

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
УНИВЕРСИТЕТ ИТМО

П. В. Кустарев, Р. И. Попов,
С. В. Быковский, В. Ю. Пинкевич

Цифровая схемотехника

Методические указания к лабораторным работам



Санкт-Петербург

2017

П. В. Кустарев, Р. И. Попов, С. В. Быковский, В. Ю.Пинкевич . Цифровая схемотехника. Методические указания к лабораторным работам. — СПб.: Университет ИТМО, 2017. — 34 с.

Содержание

Введение	5
Рекомендуемая литература	6
Требования к оформлению отчетов	7
Основы работы в среде LTspice	9
1 Лабораторная работа №1. «Введение в проектирование цифровых интегральных схем»	19
1.1 Цели работы	19
1.2 Указания к выполнению работы	19
1.3 Порядок выполнения работы	19
1.4 Задания по вариантам	21
2 Лабораторная работа №2. «Комбинационные схемы»	22
2.1 Введение	22
2.2 Цели работы	22
2.3 Указания к выполнению работы	22
2.4 Порядок выполнения работы	23
2.5 Задания по вариантам	24
3 Лабораторная работа №3. «Триггеры и схемы последовательностного типа»	25
3.1 Цели работы	25
3.2 Указания к выполнению работы	25
3.3 Порядок выполнения работы	26
3.4 Задания по вариантам	27
4 Лабораторная работа №4. «Цифровые автоматы»	28
4.1 Введение	28
4.2 Цели работы	28

4.3	Указания к выполнению работы	28
4.4	Порядок выполнения работы	29
4.5	Задания	29
4.5.1	Задания 1-й части	29
4.5.2	Задания 2-й части	31
4.5.3	Требования к конечным автоматам	33
4.5.4	Задания по вариантам	34

Введение

Лабораторные работы данного курса посвящены изучению полупроводниковых приборов и простейших цифровых схем на их основе – базовых операционных элементов, которые являются элементной базой для проектирования более сложных устройств (микропроцессоров, периферийных контроллеров и др.).

Лабораторные работы охватывают следующий список тем:

1. Лабораторная работа №1. «Введение в проектирование цифровых интегральных схем».
2. Лабораторная работа №2. «Комбинационные схемы»
3. Лабораторная работа №3. «Триггеры и схемы последовательностного типа»
4. Лабораторная работа №4. «Цифровые автоматы».

Для исследования работы электрических схем инженер-проектировщик может создать прототип устройства на макетной плате или воспользоваться ручными расчётами на бумаге. Однако, при разработке интегральной микросхемы создание прототипа может быть крайне дорогостоящим и длительным процессом. Сложность таких схем зачастую не позволяет провести требуемые расчеты вручную. Поэтому, при разработке большинства микросхем, а также сложных печатных плат, повсеместно применяется компьютерное моделирование. Наиболее известной системой аналогового моделирования является программа SPICE и её многочисленные наследники.

Лабораторные работы будут выполняться в среде моделирования LTspice, которая относится к классу SPICE-подобных симуляторов электронных схем общего назначения.

Рекомендуемая литература

1. Дэвид М. Харрис, Сара Л. Харрис. Цифровая схемотехника и архитектура компьютера. 2-е изд.
2. Жан М. Рабаи, Ананта Чандракасан, Боривож Николич. Цифровые интегральные схемы. Методология проектирования. – 2-е изд. – М.: «Вильямс», 2007.
3. David Harris, Sarah Harris. Digital Design and Computer Architecture, 2d edition.
4. N. H.E. Weste, D.M. Harris. CMOS VLSI design: A circuits and systems perspective.
5. Угрюмов Е.П. Цифровая схемотехника. Уч. пособие для ВУЗов. 2-ое изд. – СПб.: БХВ – Петербург, 2007, 800 с.
6. Хоровиц П., Хилл У., Искусство схемотехники./ Пер. с англ. 6-е изд. – М.: Мир, 2003. – 704 с.
7. Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic. Digital Integrated Circuits Prentice Hall; Prentice Hall 2 edition (January 3, 2003)
8. Baker, R. Jacob. CMOS: Circuit Design, Layout, and Simulation, Third Edition. Wiley-IEEE, 2010. <http://CMOSedu.com>
9. Weste, Neil H. E. and Harris, David M. CMOS VLSI Design: A Circuits and Systems Perspective, Fourth Edition. Boston: Pearson/Addison-Wesley, 2010.
10. Точки, Рональд, Дж., Уидмер, Нил, С. Цифровые системы. Теория и практика. – 8-е изд.. – М.: «Вильямс», 2004.
11. Сохор Ю.Н. Моделирование устройств в пакете LTspice/SwCAD. Учебно-методическое пособие. Псковск. гос. политехн. ин-т. – Псков: Издательство ППИ, 2008. – 165 с.

Требования к оформлению отчетов

Отчет выполняется в виде самостоятельного документа. Материал, изложенный в отчете, должен быть понятным без дополнительных комментариев со стороны исполнителей.

Отчет должен содержать:

- титульный лист, на котором указываются:
 - название университета;
 - название кафедры;
 - название дисциплины;
 - номер и тема лабораторной работы;
 - вариант лабораторной работы;
 - фамилия, инициалы и номер группы каждого исполнителя;
 - фамилия и инициалы преподавателя;
 - текущий год;
- содержание;
- цель и задачи работы;
- задание в соответствии с вариантом;
- отчет о выполнении заданий работы; отчет по каждому заданию содержит:
 - изучаемую схему и схемы новых иерархических блоков, использованных в схеме;
 - результаты моделирования (временная диаграмма);
 - комментарии результатов (минимум 2 предложения);
- общий вывод по работе (какие знания и навыки получены).

Требования к оформлению:

- отчет выполняется как текстовый документ в соответствии с ГОСТ 2.105-95;

- шрифт Times New Roman 12-14 pt, межстрочный интервал 1-1,5, поля с краев листа – не менее 2 см;
- сквозная нумерация страниц;
- обязательны нумерация и подписи к рисункам и таблицам, а также ссылки на них в тексте отчета;
- схемы и временные диаграммы должны быть темными на светлом фоне; если наложение временных диаграмм нескольких сигналов мешает их однозначному восприятию, они должны разноситься на отдельные координатные сетки;
- в распечатанном отчете линии на схемах и временных диаграммах должны быть четко видны.

Основы работы в среде LTspice

О моделировании в SPICE

SPICE (Simulation Program with Integrated Circuit Emphasis) – программа для моделирования поведения интегральных схем и схем на печатных платах. Изначально разработана в Калифорнийском университете в Беркли в начале 1970-х гг. В дальнейшем на ее основе были созданы другие программы, одной из которых является LTspice.

Необходимость моделирования вызвана тем, что создать дешевый прототип интегральной схемы практически невозможно – для этого ее надо фактически изготовить, что является очень затратным мероприятием. SPICE позволяет промоделировать работу схемы в разных условиях и выявить потенциальные проблемы до того, как она будет выпущена.

