

Московский государственный университет
имени М. В. Ломоносова

Физический факультет
кафедра фотоники и физики микроволн

О. Ю. Волков

ПРАКТИКУМ ПО РАДИОЭЛЕКТРОНИКЕ
ЦИФРОВЫЕ СХЕМЫ

Методическое пособие для студентов
и преподавателей практикума

Москва – 2013

УДК 378.162.33, 004.942, 53.083.8

ББК 22.3

Практикум по радиоэлектронике: цифровые схемы

Учебно-методическое пособие для студентов III курса физического факультета МГУ и преподавателей практикума по радиоэлектронике. М.: Физический факультет МГУ им. М.В. Ломоносова, 2013. — 26 с.

ISBN 978-5-8279-0104-4

Представлено переработанное описание цикла задач по цифровой электронике с введением ряда новых элементов профессиональной подготовки студентов. Программа состоит из шести задач. Каждая задача сопровождается списком контрольных вопросов и заданий, знание которых необходимо для успешного выполнения упражнений практикума.

Студенты последовательно изучают различные подходы цифровой схемотехники — от использования элементарной логики до проекта на основе программируемых интегральных логических схем (ПЛИС XC9572XL) в среде проектирования ISE. Третья задача совмещается с освоением схемотехнического описания проекта. Четвертая, пятая и шестая задачи помогают освоить язык программирования VHDL, научиться описывать на нем логические операции, простые логические конструкции, а так же изучить работу более сложных элементов — триггеров, счетчиков, регистров и схем на их основе. Помимо синтеза цифровых устройств студент знакомится с методикой симуляции их работы, что позволяет тестировать и отлаживать сложные схемы за минимальное время.

Рецензенты: д.ф.м.н. профессор *А.М. Салецкий*,
д.ф.м.н. доцент *Г.В. Белокопытов*

ВОЛКОВ Олег Юрьевич

Практикум по радиоэлектронике: цифровые схемы

Подписано в печать 18.10.2013

Объем 1,2 п.л. Тираж 70 экз. Заказ № 156

Физический факультет МГУ им. М.В. Ломоносова.
119991, Москва, ГСП-1, Ленинские горы, д. 1, стр. 2

Отпечатано в отделе оперативной печати
физического факультета МГУ им. М.В. Ломоносова

ISBN 978-5-8279-0104-4

© Волков О.Ю., 2013

Задачи

1. Логические элементы	4
1.1. Практическая часть	5
1.2. Контрольные вопросы	6
2. Простейшие логические схемы	7
2.1. Практическая часть	7
2.2. Контрольные вопросы	9
3. Триггеры	10
3.1. Практическая часть	11
3.2. Контрольные вопросы и задания	12
4. Логические схемы и триггеры на VHDL	13
4.1. Практическая часть	13
4.2. Контрольные вопросы	15
5. Счетчики и регистры	16
5.1. Практическая часть	17
5.2. Контрольные вопросы и задания	18
6. Формирователи кода, коррелятор	19
6.1. Практическая часть	19
6.2. Контрольные вопросы и задания	20
Приложения	22
Первый запуск ISE	22
Программирование ПЛИС через схемотехническое описание	23
Программирование ПЛИС на языке VHDL	24
Работа с симулятором ISim	25
Литература	26

Задача 1.

Логические элементы

Логическими элементами называются электронные схемы, реализующие элементарные функции алгебры логики.

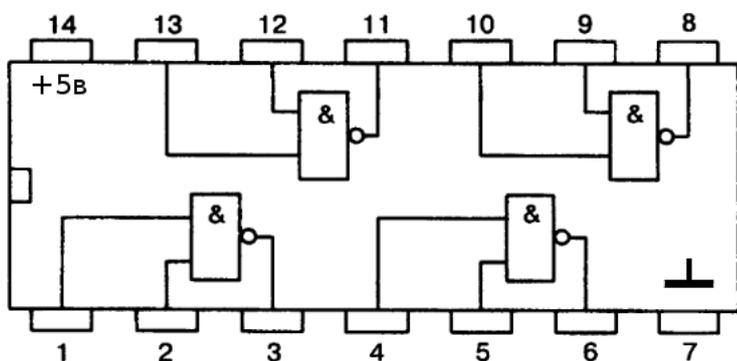


Рис. 1.1. Корпус микросхемы К155ЛА3

Микросхема К155ЛА3 содержит 4 логических элемента 2И-НЕ в корпусе DIP14. Нумерация выводов и условные обозначения элементов даны на рис. 1.1 и 1.2. Напряжение питания +5 В. Уровни логических сигналов

