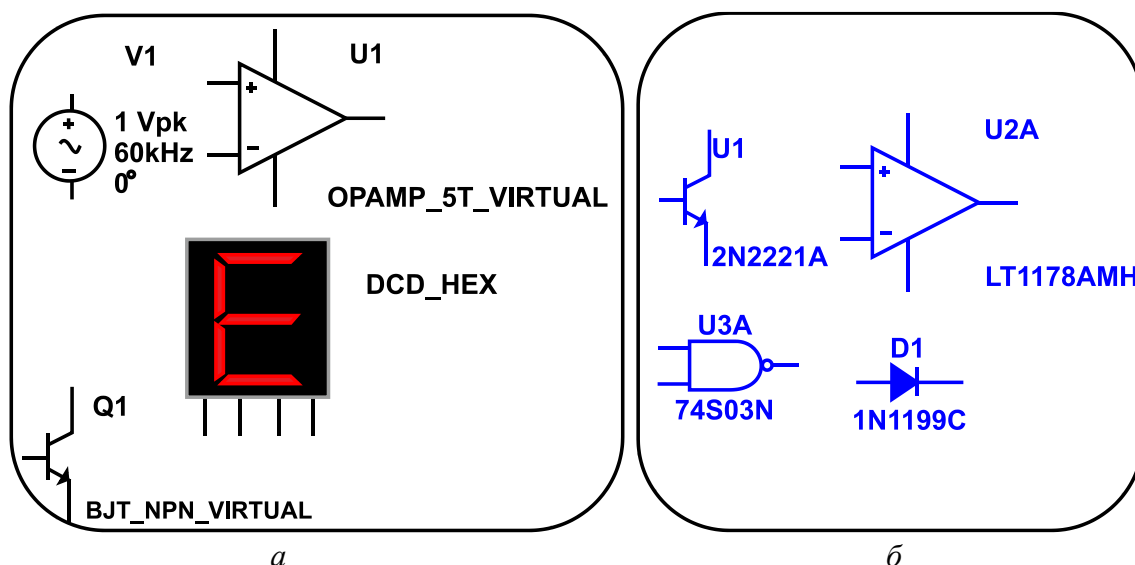


Рис. 7.9. Компоненты Multisim:
а – виртуальные; б – реальные

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

Рассмотрим основные группы компонентов базы данных Multisim.

Группа **Sources** (Источники) содержит 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) включает модели резисторов, конденсаторов, индуктивностей, трансформаторов, виртуальных механических ключей и т. д.

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

Группа **Transistors** (Транзисторы) включает 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 ТТЛ) содержит модели микросхем серий 74Sxx, 74LSxx, 74ALSxx и др.

Группа **CMOS** (цифровые микросхемы по технологии КМОП) содержит модели микросхем серии 74HCxx, 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.

7.1.4. Создание схем в Multisim

Процесс создания схемы начинается с выбора компонентов схемы. Выбранный компонент автоматически прикрепляется к курсору мыши, после чего размещается в любом месте рабочего окна (рис. 7.10).

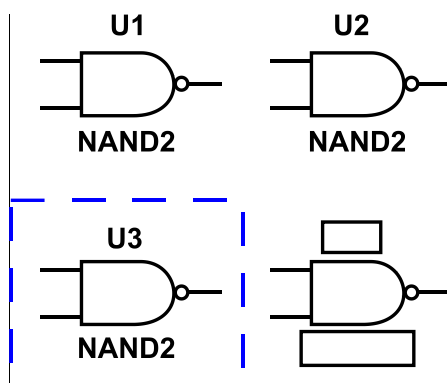



Рис. 7.10. Размещение компонентов схемы

Далее устанавливается ориентация компонентов (если это необходимо) с помощью команд всплывающего меню (нажатие правой кнопки мыши на выбранном компоненте) или комбинации клавиш $Alt + X$, $Alt + Y$ 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  устанавливается соединение между выводами элементов или соединительной точкой (рис. 7.11). Для добавления соединительной точки необходимо нажать комбинацию клавиш *Ctrl + J* или выбрать соответствующий пункт во всплывающем меню нажатием правой кнопки мыши.

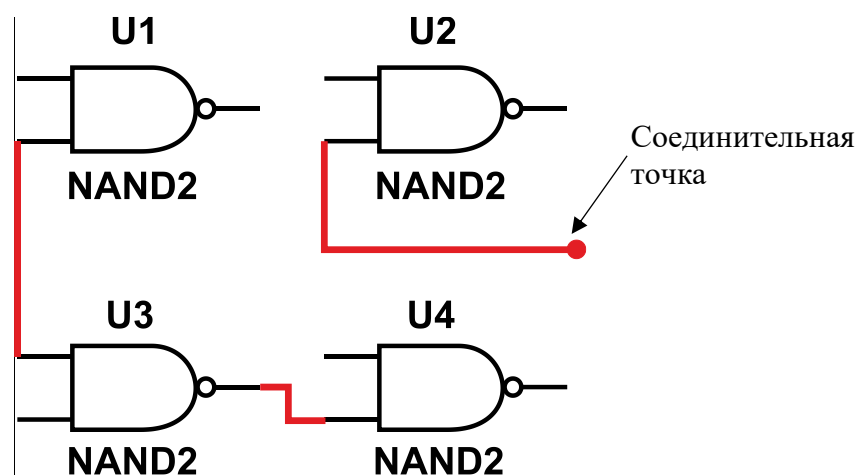



Рис. 7.11. Соединение элементов схемы

При изменении существующего соединения курсор отображается в виде перекрестия с двумя отрезками вдоль соединительной линии  (рис. 7.12).

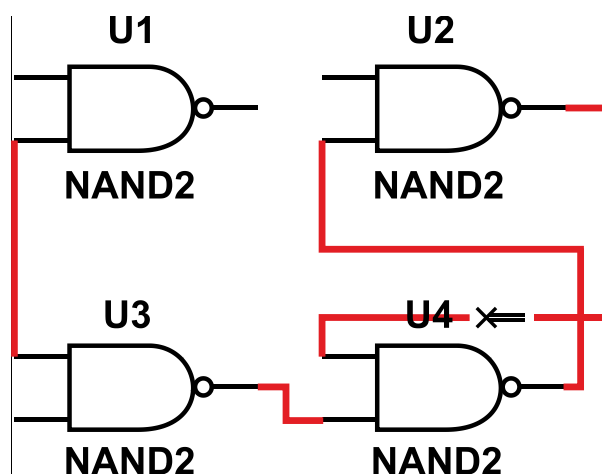


Рис. 7.12. Изменение соединения элементов схемы

Практические задания

Задание 1. Запустить среду разработки Multisim и собрать на рабочем поле среды Multisim схему для испытания *основных и базовых логических элементов* (рис. 7.13) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

Схема (рис. 7.13) 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 уровнем 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**.

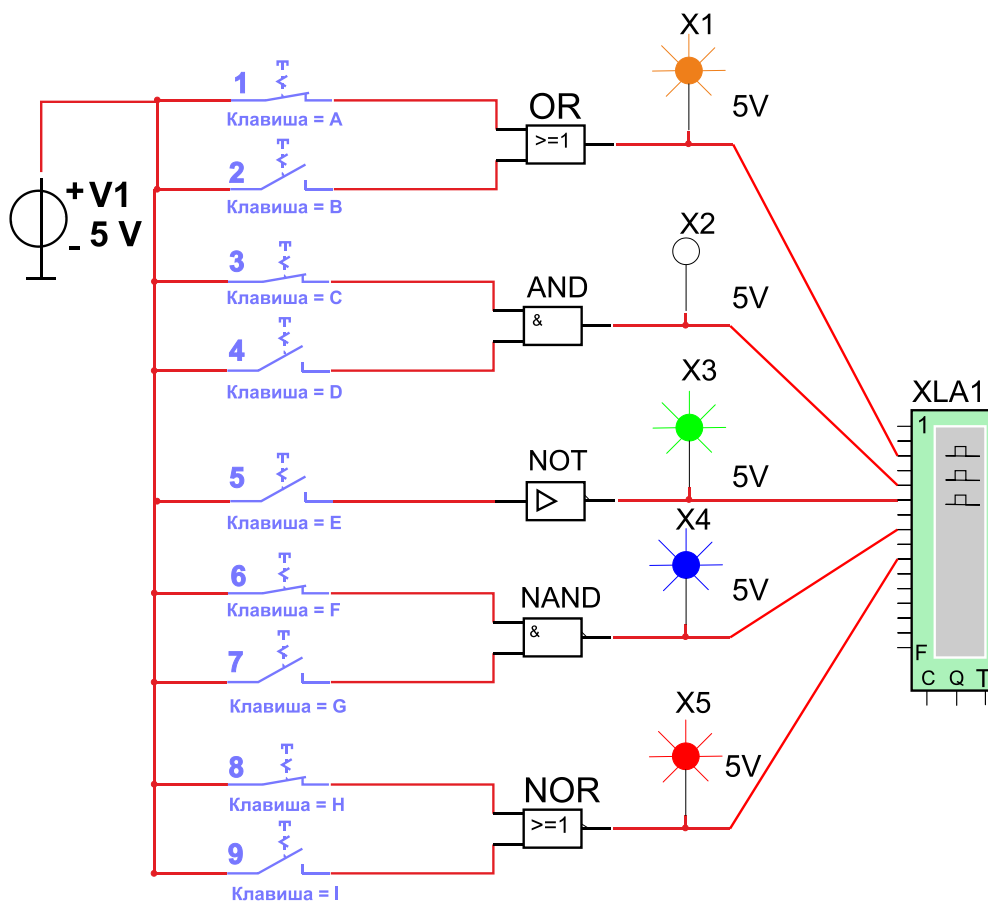


Рис. 7.13. Схема, собранная на двоичных основных и универсальных логических элементах

Для удобства измерения сигналов выходы логических элементов подключены к входам 2, 4, 6, 8 и 10 анализатора **XLA1** (рис. 7.14). 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. 7.1.