SPICE содержит модели различных электронных компонентов и элементов схем – транзисторов, резисторов, конденсаторов, источников напряжения, усилителей и т.п. Соединяя компоненты, можно построить произвольную схему, а затем подвергнуть ее анализу одним из поддерживаемых способов: передаточная функция, анализ по постоянному току, анализ по переменному току и др.

Ограничения области применимости SPICE

Несмотря на то, что SPICE во многих случаях может заменить прототипирование, часто результаты моделирования могут оказаться недостаточно точными или даже совсем не совпадать с работой реальной схемы. Реальные электронные компоненты являются сложными устройствами, работа которых зависит от множества параметров. Результаты моделирования точны настолько, насколько точны модели компонентов.

Модели схем в SPICE свободны от шума, перекрёстных помех, паразитных емкостей и др., если только они явно не внесены в модель проектировщиком. Например, сигнал в несколько пикоампер может корректно управлять схемой в SPICE-модели, но в физическом мире будет полностью забит шумом и паразитными цепями.

SPICE-моделирование не подходит для исследования излучающих и принимающих схем (антенн). Для задач такого рода используются системы, построенные на численном решении уравнений Максвелла.

SPICE не является лучшим средством для предсказания поломки компонентов. Моделирование не выдаст никакого сигнала предупреждения в случае, если будет превышено максимально допустимое значение тока или напряжения для какого-то из компонентов. Например, проектировщик может не заметить тока в несколько ампер на одной из цепей при моделировании схемы, но после изготовления такая схема при включении сгорит.

О программе LTspice

Загрузить программу LTspice можно с сайта компании Linear Technology по адресу: <http://www.linear.com/designtools/software/>. Программа является бесплатной. Доступны версии под Windows и Mac OS, под Linux LTspice можно запустить в Wine. Версия для Mac OS имеет некоторые отличия от версии для Windows.

LTspice имеет встроенную справку с описанием основных приемов работы. Также, в интернете можно найти множество уроков и статей по использованию LTspice. Рекомендуется ознакомиться со следующими видеоуроками: http://cmosedu.com/videos/LTspice/LTspice_videos.htm.

Модели транзисторов, поддерживаемые LTspice (BSIM4), имеют множество настраиваемых параметров, которые позволяют получить характеристики транзистора, характерные для конкретного техпроцесса, например 40 nm TSMC, 32 nm Intel и др. Как правило, производители полупроводников извлекают значения этих параметров путем измерения характеристик реальных транзисторов, изготовленных в рамках разработки и настройки технологического процесса на фабрике. Затем полученные параметры передаются разработчикам цифровых схем, чтобы те могли промоделировать свои проекты в SPICE-симуляторе. BSIM4-параметры для некоторых фабрик доступны на сайте MOSIS: <https://www.mosis.com/requests/test-data>.

В рамках данного курса будут использоваться модели транзисторов для техпроцесса 90 нм, полученные путем моделирования. Цифра в названии технологии, как правило, означает минимальную длину канала транзистора. Т.к. увеличение длины канала транзистора в цифровых схемах не дает никакого положительного эффекта, все транзисторы в рамках этого курса будут иметь длину канала равную 90 нм. Минимальная допустимая ширина канала обычно в два раза больше минимальной допустимой длины, т.е. для техпроцесса 90 нм минимальная ширина канала равна 180 нм. Увеличение ширины канала увеличивает его проводимость и емкости между затвором и каналом.

Создание инвертора

Рассмотрим основные принципы работы в среде LTspice и познакомимся с КМОП-схемотехникой на примере простейшего цифрового вентиля – инвертора.

Подготовка к работе

Создайте рабочий каталог для сохранения файлов проекта. Загрузите файл с параметрами транзисторов 90nm_bulk.txt (предоставляется преподавателем) и сохраните его в своем рабочем каталоге. В этом же каталоге следует сохранять схемы (файлы *.asc, *.asy), выполняемые в рамках курса.

Запустите программу LTspice. Окно программы представлено на рис. 1.

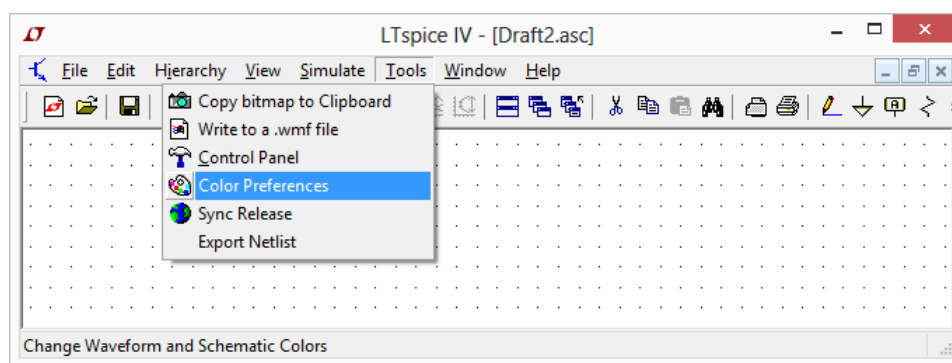


Рисунок 1: Окно программы LTspice

В меню Tools → Color Preferences (рис. 1) можно настроить цвета, в которых изображаются рабочее поле и схема.

Включить отображение сетки в рабочем поле можно с помощью View → Show Grid.

Создание и симуляция схемы

Создайте новую схему (File → New Schematic) и сохраните ее файл в своем рабочем каталоге. Добавьте к схеме SPICE-директиву, подключающую параметры для транзисторов (Edit → SPICE Directive, рис. 2). Помните, что директивы SPICE начинаются с точки. Разместите директиву на схеме (рис. 3).

На рис. 4 показана схема тестирования инвертора – простейшего цифрового вентиля. Схема включает: КМОП-инвертор из двух транзисторов (PMOS и NMOS), источник питания инвертора (VDD1), источник тестового входного напряжения (V1) и нагрузку выхода инвертора (R1 и C1). Нагрузка необходима, чтобы моделировать реальную ситуацию, когда выход инвертора управляет другими вентилями.

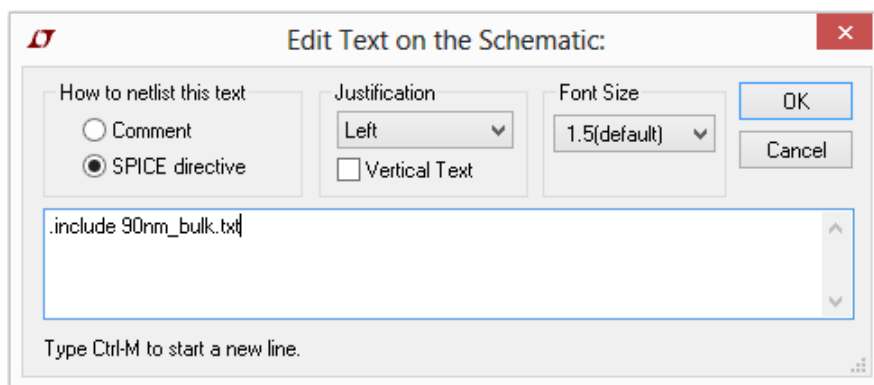


Рисунок 2: Окно ввода директивы

.include 90nm_bulk.txt

Рисунок 3: Директива, вставленная в схему

Постройте схему. Для этого добавьте на схему транзисторы pmos4 и nmos4, компоненты voltage, res, cap (Edit → Component) и землю (Edit → Place GND). Сориентируйте элементы нужным образом (Edit → Rotate и Edit → Mirror), а затем соедините проводами (Edit → Draw Wire). Проводам можно присваивать имена с помощью Edit → Label Net (просто ввести имя, тип порта оставить None). Если присвоить двум проводам в разных местах схемы одинаковое имя, это будет эквивалентно их соединению. На схеме тестирования инвертора таким образом соединены все фрагменты провода с именем VDD.