соответствуют стандарту ТТЛ логики. Логическому "0" соответствует напряжение $\leq 0,4$ В, логической "1" — напряжение $\geq 2,4$ В. Для подачи на вход ТТЛ микросхемы логического "0" указанный вход соединяют с общим проводом, логической "1" — вход подключают к сопротивлению 1 кОм, второй конец которого соединен с источником питания (+5 В).

В некоторых случаях допускается оставлять входы ТТЛ микросхем не подключенными. При этом, на входе устанавливается положительный потенциал, соответствующий уровню логической "1". Однако такой способ обеспечения логической "1" приводит к изменению чувствительности других входов, а также импульсных характеристик логического элемента.

Описание работы ТТЛ микросхемы К155ЛА3 изложено в [1] (лабораторная работа 6) и в [2] (стр. 92-100, 111-127).

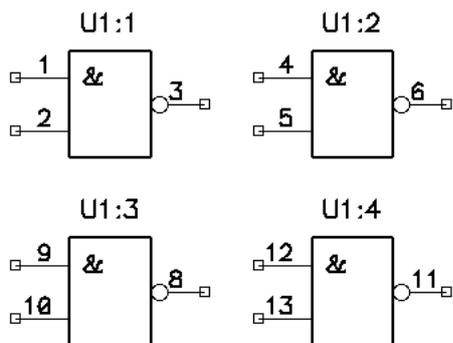


Рис. 1.2. Условные обозначения элементов микросхемы К155ЛА3

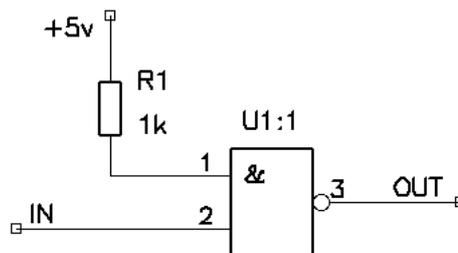


Рис. 1.3. Схема для тестирования свойств элемента микросхемы К155ЛА3

1.1. Практическая часть

1. Изучить работу логического элемента 2И-НЕ (рис. 1.2). Измерить уровень логических "0" и "1" на выходе элемента. Проверить работоспособность всех четырех элементов микросхемы путем подачи на входы выбранного элемента комбинации логических "0" и "1". Измерение напряжения на выходе логического элемента производить при помощи осциллографа с открытым входом.
2. Снять передаточную характеристику элемента 2И-НЕ (рис. 1.3). На вход схемы (точка IN) подать сигнал с генератора треугольного напряжения, размахом 5 В и смещением 2,5 В. К входу и выходу (точки IN и OUT) подключить двухлучевой осциллограф. Зарисовать форму напряжений в этих точках — когда оба сигнала поданы на два луча осциллографа, а также передаточную характеристику — когда развертка осциллографа выключена, точки IN и OUT подключены к усилителям X и Y каналов. На основе полученных графиков определить пороговую величину входного напряжения (при которой происходит переключение) и максимальный коэффициент усиления логического элемента.
3. Определить пороговую величину сопротивления ТТЛ входа (рис. 1.3). Между входом IN схемы и общим проводом подключается потенциометр (сопротивлением 3–5 кОм), к точкам IN и OUT — двухлучевой осциллограф. Вращением потенциометра на входе и выходе логического элемента устанавливается пороговое на-