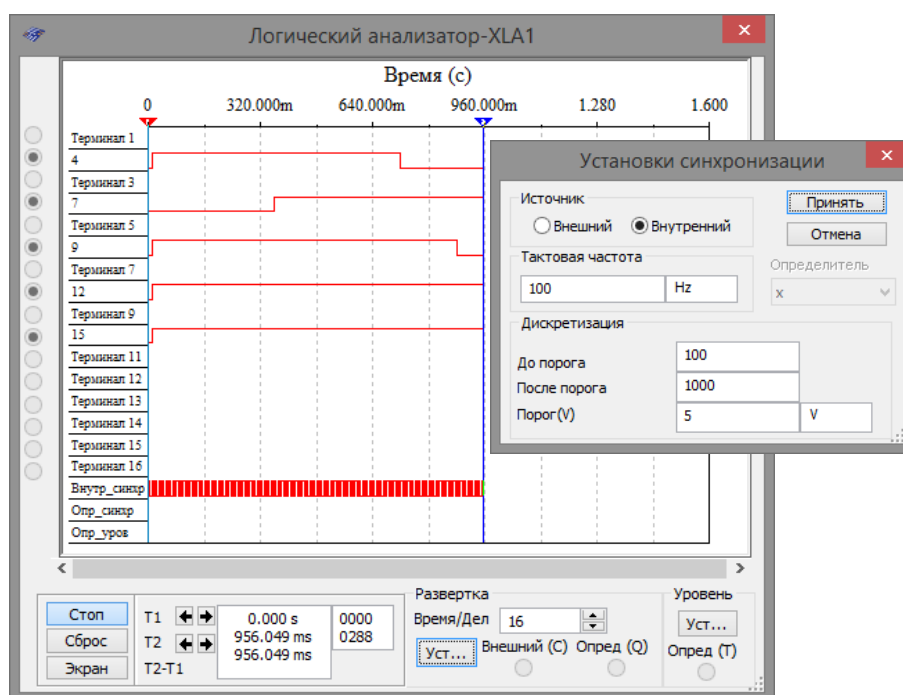


Рис. 7.14. Использование логического анализатора

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. 7.2) функции элементов **OR**, **AND** и **NOR** равны логической единице.

Таблица 7.1

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\TIL** на рабочее поле среды Multisim необходимые логические элементы и собрать схему для реализации заданной в табл. 7.2 логической функции y с тремя аргументами a , b и c . Скопировать собранную логическую схему на страницу отчета.

Таблица 7.2

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\TIL**, 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. 7.15).

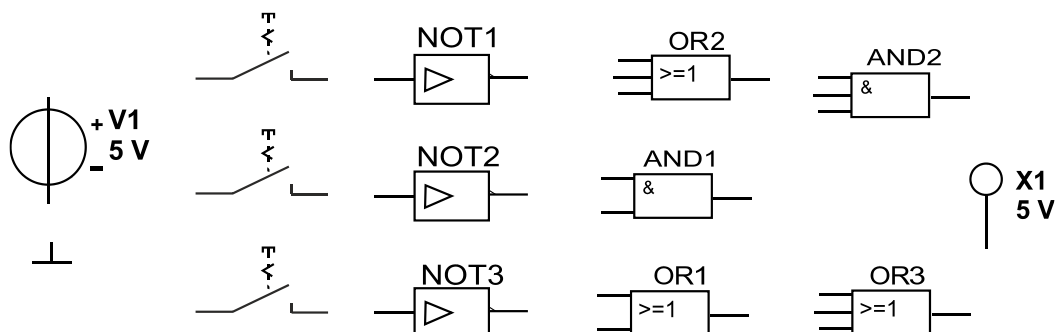


Рис. 7.15. Элементы для построения схемы

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 элементов в соответствии с логическими выражениями составляющих заданной функции и записав в отчете ожидаемые результаты выполнения операций на выходах элементов (рис. 7.16), приступим к моделированию.

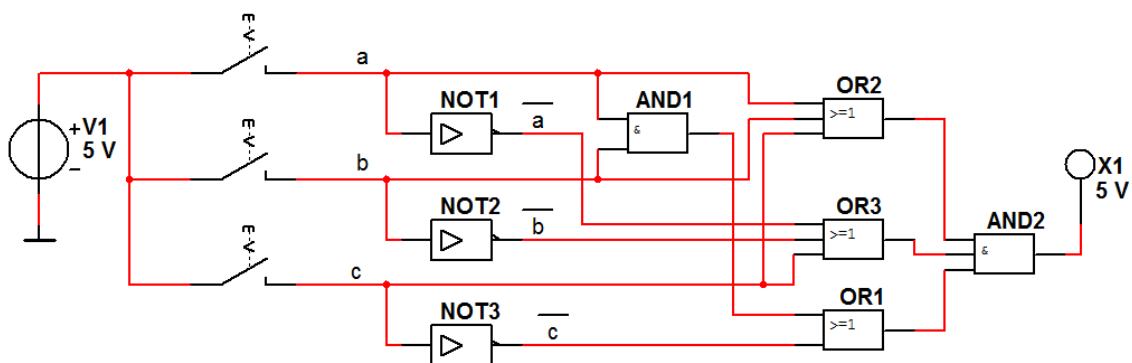


Рис. 7.16. Логическая схема

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 7.4), 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.

7.2. ЗАПОМИНАЮЩИЕ ЭЛЕМЕНТЫ. ТРИГГЕРЫ

Практические задания

Задание 1. Запустить среду разработки Multisim и собрать на рабочем поле среды Multisim схему для испытания *асинхронного RS-триггера* (рис. 7.17) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

Схема (рис. 7.17) 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.

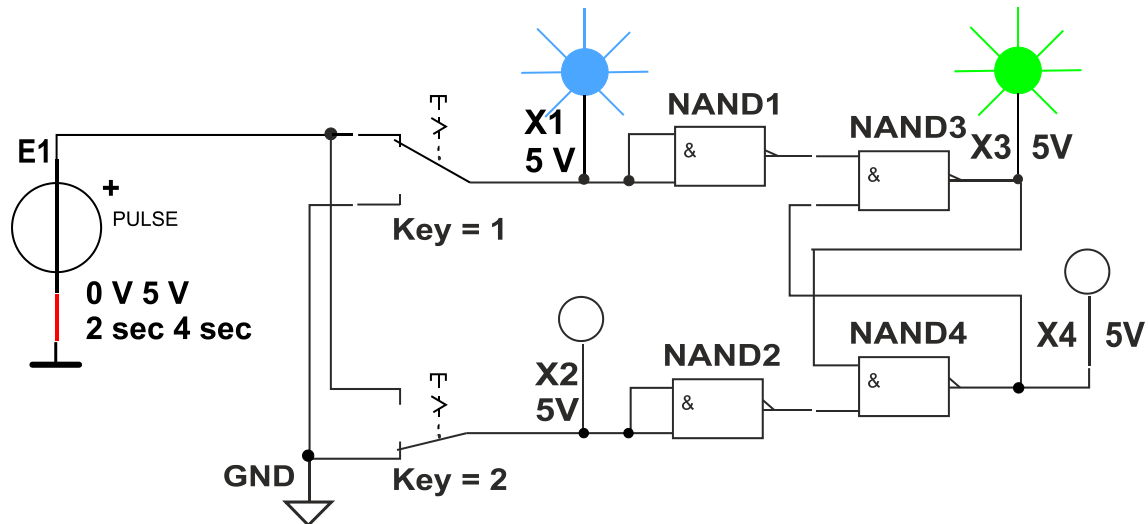


Рис. 7.17. Схема для испытания асинхронного RS -триггера

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, получить на его выходе сигналы $Q = 1$ и $\bar{Q} = 0$ (см. рис. 7.17). Убедиться, 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. 7.8), 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 $f_T = 10$ 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 таймера, приходящихся на одно деление.