Отредактируйте параметры компонентов в соответствии со схемой на рис. 4. Для редактирования параметров необходимо щелкнуть на компоненте правой кнопкой мыши.

Для транзисторов необходимо ввести длину и ширину (l и w) канала в нанометрах. Заметьте, что ширина канала PMOS-транзистора в два раза больше, чем ширина канала NMOS-транзистора. Это связано с тем, что подвижность дырок примерно в два раза ниже подвижности электронов. Следовательно, чтобы обеспечить одинаковую проводимость каналов, ширину PMOS-канала нужно сделать в два раза больше. Чтобы вывести параметры транзистора рядом с компонентом (как на рис. 4), необходимо настроить видимость параметров компонента: при зажатой клавише Ctrl щелкнуть правой кнопкой мыши на компоненте и отметить столбец Visible для поля Value2.

Для источника питания инвертора VDD1 задайте постоянное значение напряжения 1 В. Для источника напряжения V1 задайте параметры, как показано на рис. 5 (для этого в окне параметров необходимо нажать кнопку Advanced). Источник V1

генерирует прямоугольные импульсы (чередование «0» и «1»), которые будут поданы на вход инвертора для его тестирования. Заметьте, что буквы после чисел обозначают приставки кратных и дольных величин в системе СИ (k – кило-, n – нано-, p – пико-, f – фемто- и т.д.).

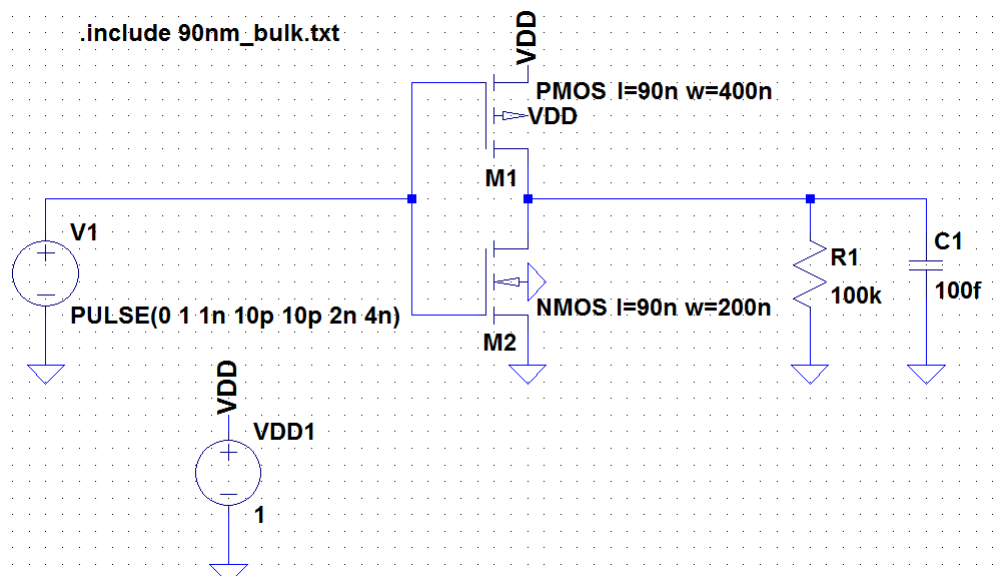


Рисунок 4: Схема тестирования инвертора

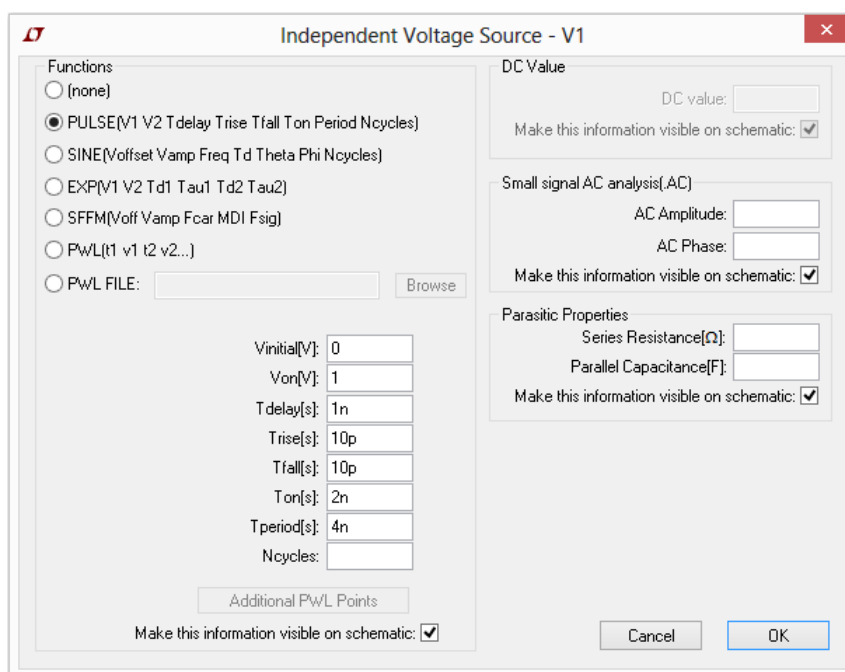


Рисунок 5: Окно настроек источника напряжения

Добавьте на схему настройки анализа переходного процесса (Edit → SPICE Analysis) с параметрами, как показано на рис. 6. Настройки задают время моделирования 8 нс и шаг моделирования 1 пс.

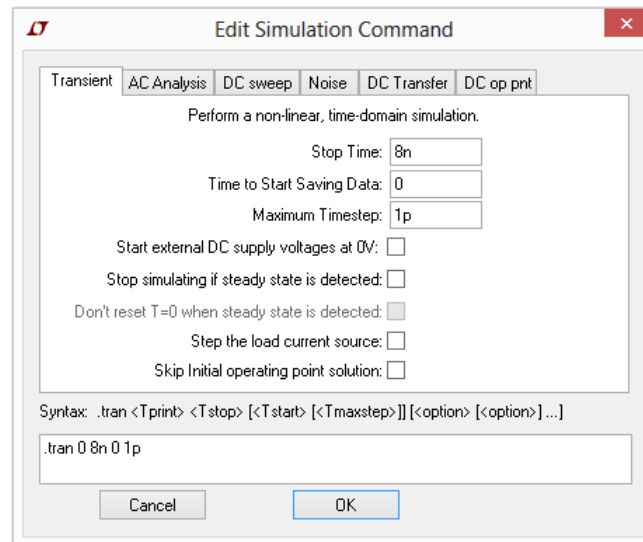


Рисунок 6: Окно настроек процедуры анализа переходного процесса

Запустите симуляцию (Simulate → Run). Появится окно для вывода временных диаграмм. Наведите курсор мыши на провод перед инвертором; курсор примет вид щупа осциллографа. Щелкните на проводе, чтобы добавить временную диаграмму входного напряжения инвертора на график. Аналогично добавьте диаграмму выходного напряжения. График должен иметь вид, как на рис. 7. Выходное напряжение инвертора противоположно входному, а значит, инвертор работает корректно.