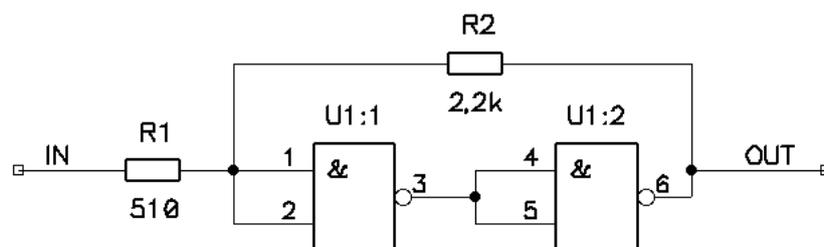


Рис. 1.4. Схема триггера Шмитта на элементах микросхемы К155ЛА3

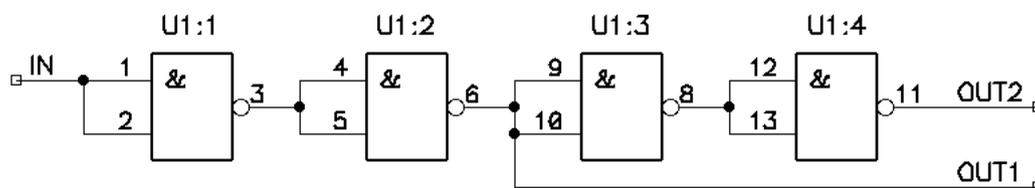


Рис. 1.5. Схема для тестирования времени переключения элемента микросхемы К155ЛА3

пряжение. Потенциометр отсоединяется и с помощью омметра измеряется значение сопротивления.

4. Снять передаточную характеристику триггера Шмитта (рис. 1.4). Получить и зарисовать в тетради форму напряжений на входе и выходе, а также передаточную характеристику схемы (петлю гистерезиса). Сравнить с характеристиками элемента 2И-НЕ. Схема подключается к генератору и осциллографу аналогично упражнению 2.
5. Определить быстродействие элемента 2И-НЕ (рис. 1.5). Подать с генератора импульсов на вход IN схемы меандр ТТЛ уровня, к выходам OUT1 и OUT2 подключить двухлучевой осциллограф. Измерить время задержки переднего фронта сигнала между выходами 2-го и 4-го элементов по уровню порогового напряжения. Определить среднее время переключения одного элемента микросхемы.

1.2. Контрольные вопросы

1. Что понимают под передаточной характеристикой?
2. Понятия "быстродействие" и "время нарастания сигнала".
3. Типовая схема ТТЛ элемента с двумя входами.

Задача 2.

Простейшие логические схемы

Времязадающие цепочки состоят из комбинации емкости и сопротивления. Время релаксации цепочки τ определяется произведением сопротивления и емкости цепочки:

$$\tau \approx RC. \quad (2.1)$$

Использование дифференцирующих или интегрирующих цепочек в цифровых схемах позволяет создавать логические схемы, состояние которых зависит от времени. Примером могут служить формирователи импульсов заданной длительности и автоколебательные системы, рассмотренные в рамках данной задачи.

Задача посвящена знакомству с простыми схемами, получаемыми на логических элементах. Теория рассмотрена в [1] (лабораторные работы 6, 7) и в [2] (стр. 10-91, 221-222).

2.1. Практическая часть

Параметры: частота генерации f_0 или период повторения T_0 (упражнения 1 и 2), скважность s (упражнение 2), длительность импульса τ (упражнение 3), логическое выражение (упражнение 4).

1. Собрать простейший автоколебательный мультивибратор (рис. 2.1). Рассчитать времязадающую цепочку мультивибратора R1C1 по формуле

$$R1C1 \simeq \frac{T_0}{3} \equiv \frac{1}{3f_0} \quad (2.2)$$

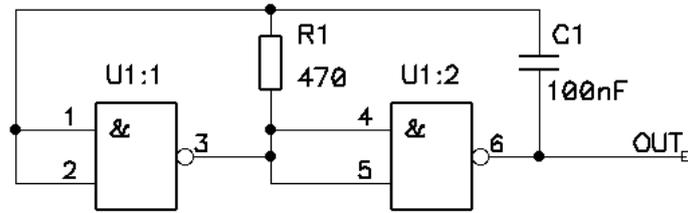


Рис. 2.1. Простейший мультивибратор.