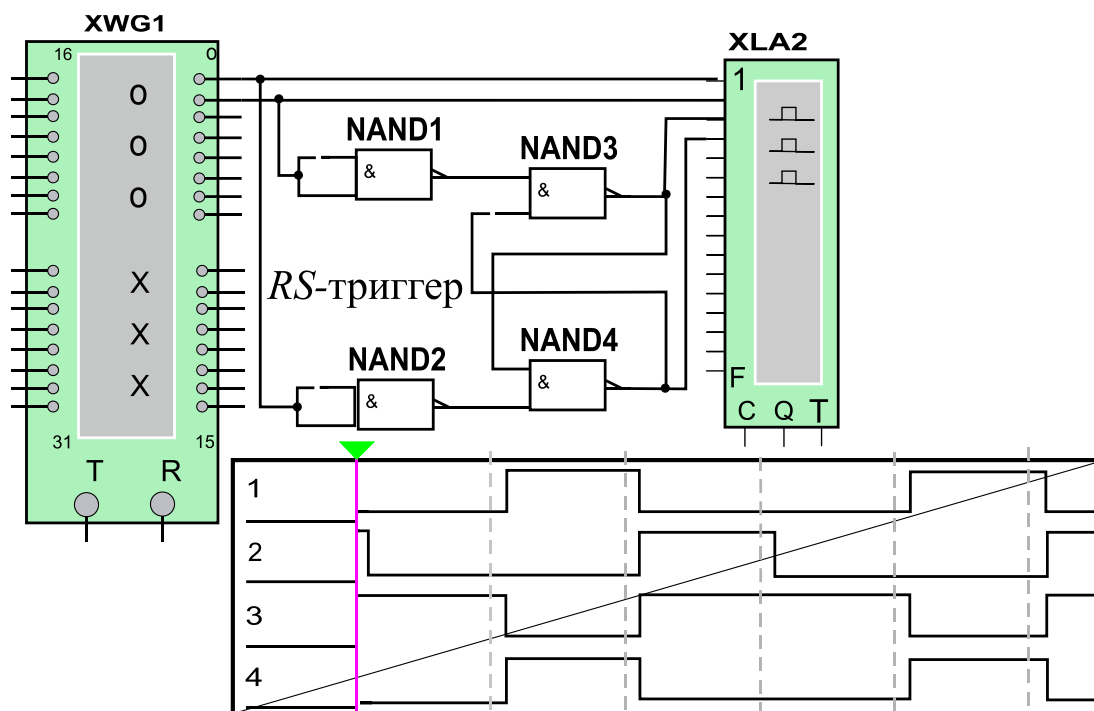


Рис. 7.18. Схема для испытания асинхронного *RS*-триггера

Получить на экране анализатора **XLA2** временную диаграмму состояний *RS*-триггера (см. рис. 7.18, 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. 7.19) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

В схему (рис. 7.19) включены: генератор **XWG1** (частота $f_z = 500$ кГц); логический анализатор **XLA1**; триггеры в интегральном исполнении: универсальный **JK**, счетный **T** и задержки **D**.

На \overline{ICLR} - и $\overline{1PK}$ -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.

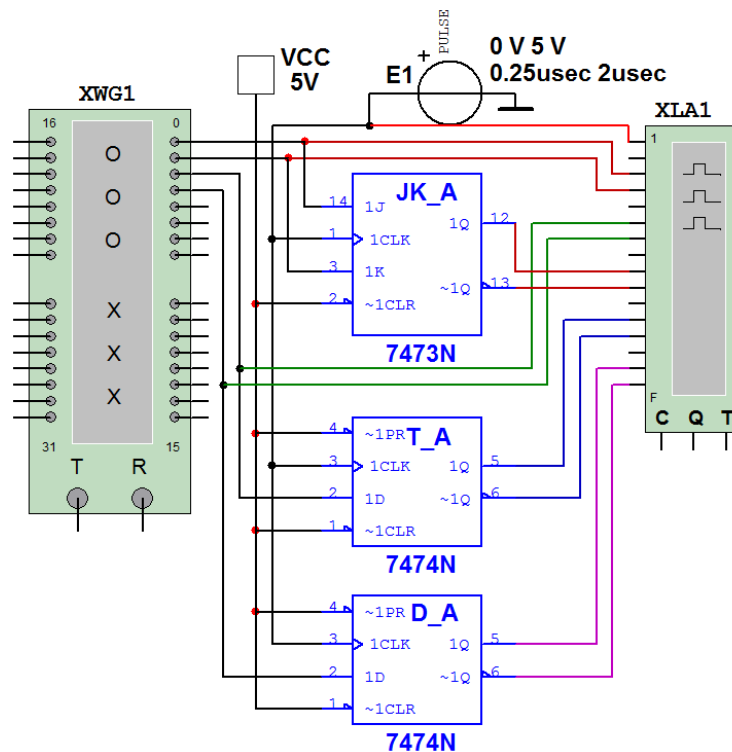


Рис. 7.19. Схема для испытания триггеров *JK*, *T* и *D*

Для формирования выходных сигналов генератор **XWG1** нужно запрограммировать, т. е. ввести в ячейки памяти кодовые комбинации из единиц и нулей согласно варианту (таблица 7.3).

Таблица 7.3

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. 7.20):

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. 7.20)). Перед моделированием выделить в окне генератора **XWG1** ячейку с адресом 0 начала счета и вывода сигналов.

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 $Q = 1$.

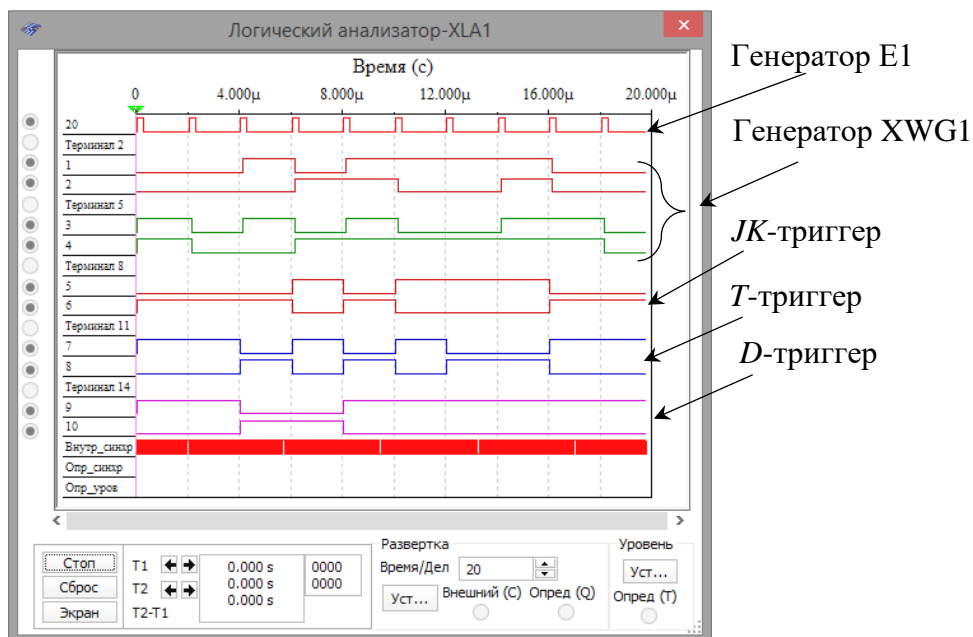


Рис. 7.20. Восемь ячеек памяти генератора с четырехразрядными кодовыми комбинациями

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.

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

Практические задания

Задание 1. Запустить среду разработки Multisim, собрать на рабочем поле среды Multisim схему для испытания *дешифратора* **DC** (рис. 7.21) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

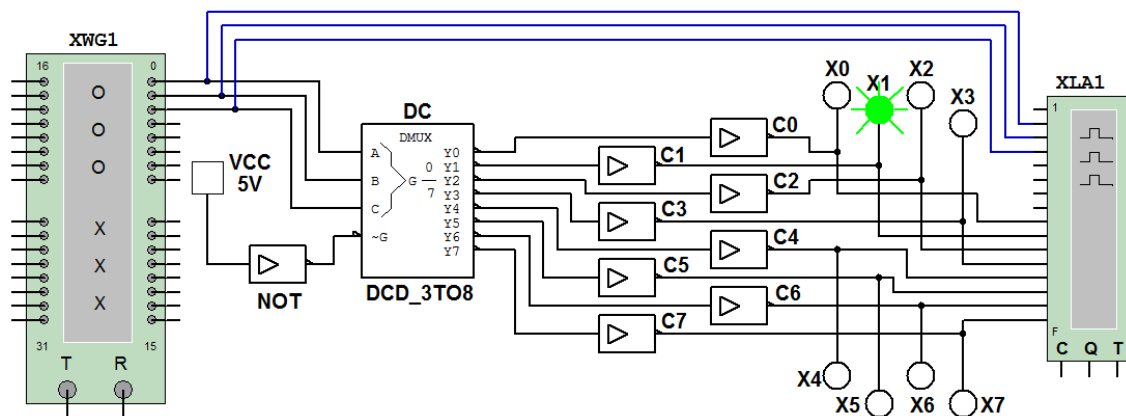


Рис. 7.21. Схема для испытаний дешифратора

Схема содержит интегральный *дешифратор* **DC** (*decoder*) 3×8 , имеющий 3 информационных входа **A**, **B** и **C** (для кода 4–2–1), 8 выходов (**Y0**, ..., **Y7**) 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 – нулевые (см. рис. 7.22, справа). В зависимости от входного двоичного кода, например 001, на выходе **DC** появляется сигнал 1 только на одной (второй, см. рис. 7.21) 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**, ..., **C7**;

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. 7.22.