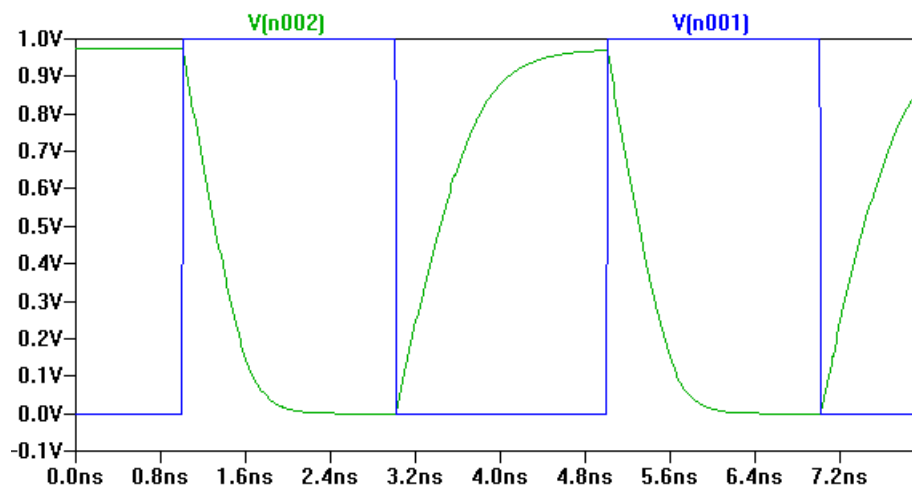


Рисунок 7: Результаты моделирования переходного процесса в инверторе

Создание иерархических элементов

Для разработки более сложных схем полезно будет иметь готовый инвертор в «библиотеке» и вставлять его в новые схемы как иерархический элемент.

Для создания библиотечного элемента требуется создать схему (File → New Schematic) и символ (File → New Symbol) с одинаковыми именами, например, inverter.

Постройте схему инвертора, как показано на рис. 8. У этого инвертора будет настраиваемый параметр – множитель для ширины канала транзисторов. Это позволит создавать инверторы разного размера.

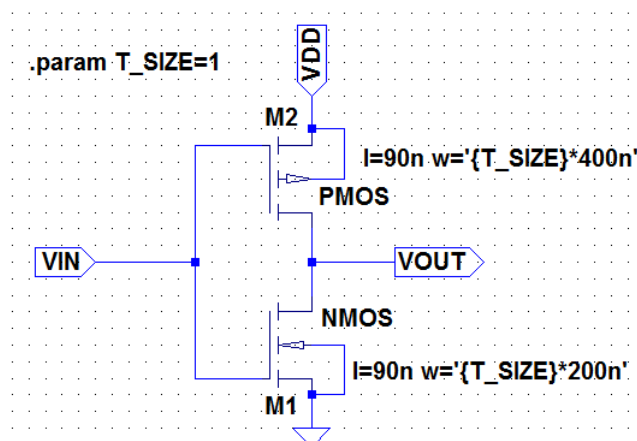


Рисунок 8: Схема инвертора

Для создания параметра поместите на схему SPICE-директиву «.param», как показано на рис. 8. У параметра задано значение по умолчанию, равное 1. Имя параметра можно использовать при задании параметров компонентов схемы. В нашем случае он использован при вводе ширины канала транзисторов. Заметьте, что выражение для ширины канала транзистора должно быть заключено в одиночные кавычки.

В схеме необходимо создать именованные порты ввода и вывода с помощью Edit → Label Net и выбора типа порта.

В качестве символов элементов следует использовать общепринятые обозначения элементов по стандартам ГОСТ, IEC или ANSI. Нарисуйте ANSI-символ инвертора, как показано на рис. 9. Для рисования используйте команды из меню Draw. Добавьте в символ порты (Edit → Add Pin/Port) с такими же названиями, как и в схеме элемента. Для порта выберите, надо ли отображать название, и если да, то с какой стороны от него будет отображаться точка электрического соединения.

Теперь инвертор можно вставлять в новые схемы как библиотечный элемент. Постройте схему тестирования инвертора с применением иерархического символа (рис. 10). Промоделируйте схему. Результаты должны совпасть с результатами моделирования предыдущей схемы.

Отредактируйте параметры инвертора. Для этого щелкните по символу правой кнопкой мыши и в строке PARAMS введите «T_SIZE=4». Промоделируйте схему

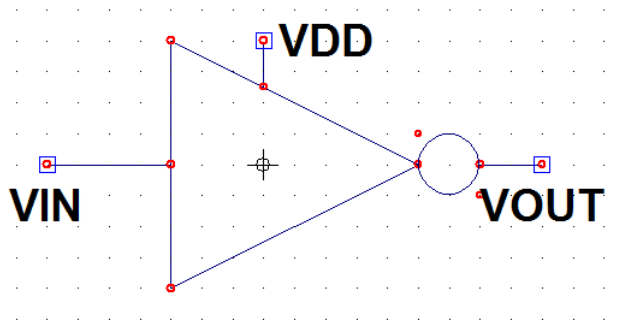


Рисунок 9: Символ инвертора

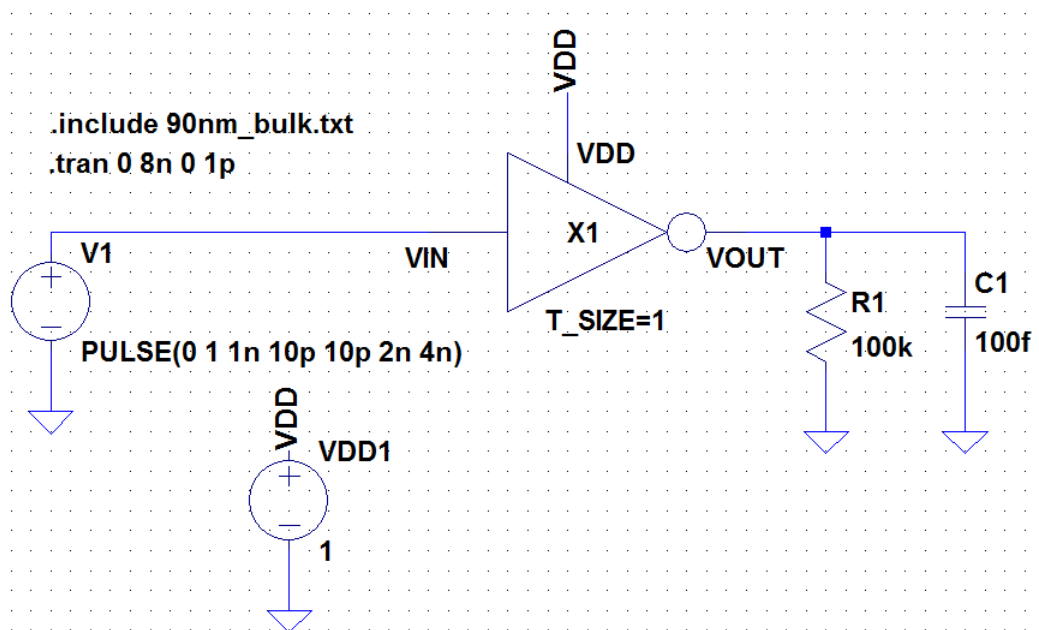


Рисунок 10: Схема тестирования инвертора с использованием разработанного символа

еще раз. Длительность переходных процессов при переключении инвертора должна уменьшиться.