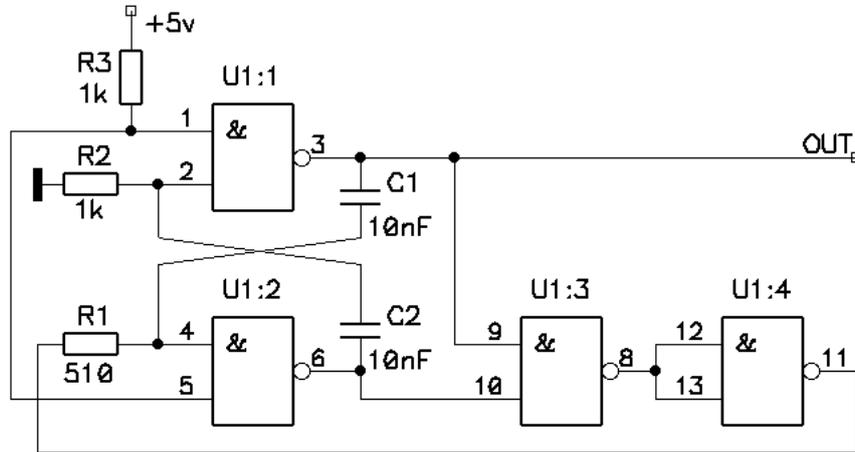


Рис. 2.2. Мультивибратор с заданной скважностью.

для значений $R1$ из ряда 150, 220, 330, 470, 680 и 1000 Ом. Для каждого $R1$ выполнить измерение полученной частоты генерации и скважности. Построить график зависимости частоты и скважности генератора от значения $R1$.

- Собрать автоколебательный мультивибратор с заданной скважностью (рис. 2.2). Длительность импульса τ_1 задается элементами $R1C1$, а временной интервал между ними τ_2 — элементами $R2C2$. Период генерации равен:

$$T_0 = \tau_1 + \tau_2 \simeq R1 C1 + R2 C2. \quad (2.3)$$

Скважность оценивается соотношением:

$$s = \frac{T_0}{\tau_1} \simeq 1 + \frac{R2 C2}{R1 C1}. \quad (2.4)$$

По заданным параметрам рассчитать и собрать схему, измерить частоту, период и скважность генерируемого сигнала, зарисовать форму сигнала.

- Собрать ждущий мультивибратор (рис. 2.3). Мультивибратор запускается фронтом отрицательного импульса (приходом логи-

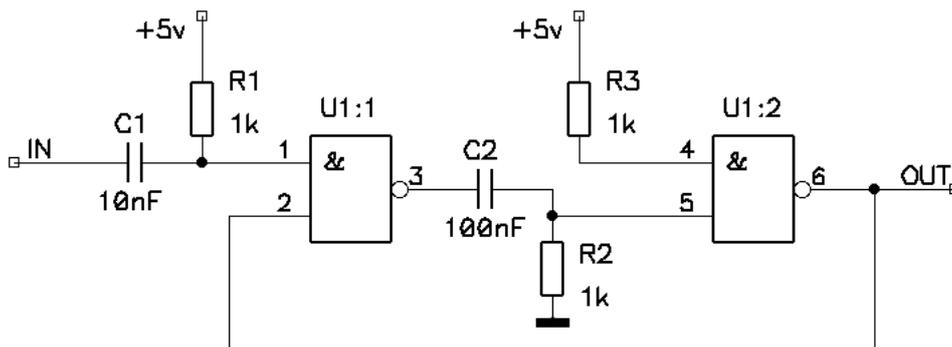


Рис. 2.3. Ждущий мультивибратор.

ческого "0") и формирует на выходе логический "0" заданной длительности. Цепочка $R1C1$ необходима для формирования короткого импульса $\tau_1 \ll \tau$, предназначенного для запуска ждущего мультивибратора. В то же время, τ_1 должно превышать удвоенное время задержки распространения сигнала через один элемент микросхемы $\tau_{\text{задержки}}$. Таким образом:

$$\tau_{\text{задержки}} \ll R1C1 \ll \tau. \quad (2.5)$$

Длительность импульса τ , формируемого ждущим мультивибратором задается цепочкой $R2C2$. На вход ждущего мультивибратора подается сигнал с внешнего генератора импульсов с периодом $T \gg \tau$. Необходимо убедиться, что при изменении скважности входного сигнала длительность формируемого схемой импульса остается неизменной.

4. Сконструировать функции НЕ, ИЛИ, ИСКЛЮЧАЮЩЕЕ_ИЛИ на элементах 2И-НЕ. Для каждой схемы привести таблицу истинности. Упражнение выполняется в тетради.