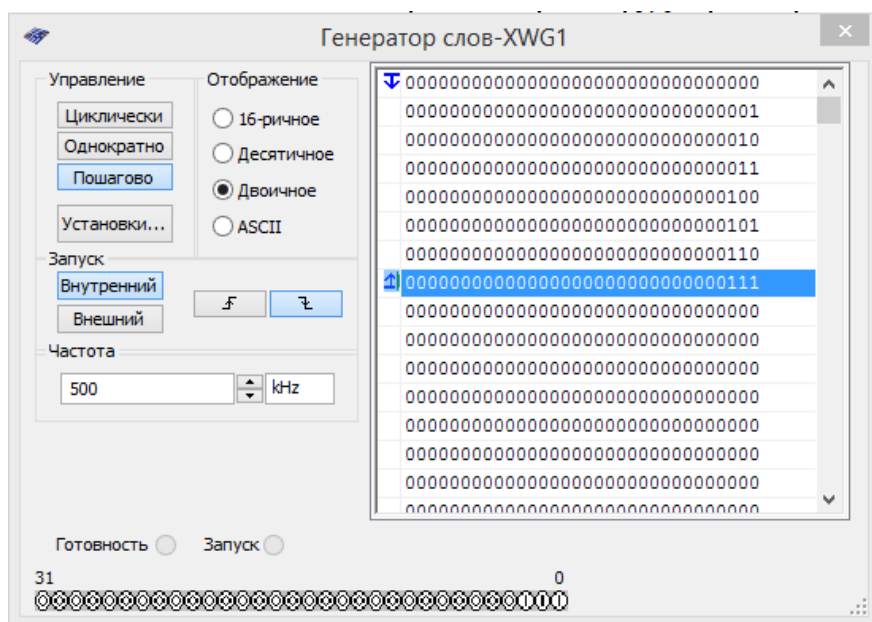


Рис. 7.22. Логический генератор

На экран логического анализатора **XLA1** (рис. 7.22) выводятся временные диаграммы как трех входных (**A**, **B**, **C**), так и восьми (**Y0**, **Y1**, ..., **Y7**) выходных сигналов при пошаговом режиме генератора **XWG1**.

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

Запустить программу моделирования дешифратора. Щелкая мышью на кнопке «Пошагово» генератора **XWG1**, последовательно подавать на вход дешифратора логические слова.

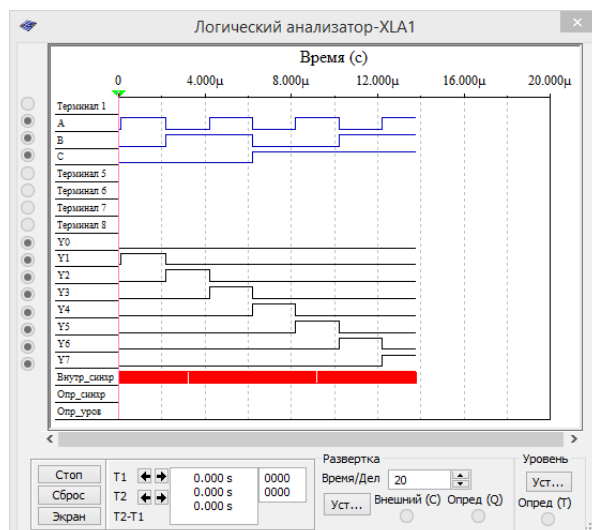


Рис. 7.23. Логический анализатор

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. 7.4 переключений (функций $Y_i = (A_i B_i C_i; G_i)$) на выходах дешифратора **ДС 3×8**.

Таблица 7.4

Переключение на выходах дешифратора **ДС 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. 7.24) и установить в диалоговых окнах компонентов их параметры или режимы работы.

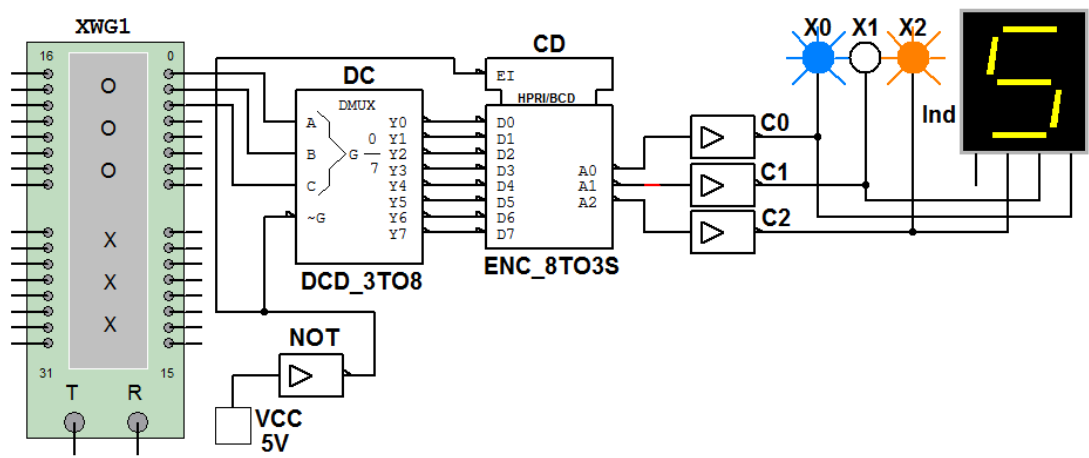


Рис. 7.24. Схема для испытаний шифратора

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. 7.5).

Таблица 7.5

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. 7.24) 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. 7.25) и установить в диалоговых окнах компонентов их параметры или режимы работы.

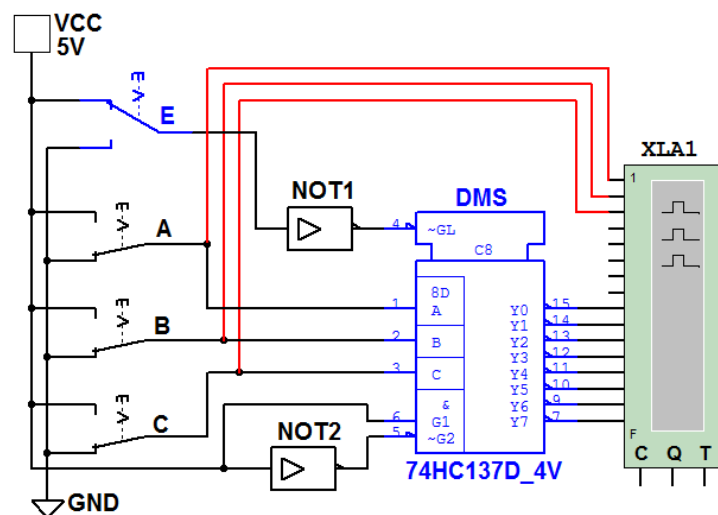


Рис. 7.25. Схема для испытаний демультиплексора

Демультиплексор DMS (логические микросхемы CMOS\74HC_4V – 74HC137D_4V) 1×8 (из 1 в 8) 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.

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. 7.25) 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 мультимплексор превратится в дешифратор.

Задание 4. Собрать на рабочем поле среды Multisim схему (рис. 7.26) для испытания *мультимплексора MS 8×1 (из 8 в 1) (Логические микросхемы TTL\74STD\74151N)* и установить в диалоговых окнах компонентов их параметры или режимы работы.

Мультимплексор 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 мультимплексор может принимать информацию. Если предположить, что к входам D0, D1, ..., D7 мультимплексора MS присоединено 8 источников цифровых сигналов – генераторов последовательных двоичных слов, то байты от любого из них можно передавать на выход Y.