Использование библиотеки цифровых элементов

Моделирование сложных цифровых устройств на уровне отдельных транзисторов требует больших вычислительных ресурсов. Однако, это не является необходимым, поскольку процессы в цифровых схемах могут быть адекватно представлены значительно более простыми моделями.

Для LTspice существуют специальные библиотеки, предназначенные для работы с цифровой логикой.

Стандартная библиотека Digital содержит модели некоторых вентилях и триггеров.

Библиотека 74HCT содержит модели основных вентилях, триггеров и некоторых БОЭ. Библиотека DVIEW содержит специальные элементы DVIEW5 и DVIEW10, которые масштабируют и сдвигают уровни напряжения входных сигналов. Так, к входам элемента DVIEW5 можно подключить 5 сигналов, изменяющихся в пределах от 0 до 5 В, а на выходах будет 5 сигналов, каждый из которых имеет диапазон изменения 0,5 В, но его нулевой уровень находится на 0 В, 1 В, 2 В, 3 В или 4 В соответственно. Это позволяет удобно отобразить несколько сигналов на одном графике без наложения.

Действия для установки библиотек 74HCT и DVIEW описаны на странице <http://spot.pcc.edu/~ghecht/LTspice.html> (на Windows для копирования файлов в системные каталоги могут потребоваться права администратора). Внимание! Возможно, установку файлов потребуется произвести не только в «C:\Program Files\LTC\LTspice...», но и в «C:\Users\username\Documents\LTspice...».

Для использования этих библиотек на схему надо добавить директивы «.lib 74hct.lib» и «.lib dview.lib». Элементы библиотек будут доступны при добавлении компонента на схему. 74HCT отображается в виде отдельного каталога, а элементы DVIEW находятся в каталоге Digital.

Напряжение питания VDD для элементов из библиотеки 74HCT равно 5 В. Для моделирования рекомендуется задавать максимальный шаг в 1 мкс (1u), а период изменения сигнала делать кратным 1 мс (1m). Примеры схемы с применением элементов библиотек 74HCT и DVIEW и временной диаграммы приведены на рис. 11 и рис. 12.

Внимание! Каждая библиотека содержит модели вентилях, триггеров и т.п. со своими рабочими уровнями напряжения и временами переключения. В связи с этим, в рамках одной схемы нельзя смешивать элементы разных библиотек.

Лабораторная работа №1. «Введение в проектирование цифровых интегральных схем»

1.1 Цели работы

- Получить базовые знания о принципах построения цифровых интегральных схем с использованием технологии КМОП
- Познакомиться с основными параметрами цифровых вентилях

1.2 Указания к выполнению работы

Необходимая теория: [1], главы 1-2.

При построении схем вентилях необходимо использовать КМОП-транзисторы с параметрами из файла 90mn_bulk.txt (см. раздел «Основы работы в среде LTspice»).

Если не указано иное, при измерении задержки распространения сигнала все входы тестируемой схемы должны управляться единичным инвертором, а все выходы должны быть нагружены единичными инверторами. Это нужно для того, чтобы промоделировать реальную ситуацию в микросхеме, когда входом вентиля управляет другой вентиль (т.е. источник с ограниченной мощностью, в отличие от идеального источника в симуляторе).

1.3 Порядок выполнения работы

1. Постройте схему вентиля согласно варианту задания.
2. Проведите моделирование работы схемы и определите задержку распространения сигнала через тестируемый вентиль.
3. Подключите на выход вентиля еще один аналогичный вентиль и измерьте значение задержки распространения сигнала через тестируемый вентиль.
4. Подключите на выход вентиля параллельно 4 аналогичных вентилях и также измерьте значение задержки через тестируемый вентиль.

5. Сделайте вывод о влиянии нагрузки на временные характеристики работы тестируемого вентиля и объясните причину их изменения.
6. Постройте схему, реализующую логическую функцию в заданном логическом базисе, согласно варианту задания.
7. Измерьте максимальную и минимальную задержку распространения сигнала через схему от входов к выходу функции.
8. Определите максимальную частоту изменения входных сигналов, при которой построенная схема сохраняет работоспособность.

1.4 Задания по вариантам

№ Варианта	Вентиль	Логическая схема и логический базис
1	OR	$Y = X1 \vee X2 \vee X3$ Логический базис: И-НЕ
2	NOR	$Y = \neg(X1 \vee X2) \vee X3$ Логический базис: И-НЕ
3	AND	$Y = X1 \wedge X2 \wedge X3$ Логический базис: ИЛИ-НЕ
4	XOR	$Y = (X1 \vee X2) \oplus X3$ Логический базис: ИЛИ-НЕ
5	OR	$Y = \neg(X1 \wedge X2) \vee X3$ Логический базис: И-НЕ
6	NOR	$Y = X1 \wedge \neg(X2 \vee X3)$ Логический базис: И-НЕ
7	AND	$Y = X1 \wedge \neg X2 \wedge \neg X3$ Логический базис: ИЛИ-НЕ
8	XOR	$Y = X1 \oplus X2 \oplus X3$ Логический базис: И-НЕ
9	AND	$Y = \neg X1 \wedge \neg X2 \wedge X3$ Логический базис: ИЛИ-НЕ
10	XOR	$Y = \neg X1 \wedge (X2 \oplus X3)$ Логический базис: И-НЕ

Лабораторная работа №2. «Комбинационные схемы»

2.1 Введение

Комбинационные логические элементы часто группируются в «строительные блоки», используемые для создания сложных систем. Это позволяет абстрагироваться от излишней детализации уровня логических элементов и подчеркнуть функцию «строительного блока». Такие блоки называются базовыми операционными элементами (БОЭ).

При создании реальных цифровых схем проектирование ведется в основном с применением БОЭ, поэтому необходимо хорошо представлять существующие виды БОЭ и их особенности, уметь комбинировать их для достижения разных целей.

2.2 Цели работы

- Получить базовые знания о принципах построения и функционирования цифровых схем комбинационного типа
- Изучить схемотехнику базовых операционных элементов цифровых схем комбинационного типа

2.3 Указания к выполнению работы

Необходимая теория: [1], главы 1-2.

При построении схем заданных БОЭ и комбинационной функции рекомендуется использовать вентили из библиотек Digital или 74НСТ (см. раздел «Основы работы в среде LTspice»). В противном случае для больших схем симуляция может выполняться некорректно.