2.2. Контрольные вопросы

1. Булева алгебра. Коммутативность. Идемпотентность. Ассоциативность. Дистрибутивность. Комплементарность. Инволютивность отрицания. Законы де Моргана. Законы поглощения.
2. Двоичный код. Код Грея. Таблица истинности логической схемы.
3. Методы упрощения логической функции. Карты Карно.

Задача 3.

Триггеры

Триггерами называются логические схемы, имеющие несколько стабильных состояний, зависящих не только от сигналов на входах, но и от предыдущего состояния схемы, то есть обладающие эффектами памяти. В цифровой технике применяются триггеры, имеющие 2 стабильных состояния. Для ввода сигналов в триггерах имеются один или несколько информационных входов. Различают асинхронные и синхронные триггеры, синхронные — управляемые уровнем (статическое управление) или фронтом напряжения (динамическое управление).

Задача посвящена изучению принципов работы простейших триггеров и их построению на основе логических элементов. Работа триггеров рассмотрена в [1] (лабораторная работа 7) и в [2] (стр. 151-242). Упражнения выполняются на макетной плате с установленной программируемой логической интегральной схемой (ПЛИС) XC9572XL. На макетной плате имеются четыре управляющих кнопки и четыре светодиода, позволяющие подавать различные комбинации входных сигналов, а так же наблюдать состояния выходов собираемой схемы.

Перед началом выполнения упражнений необходимо освоить работу со средой ISE, программатором и макетной платой, на примере готового проекта, поставляемого разработчиками макетной платы (стр. 22). Описание работы с ISE подробно рассмотрено в [3] (глава 3).

Для каждого упражнения необходимо создать собственный проект. Порядок создания проекта описан в приложении на стр. 23.

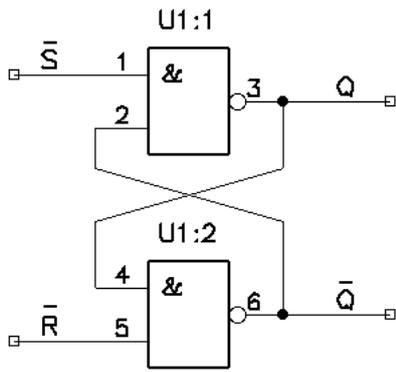


Рис. 3.1. Асинхронный RS-триггер

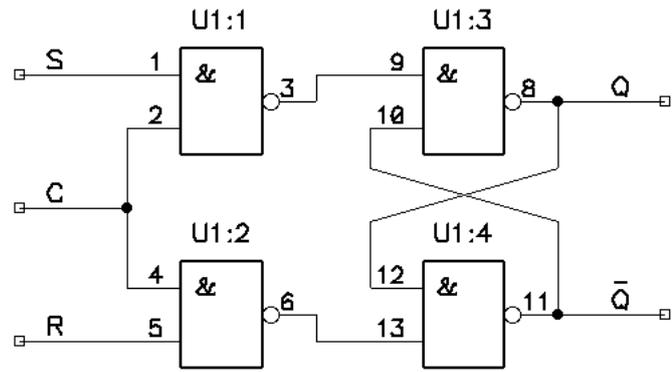


Рис. 3.2. Синхронный RS-триггер

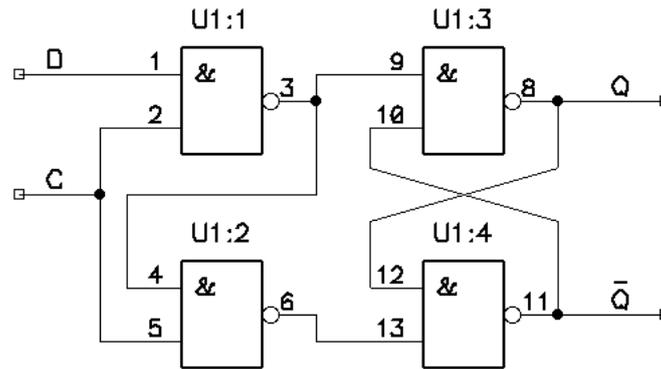


Рис. 3.3. D-триггер

3.1. Практическая часть

1. Собрать схему асинхронного RS-триггера (рис. 3.1). Входы \bar{R} и \bar{S} подключить к кнопкам 1 и 2. Выходы — к светодиодам 1 и 2. Записать таблицу истинности.
2. Собрать схему синхронного RS-триггера (рис. 3.2). Входы R, S и C подключить к кнопкам 1–3. Выходы — к светодиодам 1 и 2. Найти состояния хранения. Записать таблицу истинности.
3. Собрать синхронный D-триггер, управляемый уровнем (рис. 3.3). Входы D и C подключить к кнопкам 1 и 2, выходы — к светодиодам 1 и 2. Записать таблицу истинности.
4. Собрать D-триггер, управляемый фронтом. Вход D подключить к кнопке 1, вход C — к кнопке 2. Выходы D-триггера — к светодиодам 1 и 2. Записать таблицу истинности.