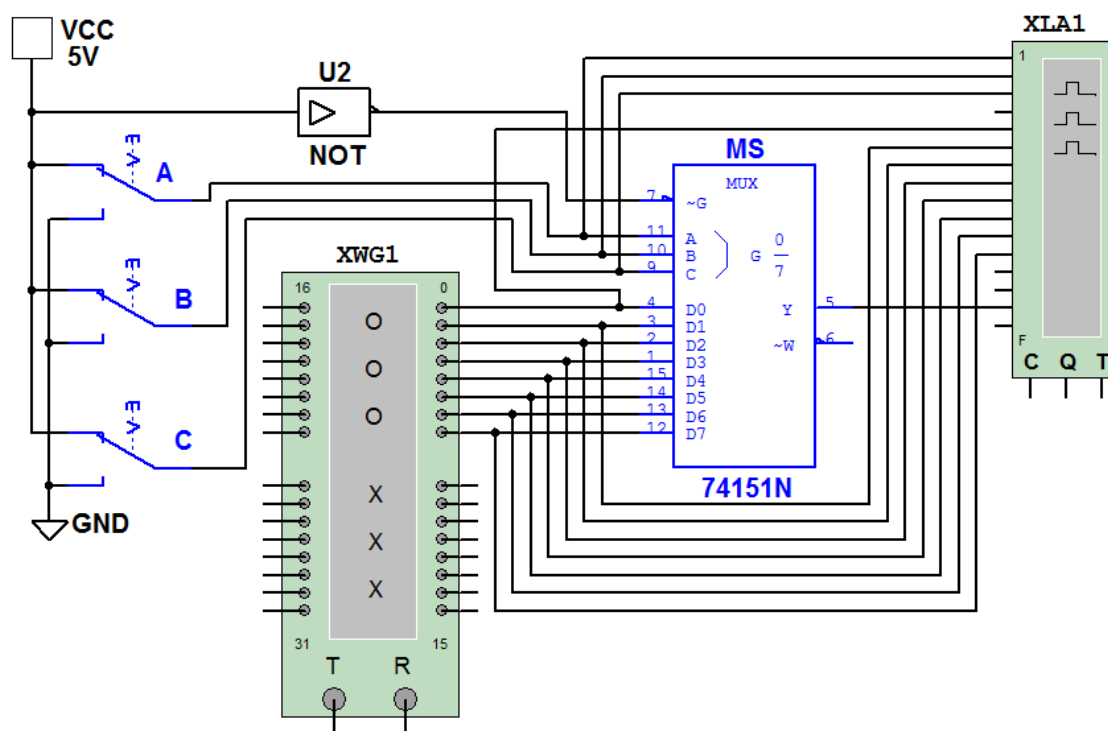


Рис. 7.26. Схема для испытаний мультиплексора

Скопировать схему (рис. 7.26) на страницу отчета.

Для иллюстрации работы мультиплексора MS запишем в ячейки памяти генератора XWG1 произвольные 8-разрядные кодовые слова (рис. 7.27), а с помощью ключей А, В, С сформируем управляющий сигнал **111**.

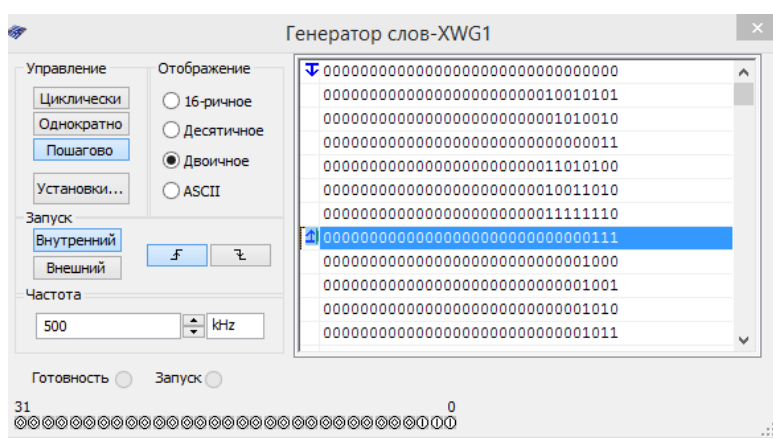


Рис. 7.27. Генератор XWG1 с произвольными 8-разрядными кодовыми словами

Последовательно щелкая мышью на кнопке «Пошагово» генератора XWG1 и при $G = 1$ поступающие на вход D7 мультиплексора

байты (сигнал 01001110) с 8-го разряда (на рис. 7.27, слева 8-й разряд показан стрелкой) логических слов генератора XWG1 передаются на выход Y и на вход анализатора (см. рис. 7.28).

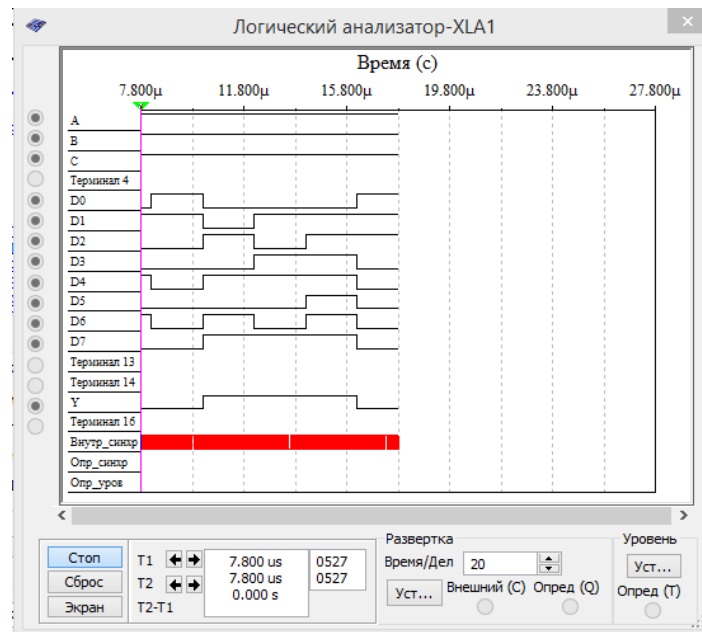


Рис. 7.28. Анализатор XLA1 с произвольными 8-разрядными кодовыми словами

Если ключ **A** установить в нижнее положение (сформировав тем самым адресный код 011), то с входа **D3** на выход **Y** мультиплексора будут поступать байты 4-го 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. 7.27, слева).

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.

Задание 4. Собрать на рабочем поле среды Multisim схему для испытания *демультиплексора DMS 1×16 (из 1 в 16)* (рис. 7.29) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему (рис. 7.29) в отчет.

В исследуемой модели демультиплексора соответствующий активный выход имеет низкий логический уровень (рис. 7.29), поэтому пробник на этом выходе не светится. Так, при подаче последней кодовой комбинации 1111 на вход демультиплексора не светится пробник **X16**, так как активным является выход **15** (см. рис. 7.31).

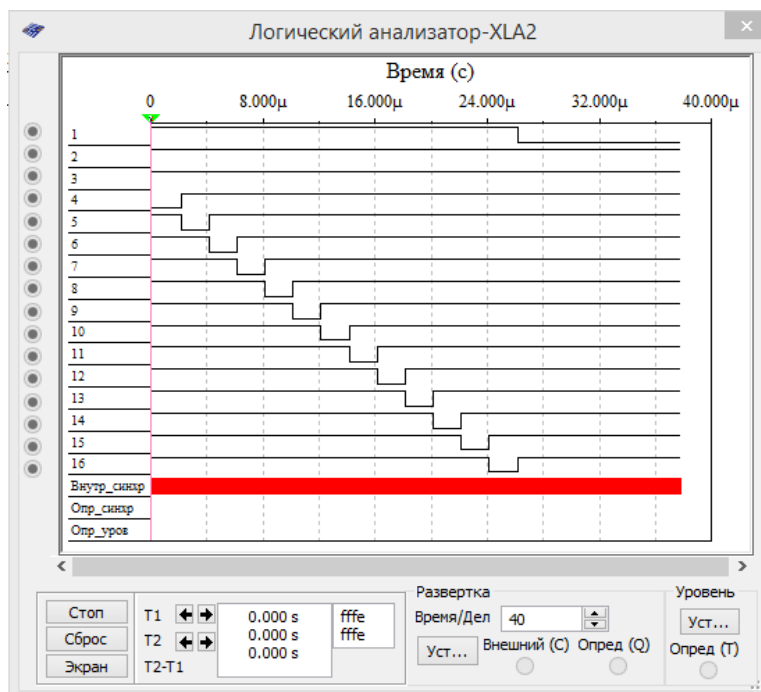


Рис. 7.31. Показания логического анализатора при исследовании демультиплексора

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

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.