При измерении задержки распространения сигнала все входы схемы должны управляться единичным инвертором, а все выходы схемы должны быть нагружены единичными инверторами. Это нужно для того, чтобы промоделировать реальную ситуацию в микросхеме, когда входом вентиля управляет другой вентиль (т.е.

источник с ограниченной мощностью, в отличие от идеального источника в симуляторе).

2.4 Порядок выполнения работы

1. С использованием произвольных вентилях реализовать БОЭ согласно варианту задания. Оформить БОЭ как иерархический элемент для библиотеки, задав ему стандартное условное графическое обозначение.
2. Построить таблицу истинности полученной схемы. Для схемы, у которой более пяти входов, допускается таблицу истинности приводить не полностью.
3. Измерить максимальную и минимальную задержку распространения сигнала через схему.
4. Разработать реализацию заданной функции в виде комбинационной схемы с применением различных стандартных БОЭ и, в случае необходимости, дополнительных вентилях. Применяемые БОЭ должны быть построены из вентилях. Все переменные в функции — четырехразрядные, беззнаковые. Используемый в функциях сдвиг является циклическим. В случае переполнения необходимо отбрасывать выходящие за разрядную сетку старшие разряды.
5. Для построенной схемы требуется:
 - (а) проверить работу схемы на нескольких наборах аргументов, включая граничные случаи (переполнения и пр.);
 - (б) измерить задержку распространения сигнала через схему.
6. Подготовить отчет по проделанной работе.

2.5 Задания по вариантам

№	БОЭ	Функция
1	Демультимплексор «1 в 4»	<pre> if (X == 0) Y = 1; else if (X < 5) Y = 2; else Y = X + 3; </pre>
2	Полный четырехразрядный компаратор	<pre> if (X1 < X2) Y = X1; else if (X1 > X2) Y = X2 - 4; else Y = X2; </pre>
3	Схема мажоритарного контроля с пятью входами	<pre> if ((X1 + X2) == 0) Y = X1 - X2; else Y = X2 + 2; </pre>
4	Позиционный шифратор «8 в 3»	<pre> Y = X1 * 2; if (X2 > 4) Y = Y * 4; </pre>
5	Преобразователь BCD-кода в двоичный код (числа от 0 до 99)	<pre> Y = X1 + X2 - X3; if (Y < 2) Y = Y << 2; </pre>
6	Позиционный дешифратор «3 в 8»	<pre> if (X1 < X2) Y = X1 - 2; else Y = X2 - 3; </pre>
7	Трёхразрядный двоичный сумматор с переносом	<pre> if (X1 == 0) Y = X2 << X3; else Y = X2 >> X3; </pre>
8	Шифратор кода Грея для трехразрядного двоичного числа	<pre> if (X1 > 0) Y = X1 - X2; else Y = -X1; </pre>
9	Мультимплексор «4 в 1»	<pre> if (X1 < X2) Y = X2 - X1; else Y = X1 * 2 - X2; </pre>
10	Дешифратор трехразрядного кода Грея	<pre> if (((X1 + X2) & 0x3) == 0) Y = X1 + X2; else Y = (X1 + X2 + 4) & 0x3; </pre>

Лабораторная работа №3. «Триггеры и схемы последовательностного типа»

3.1 Цели работы

- Получить знания о структуре и принципах работы триггеров различных типов
- Познакомиться с основами построения схем последовательностного типа на базе триггеров

3.2 Указания к выполнению работы

Необходимая теория: [1], главы 1-2.

При построении схемы заданного триггера необходимо использовать вентили, построенные на базе КМОП-транзисторов с параметрами из файла 90mn_bulk.txt (см. раздел «Основы работы в среде LTspice»).

При построении заданной последовательностной схемы рекомендуется использовать триггеры и вентили из библиотеки 74HCT или триггеры, построенные на базе вентилях из библиотеки Digital (см. раздел «Основы работы в среде LTspice»). В противном случае для больших схем симуляция может выполняться некорректно.

Последовательностная схема должна быть реализована только с использованием триггеров и произвольных вентилях. Более сложные библиотечные элементы использовать в схеме не допускается. В схеме тестирования можно использовать произвольные библиотечные элементы.

Если не указано иное, при измерении задержки распространения сигнала все входы тестируемой схемы должны управляться единичным инвертором, а все выходы должны быть нагружены единичными инверторами. Это нужно для того, чтобы промоделировать реальную ситуацию в микросхеме, когда входом вентиля управляет другой вентиль (т.е. источник с ограниченной мощностью, в отличие от идеального источника в симуляторе).

3.3 Порядок выполнения работы

1. Построить схему синхронного триггера на вентилях согласно варианту задания.
2. В процессе моделирования определить временные характеристики работы триггера:
 - время предустановки (T_{su});
 - время удержания (T_{hold});
 - время переключения.
3. Определить максимальную частоту синхросигнала, при которой способен работать разработанный триггер.
4. Разработать синхронную последовательностную схему на базе заданного триггера. Триггер, работающий по уровню, рекомендуется заменять на аналогичный триггер, работающий по положительному фронту. Разрядность входной и выходной шин данных последовательностной схемы составляет 4 разряда.
5. Провести моделирование разработанной последовательностной схемы и получить временные диаграммы, отражающие результаты работы схемы.

3.4 Задания по вариантам

№ Варианта	Триггер	Синхронизация	Вентиль	Последовательностная схема
1	T-триггер	По фронту	2И-НЕ	Суммирующий двоичный счетчик с параллельным переносом
2	D-триггер	По фронту	2И-НЕ	Сдвиговый регистр со сдвигом вправо
3	RS-триггер	По фронту	2И-НЕ	Вычитающий двоичный счетчик с параллельным переносом
4	JK-триггер	По фронту	2И-НЕ	Вычитающий двоичный счетчик со сквозным переносом
5	T-триггер	По фронту	2ИЛИ-НЕ	Десятичный суммирующий счетчик
6	D-триггер	По уровню	2ИЛИ-НЕ	Циклический сдвиговый регистр со сдвигом влево
7	RS-триггер	По уровню	2ИЛИ-НЕ	Сдвиговый регистр со сдвигом влево
8	JK-триггер	По уровню	2ИЛИ-НЕ	Вычитающий десятичный счетчик
9	T-триггер	По фронту	2ИЛИ-НЕ	Суммирующий двоичный счетчик со сквозным переносом
10	D-триггер	По фронту	2ИЛИ-НЕ	Сдвиговый регистр с арифметическим сдвигом вправо
11	RS-триггер	По фронту	2ИЛИ-НЕ	Циклический сдвиговый регистр со сдвигом вправо

Лабораторная работа №4. «Цифровые автоматы»

4.1 Введение

Одной из важнейших моделей, применяемых в проектировании последовательных цифровых схем, является конечный автомат (КА) – формальная модель системы с конечным числом состояний. Она позволяет описать поведение некоторого устройства в соответствии с внешними условиями и предысторией. Строго говоря, любая цифровая схема с памятью (в т.ч. триггер) может быть описана как КА.