3.2. Контрольные вопросы и задания

1. Каким образом в сложной логической схеме можно выявить наличие триггеров?
2. Перечислите известные вам виды триггеров.
3. Что такое запрещенное состояние, и в каких триггерах оно возможно?
4. Классифицируйте триггеры по способу тактирования.
5. Каково отличие триггера, управляемого по фронту, от управляемого по уровню сигнала?
6. В чем особенность построения таблицы истинности для триггеров?
7. Составить таблицу истинности для триггера, имеющего прямые или инверсные входы R и S , выходы Q и \bar{Q} .
8. Составить таблицу истинности для триггера, имеющего прямые или инверсные входы R , S и C (тактирование уровнем).
9. Составить таблицу истинности для триггера с прямыми или инверсными входами D и C (тактирование уровнем).
10. Составить таблицу истинности для триггера с прямыми или инверсными входами D и C (тактирование фронтом).
11. Составить таблицу истинности JK триггера, тактируемого положительным или отрицательным фронтом, имеющего входы J , K , \bar{R} и \bar{S} .
12. Составить таблицу истинности JK триггера, тактируемого положительным или отрицательным фронтом, имеющего входы J_1 , K_1 , \bar{J}_2 , \bar{K}_2 .

Задача 4.

Логические схемы и триггеры на VHDL

Задача частично повторяет упражнения, связанные с работой простых логических функций и триггеров, рассмотренных в [1] (лабораторные работы 6, 7) и в [2] (стр. 10-242). Упражнения выполняются на макетной плате с ПЛИС XC9572XL. Особенностью выполнения упражнений является подготовка проектов для ПЛИС на языке VHDL, являющимся универсальным языком программирования ПЛИС различных производителей. Информация по синтаксису языка VHDL приведена в [4,5], примеры — в [3] (глава 4).

Для каждого упражнения необходимо создать собственный проект. Порядок создания проекта описан в приложении на стр. 24.

4.1. Практическая часть

1. Получить функции 2И-НЕ, 2ИЛИ, 2ИСКЛЮЧАЮЩЕЕ_ИЛИ.

На языке VHDL функции описываются следующим образом:

```
Y1 <= not ( X1 and X2 );    -- 2И НЕ
Y2 <= X1 or X2;            -- 2ИЛИ
Y3 <= X1 xor X2;           -- 2ИСКЛЮЧАЮЩЕЕ ИЛИ
```

Входы всех трех схем подключить к кнопкам 1 и 2. Выходы — к светодиодам 1–3 (каждую схему к своему светодиоду). Проконтролировать таблицу истинности, по выполненной в упражнении 4 задачи 2.

2. Собрать одновременно работающие асинхронный и синхронный RS-триггеры (рис. 3.1, 3.2). На языке VHDL RS-триггер описывается следующим образом:

```

-- после architecture Behavioral of ....., но перед begin
SIGNAL Q   : STD_LOGIC := '0';
SIGNAL neQ : STD_LOGIC := '1';

-- после architecture .... и begin, но перед end Behavioral;
Q   <= not ( neS and neQ );
neQ <= not ( neR and Q );
led1 <= Q;  -- поскольку сами сигналы Q и neQ нельзя непосредственно подклю-
led2 <= neQ; -- чать к выходам ПЛИС, они используются в выражениях (читаются)

```

Описание синхронного RS триггера предлагается составить самостоятельно. Входы R и \bar{R} схем подключить к кнопке 1, S и \bar{S} — 2, C — 3. Выходы Q и \bar{Q} обоих триггеров — к светодиодам 1–4.