7.4. ЗАПОМИНАЮЩИЕ УЗЛЫ. РЕГИСТРЫ

Практические задания

Задание 1. Запустить среду разработки Multisim, собрать на рабочем поле среды Multisim схему для испытания *универсального регистра сдвига* (рис. 7.32) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

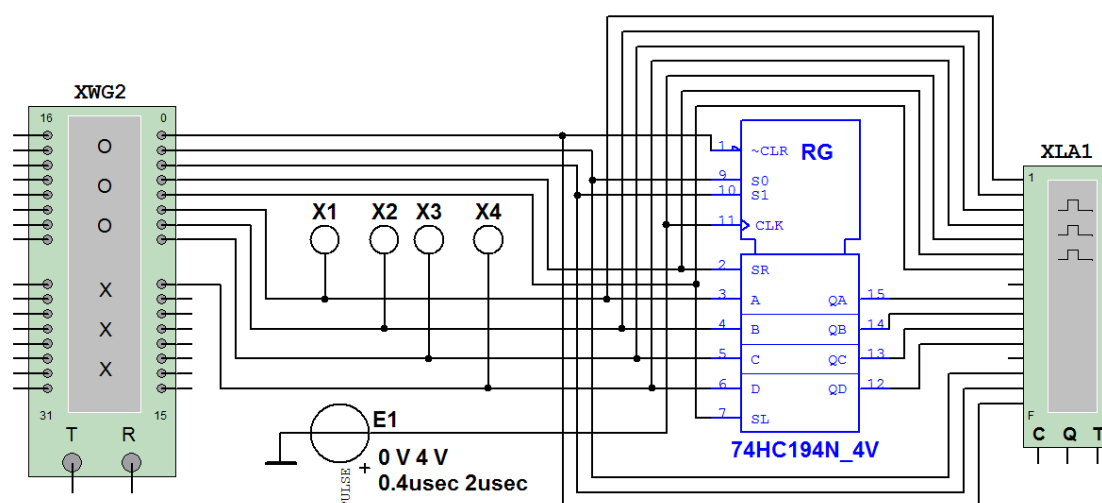


Рис. 7.32. Схема для испытания универсального регистра сдвига

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 $\overline{\text{CLR}}$ 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 7.6.

План исследования параллельного регистра сдвига

Входы										Выходы			
Сброс	Старт	Режим		Последова- тельный вход		Параллельный вход							
$\overline{\text{CLR}}$	$\overline{\text{CLX}}$	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, когда триггеры находятся в режиме хранения.

Руководствуясь схемой соединения генератора XWG1 с регистром (см. рис. 7.32), при записи чисел в ячейки памяти генератора в младший разряд 9-разрядных чисел нужно заносить значение сигнала $\overline{\text{CLR}}$: 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 $\overline{\text{CLK}}$ 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.

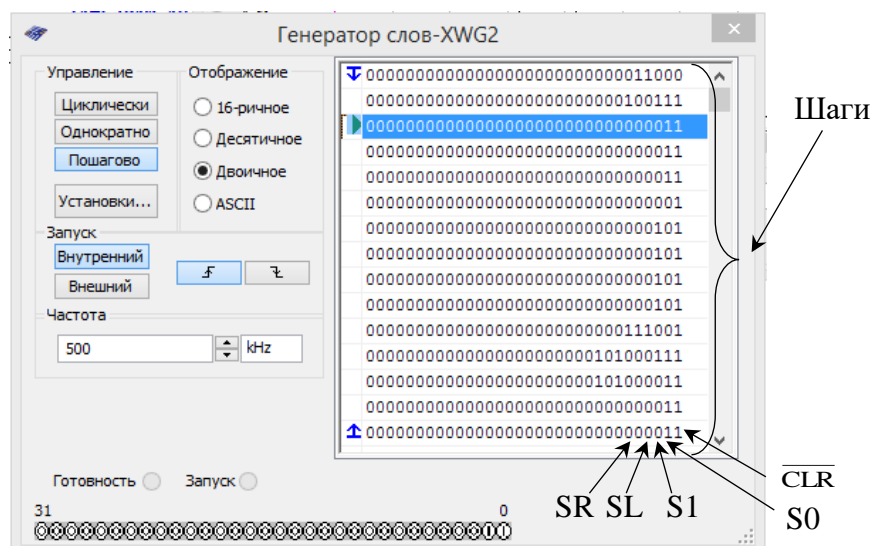


Рис. 7.33. Запись 9-разрядных кодовых комбинаций в 15 ячеек памяти генератора

В качестве примера на рис. 7.33 приведена запись 9-разрядных кодовых комбинаций в 15 ячеек памяти генератора XWG1, а на рис. 7.34 – 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 $\overline{\text{CLR}} = 0$ (см. первую строку табл. 7.6 и рис. 7.34) на всех входах и выходах регистра установились нулевые значения. На втором шаге при $\overline{\text{CLR}} = 1$, $\text{SR} = 0$, $\text{SL} = 0$ (разрешение записи числа в регистр), $\text{S0} = 1$ и $\text{S1} = 1$ происходит загрузка 4-разрядного двоичного числа DCBA = 0001 в регистр.

При задании направления сдвига данных влево ($\text{S0} = 1$ и $\text{S1} = 0$, такт или шаг 3) сигнал 0001 выводится на выходы: $\text{QD} = 0$, $\text{QC} = 0$, $\text{QB} = 0$ и $\text{QA} = 1$. С приходом очередного тактового импульса (шаги 4, 5 и 6) происходит перезапись (сдвиг) содержимого триггера каждого разряда в соседний разряд (от разряда A к разряду D) без изменения порядка следования единиц и нулей. По окончании шестого тактового импульса на

выходе устанавливается число 1000 (см. рис. 7.34). 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. 7.34) 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.

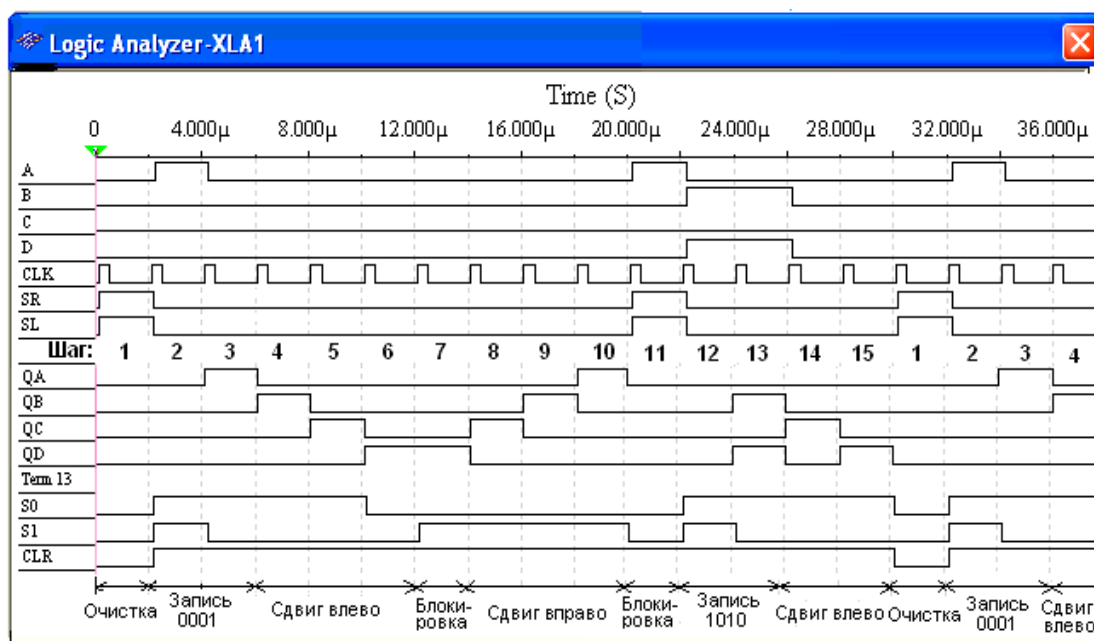


Рис. 7.34. Моделирование параллельного регистра в виде временных диаграмм сигналов

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. 7.34). 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, ..., 10, на рис. 7.33) 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 11-го 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. 7.35) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему в отчет.

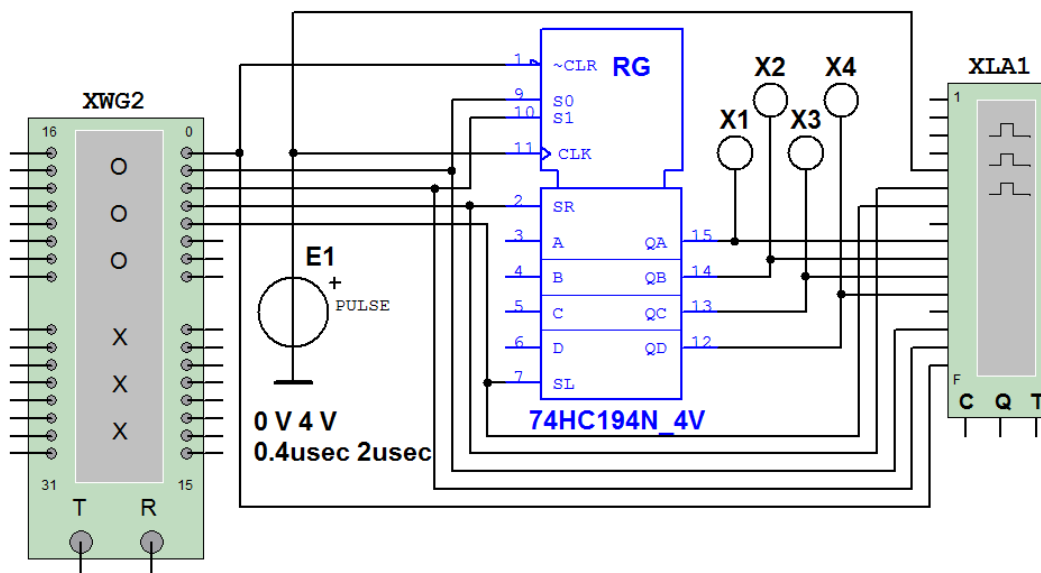


Рис. 7.35. Схема для испытания последовательного регистра сдвига

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. 7.36). Регистр последовательно сдвигает влево эти сигналы от QA к QD , на выходе QD они теряются (см. шаги 3, ..., 9 на рис. 7.36).