Запись логики работы разрабатываемого цифрового устройства в виде КА дает много преимуществ. Это позволяет упростить анализ поведения, обеспечивает возможность автоматической оптимизации цифровой схемы, позволяет проводить верификацию и т.д. В связи с этим важно уметь проектировать и анализировать цифровые схемы в терминах КА.

Выделяют два основных типа КА – автоматы Мура и автоматы Мили. Каждый КА имеет набор состояний, начальное состояние, входной и выходной алфавит, логику перехода между состояниями и логику выдачи выходных сигналов.

4.2 Цели работы

- Познакомиться с применением конечных автоматов при разработке цифровых схем
- Получить навыки проектирования и отладки схем с цифровыми автоматами

4.3 Указания к выполнению работы

Необходимая теория: [1], глава 3.

При построении схемы рекомендуется использовать триггеры и вентили из библиотеки 74НСТ (см. раздел «Основы работы в среде LTspice»). В противном случае для больших схем симуляция может выполняться некорректно.

4.4 Порядок выполнения работы

1. Разработать реализацию конечного автомата по заданному в варианте графу переходов с применением указанного способа кодирования состояний и типа триггеров для хранения состояния.
2. Разработать конечный автомат и его реализацию для заданного в варианте цифрового устройства. Необходимо применять указанный способ кодирования состояний и тип триггеров для хранения состояния.
3. Для каждого автомата разработать схему и оформить ее как иерархический элемент.
4. Провести тестирование разработанных автоматов. Для этого разработать схемы тестирования, которые должны обеспечивать проверку поведения автоматов в различных условиях (достаточно нескольких переходов), в том числе сброс в начальное состояние.
5. Для цифрового устройства из второй части задания в отчете необходимо привести граф переходов разработанного конечного автомата и объяснение логики его работы. Также, для каждого разработанного автомата необходимо привести в отчете:
 - тип автомата (Мили или Мура);
 - таблицы кодирования состояний, входных и выходных сигналов, таблицы переходов и выходов;
 - схему;
 - схему тестирования;
 - временные диаграммы, доказывающие корректную работу автомата.

4.5 Задания

Распределение заданий по вариантам см. в конце раздела.

4.5.1 Задания 1-й части

Графы переходов конечных автоматов для реализации представлены на рис. 4.1-4.4. Кодирование состояний указано в задании по варианту. Кодирование всех входов и выходов автомата – прямое унитарное. При комбинациях входных сигналов, не

указанных на графе переходов, переход в новое состояние на текущем такте не производится (иначе говоря, производится переход в то же самое состояние).