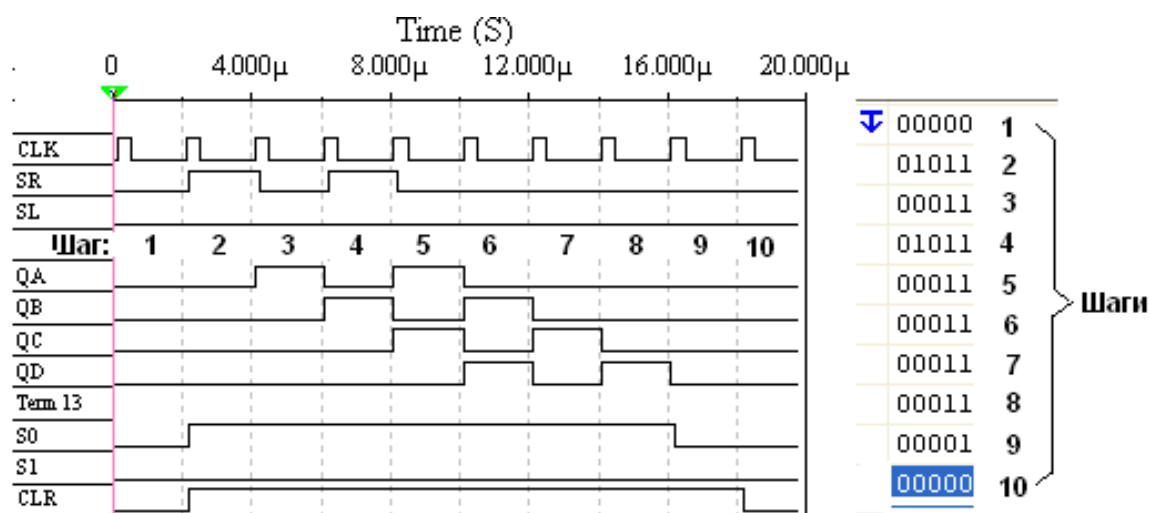


Рис. 7.36. Моделирование последовательного регистра сдвига влево в виде временных диаграмм сигналов

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, рис. 7.37).

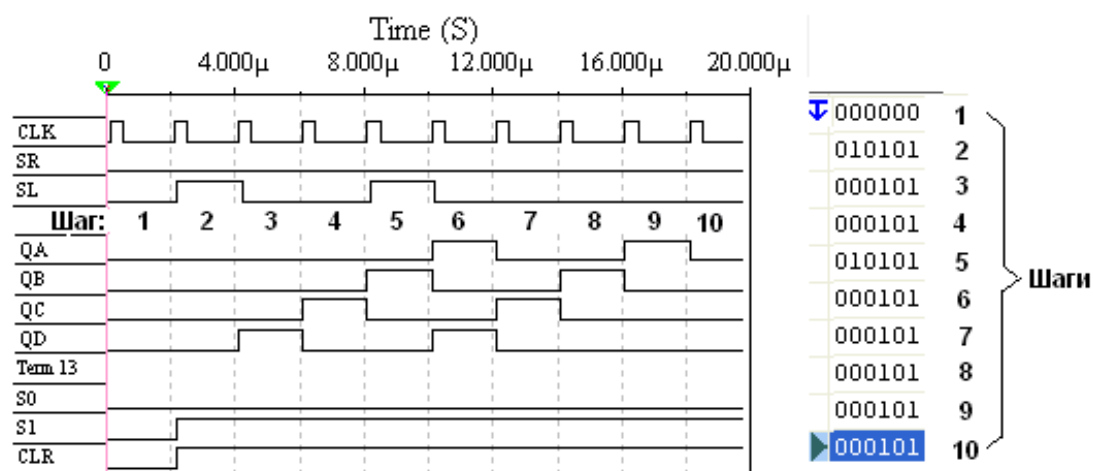


Рис. 7.37. Моделирование последовательного регистра сдвига вправо в виде временных диаграмм сигналов

Задание 4. Составить план исследования последовательного регистра 74HC194_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. 7.36) и вправо (см. рис. 7.37).

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.

7.5. ЗАПОМИНАЮЩИЕ УЗЛЫ. СЧЕТЧИКИ

Практические задания

Задание 1. Запустить среду разработки Multisim, собрать на рабочем поле среды Multisim схему для испытания *синхронного двоичного счетчика* (рис. 7.38) и 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. 7.38) включен синхронный двоичный 4-разрядный счетчик 74НС161, к входу \overline{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.

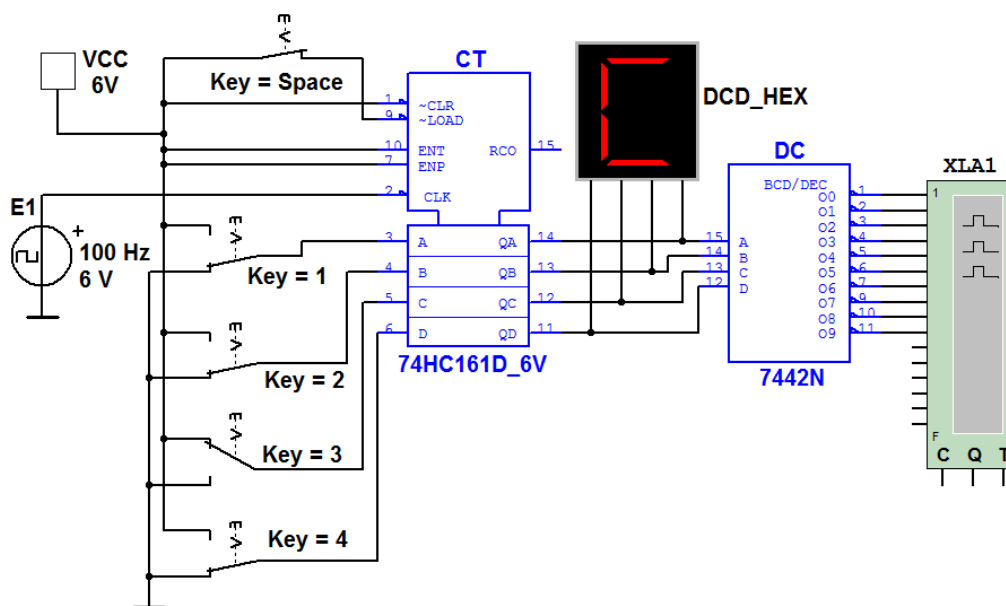


Рис. 7.38. Схема для испытания синхронного двоичного счетчика

К входам **A**, **B**, **C** и **D** счетчика **CT** подключен 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 **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 11-го тактового импульса и появляются вновь только с приходом 17-го импульса (рис. 7.39).

Разомкнуть ключ **Space**. 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. (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.

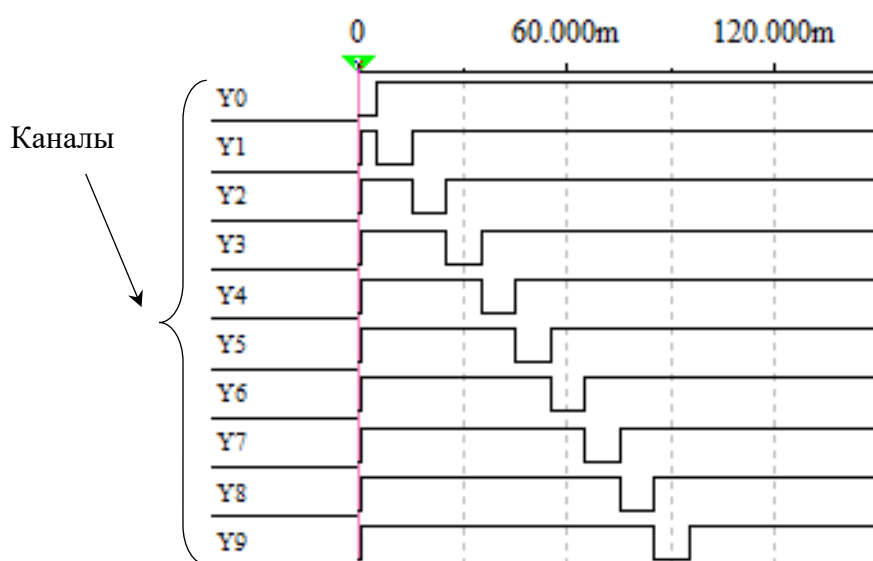


Рис. 7.39. Результаты моделирования синхронного суммирующего счетчика

Скопировать на страницу отчета результаты моделирования синхронного суммирующего счетчика (см. рис. 7.39).

Задание 3. Собрать на рабочем поле среды Multisim схему для испытания *реверсивного двоичного счетчика* (рис. 7.40) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

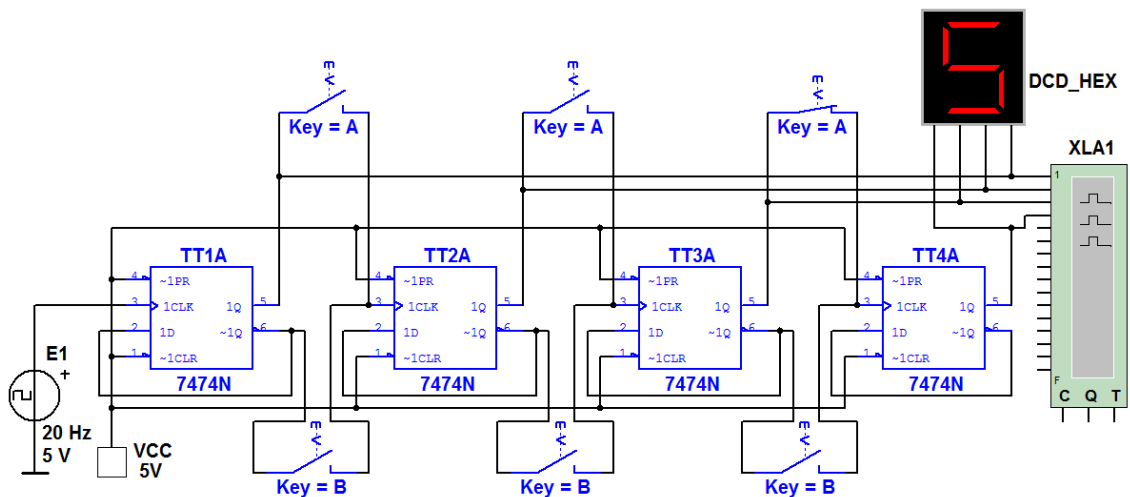


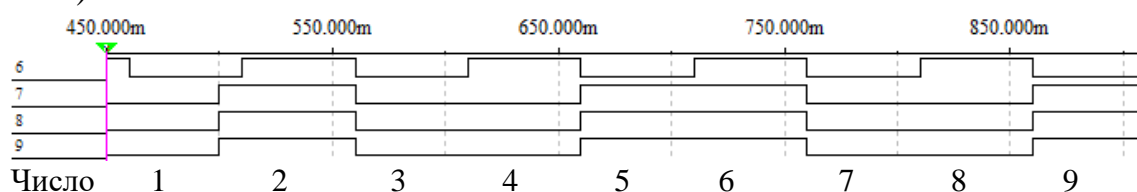
Рис. 7.40. Схема для испытания реверсивного двоичного счетчика

В схеме реверсивного двоичного счетчика (рис. 7.40) с помощью групп ключей **A** и **B** осуществляется замыкание или размыкание выходов высокого или низкого уровня предыдущего триггера с входами $\overline{1PR}$ или $\overline{1CLR}$ 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 $V = 5$ В, частоту таймера $f_a = 2$ кГц, число импульсов, приходящихся на одно деление, **Clocks/div** = 60. Разомкнуть ключи **B** и замкнуть ключи **A**. Запустить программу моделирования счетчика.

При высвечивании числа 15 на 7-сегментном индикаторе щелкнуть мышью на кнопке **Stop** (остановки моделирования) и скопировать окно анализатора с результатами моделирования на страницу отчета (см. верхнюю часть рис. 7.41).

Разомкнуть ключи **A** и замкнуть ключи **B**. Щелкнуть мышью на 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. 7.41).



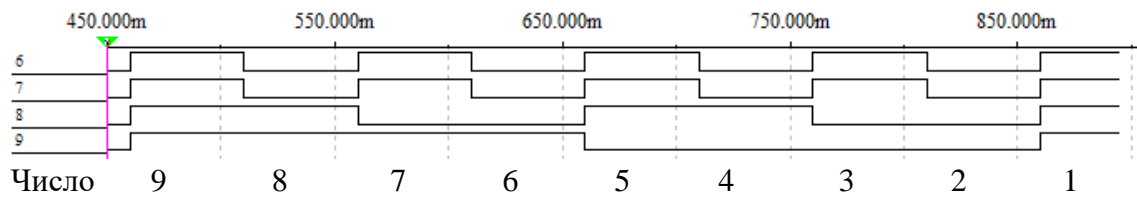


Рис. 7.41. Результаты моделирования реверсивного двоичного счетчика

Задание 4. Собрать на рабочем поле среды Multisim схему для испытания *десятичного счетчика* (рис. 7.42) и установить в диалоговых окнах компонентов их параметры или режимы работы. Скопировать схему на страницу отчета.

Результаты моделирования 4-разрядного двоичного счетчика (см. рис. 7.39 и 7.40) 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.

В функциональной схеме десятичного счетчика (рис. 7.42), 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 с встроенными логическими элементами И (**AND**), ИЛИ (**OR**) и НЕ (**NOT**).

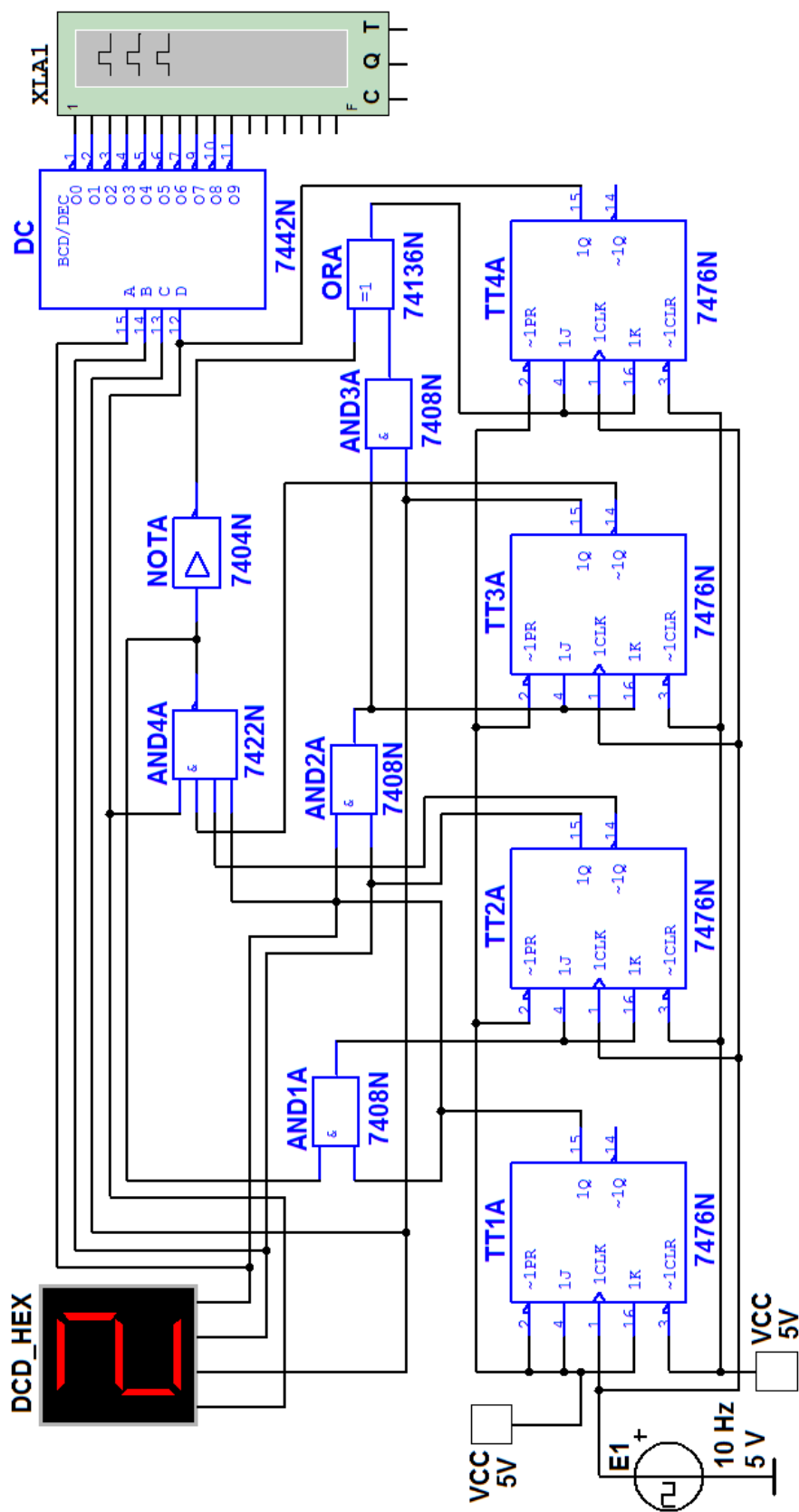


Рис. 7.42. Схема для испытания десятичного счетчика

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.