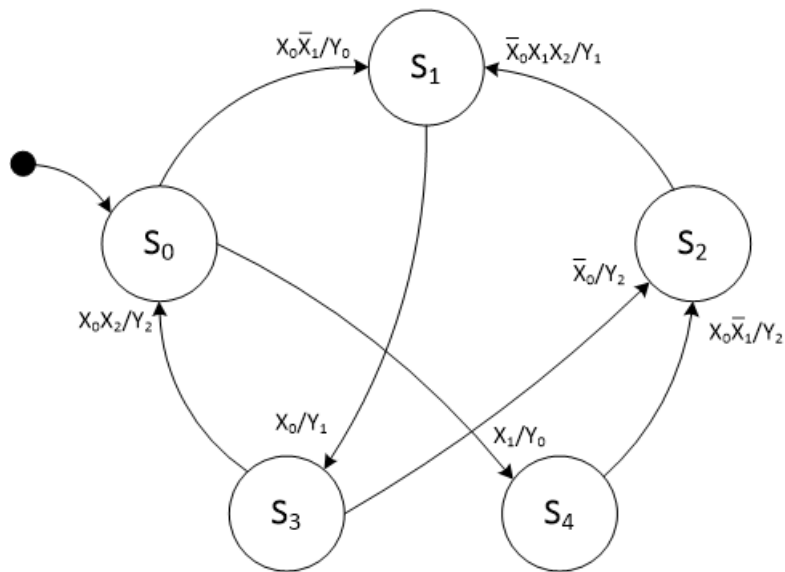


Рисунок 4.1: Граф переходов автомата 1

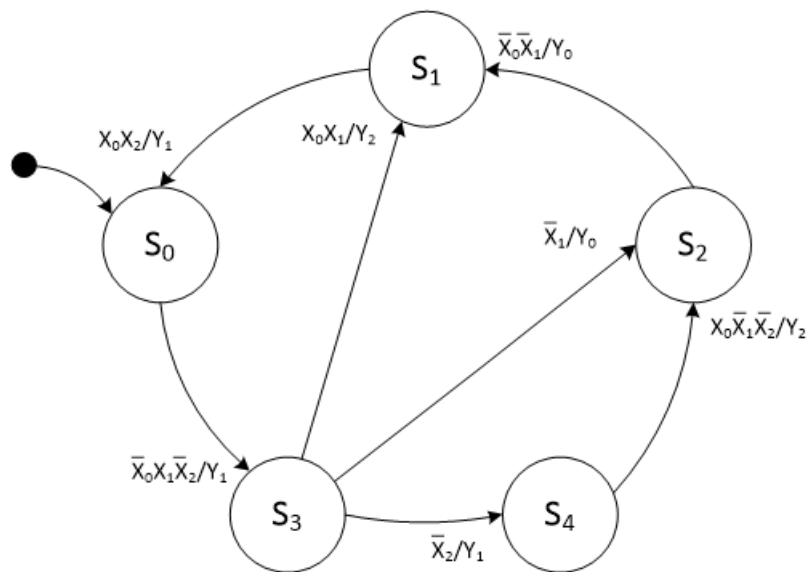


Рисунок 4.2: Граф переходов автомата 2

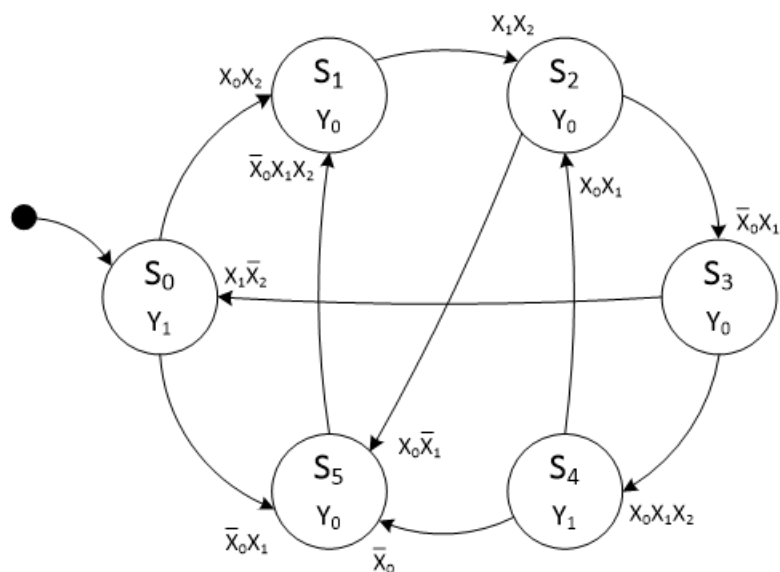


Рисунок 4.3: Граф переходов автомата 3

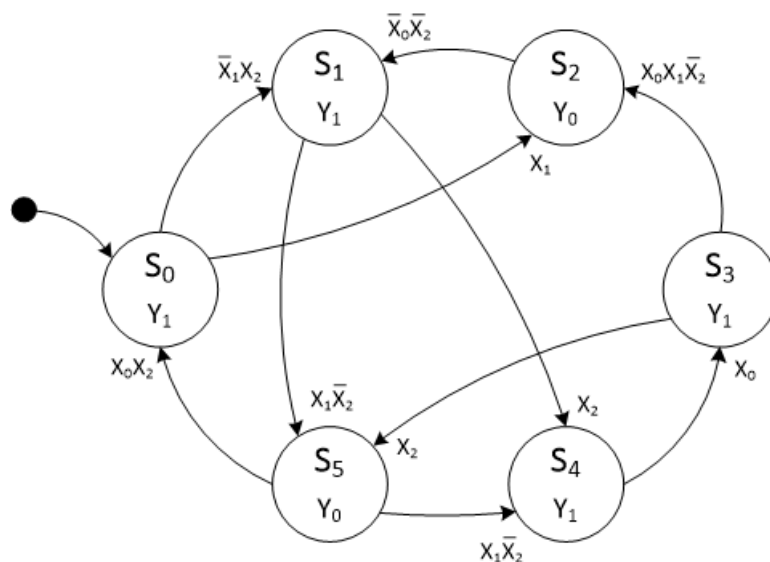


Рисунок 4.4: Граф переходов автомата 4

4.5.2 Задания 2-й части

1. **Распознаватель.** Конечный автомат для распознавания определенных последовательностей во входном потоке битов. Автомат имеет одну входную линию – информационную. Автомат имеет столько выходных линий, сколько последовательностей он может распознавать. На рис. 4.5 показан пример распознавателя с входными и выходными сигналами. Этот автомат сигнализирует, когда во входном потоке битов встречаются последовательности «010» и «101».

Автомат сканирует поток битов, который подается на информационный вход. Если встречается одна из заданных комбинаций, автомат устанавливает соответствующую

щий двоичный выход в «1» на период одного такта синхронизации. Остальное время выходы автомата находятся в состоянии «0». Наложение комбинаций не поддерживается (одна распознанная комбинация не может являться частью другой).

2. Контроллер светофора. Конечный автомат для управления светофором в автоматическом и ручном режиме. Светофор имеет две кнопки: принудительное переключение на красный и удержание зеленого. Обычная схема переключения светофора в автоматическом режиме должна быть следующей: красный, желтый, зеленый, желтый, и снова красный. Стандартное время горения каждым цветом – 1 такт. При нажатии на кнопку принудительного переключения на красный светофор должен переключиться на красный свет из любого состояния, и должен оставаться гореть красным все время, пока удерживается кнопка. При нажатии на кнопку удержания зеленого светофор должен переключаться по обычной схеме, пока не дойдет до зеленого, и должен оставаться гореть зеленым все время, пока удерживается кнопка. Кнопка принудительного переключения на красный имеет приоритет перед кнопкой удержания зеленого. Автомат имеет две входных линии: по одной от каждой из кнопок. «1» на линии кнопки означает, что она нажата. Автомат имеет три выходные линии, каждая из которых предназначена для управления сегментом соответствующего цвета (красный, желтый, зеленый). «1» на выходной линии означает зажигание сегмента. См. рис. 4.6.

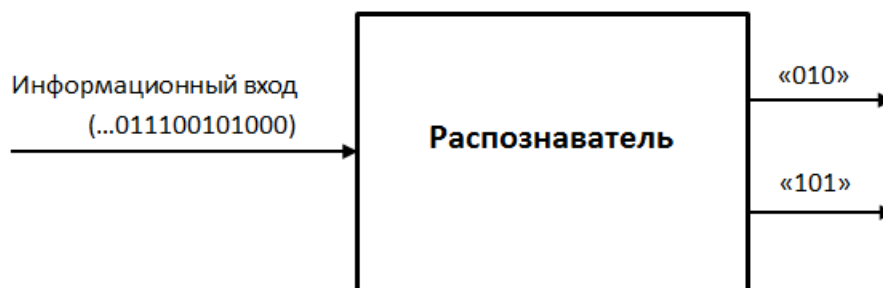


Рисунок 4.5: Входные и выходные сигналы автомата «распознаватель»



Рисунок 4.6: Входные и выходные сигналы автомата «контроллер светофора»

4.5.3 Требования к конечным автоматам

Разработанные конечные автоматы должны удовлетворять следующим требованиям:

1. Автомат должен быть реализован с применением триггеров и произвольных вентилей. Использование более сложных библиотечных БОЭ для реализации автомата не допускается. В схеме тестирования автомата можно использовать произвольные библиотечные элементы.
2. Триггеры в составе автомата должны использоваться только для хранения его состояния.
3. Все триггеры в составе автомата должны работать по положительному фронту.
4. Сигнал синхронизации на все триггеры должен подаваться напрямую, без промежуточных комбинационных элементов.
5. Необходимо предусмотреть сигнал сброса всех триггеров в начальное состояние. Для состояния «0» это сигналы RESET или CLEAR, для состояния «1» – PRESET.
6. В начале работы схемы тестирования автомата необходимо выполнять сброс в начальное состояние.
7. Переход между состояниями автомата должен осуществляться только по фронту сигнала синхронизации.

Т-триггер должен иметь вход «Т», разрешающий переключение в противоположное состояние по фронту синхросигнала. Т-триггер может быть реализован на базе библиотечного триггера с применением дополнительных вентилей.

4.5.4 Задания по вариантам

№ варианта	Автомат	Кодирование состояний	Триггер
1	Автомат 1	Прямое унитарное	JK
	Распознаватель последовательностей: 110, 001	Двоичное	D
2	Автомат 2	Двоичное	D
	Контроллер светофора	Прямое унитарное	JK
3	Автомат 3	Прямое унитарное	RS
	Распознаватель последовательностей: 01, 10, 11	Двоичное	RS
4	Автомат 4	Двоичное	JK
	Контроллер светофора	Двоичное	T
5	Автомат 1	Двоичное	RS
	Распознаватель последовательностей: 110, 001	Прямое унитарное	RS
6	Автомат 2	Прямое унитарное	JK
	Распознаватель последовательностей: 01, 10, 11	Прямое унитарное	D
7	Автомат 3	Двоичное	D
	Распознаватель последовательностей: 110, 001	Двоичное	JK
8	Автомат 4	Прямое унитарное	RS
	Распознаватель последовательностей: 01, 10, 11	Двоичное	T
9	Автомат 1	Прямое унитарное	D
	Контроллер светофора	Двоичное	RS
10	Автомат 2	Двоичное	RS
	Распознаватель последовательностей: 110, 001	Прямое унитарное	T
11	Автомат 3	Прямое унитарное	JK
	Контроллер светофора	Прямое унитарное	D
12	Автомат 4	Двоичное	D
	Распознаватель последовательностей: 01, 10, 11	Прямое унитарное	JK