3. Собрать в одном проекте ждущий мультивибратор, два D-триггера, T-триггер. На VHDL ждущий мультивибратор описывается следующим образом:

```

-- после architecture Behavioral of ....., но перед begin
SIGNAL n : INTEGER := 0;
SIGNAL C : STD_LOGIC := '0'; -- for use in D and T triggers

-- после architecture .... и begin, но перед end Behavioral;
process (clk)
begin
  if(clk'event and clk = '1') then
    if(n > 0) then
      n <= n-1;
    elsif(bot1 = '0') then
      C <= '1';
      n <= 100000000; -- for clock 100 MHz - period 1 sec
    else
      C <= '0';
    end if;
  end if;
end process;
led1 <= C;

```

Ждущий мультивибратор, подключить к кнопке 1 и светодиоду 1. D-триггер, управляемый уровнем: входы C и D подключить к выходу ждущего мультивибратора и кнопке 2, выход к светодиоду 2. D-триггер, управляемый фронтом: входы C и D подключить к выходу ждущего мультивибратора и кнопке 3, выход к светодиоду 3. T-триггер: вход подключить к выходу ждущего мультивибратора, выход — к светодиоду 4.

4. Собрать JK-триггер. Описать делитель частоты тактового генератора, чтобы получить период повторения 3 с. Выход подключить к светодиоду 1. Вход С JK-триггера подключить к делителю частоты, входы \bar{R} , \bar{S} , J и K — к кнопкам 1–4. Выходы — к светодиодам 3 и 4. Составить таблицу истинности JK триггера.

4.2. Контрольные вопросы

1. Чем отличается тип данных `std_logic` от типа `bit`? Как вы понимаете отличие состояний "сильная 1" от "слабая 1" на практике?
2. Чем отличается тип данных `std_logic_vector` от типа `integer`? Что такое подтипы `natural`, `positive`?
3. Чем отличаются константы, переменные и сигналы? Где и как их можно использовать?
4. Чем отличаются "=", "<=" и ":="? Приведите пример использования.
5. Чем отличается операция "+ and" и "&"? Приведите пример для двух чисел "1010" и "0011".
6. Что означает выражение

```
clk'event and clk = '1'
```

и отдельные его части. Где и для чего оно применяется.

7. Что такое структурное описание схемы? Чем отличается от поведенческого описания?
8. Что такое процесс? Что такое лист чувствительности?
9. Что такое атрибуты? Какие атрибуты стоит использовать в проектах, а какие только при симуляции?

Задача 5.

Счетчики и регистры

Счетчики — это устройство для регистрации количества импульсов, подаваемых на его вход. Коэффициент пересчета N определяет возможное число состояний схемы. Простейшие счетчики можно построить путем последовательного соединения счетных триггеров — такие счетчики называются последовательным и обладают сравнительно низким быстродействием. Современные ПЛИС позволяют создавать параллельные счетчики с большим числом разрядов, время переключения которых соответствует максимально возможному быстродействию микросхемы. Для этого используются линии глобального тактирования кристалла.

Регистры — группа определенным образом соединенных триггеров, предназначенных для временного запоминания многоразрядных чисел. Различают последовательные, параллельные и параллельно-последовательные регистры.

Задача посвящена изучению счетчиков, регистров сдвига и схем на их основе. Упражнения выполняются на языке VHDL для макетной платы с ПЛИС XC9572XL. Реализация схем, входящих в задачу на дискретной логике подробно рассмотрена в [1] (лабораторная работа 8) и в [2] (стр. 263-313, 329-336). Особенностью выполнения упражнений задачи является обращение к линейке светодиодов как к 4-разрядному вектору. При этом, в ucf файле необходимо ставить соответствие элемент вектора LED(3 downto 0) и ножку порта, например:

```
NET "LED(0)" LOC = "P5" ;
```

Для каждого упражнения необходимо создать собственный проект. Дополнительно, работа каждого устройства проверяется в симуляторе ISim (схема устранения дребезга контактов при этом исключается). Порядок работы с симулятором описан в приложении на стр. 25.

5.1. Практическая часть

Параметры: N — число пересчета реверсивного счетчика (упражнение 2).

1. Собрать счетчик с естественным порядком счета. На VHDL схема, считающая число нажатий (с устранением "дребезга контактов") кнопки 1 и отображающая результат в двоичном коде на светодиодах 1–4 описывается следующим образом:

```
-- после описания IEEE библиотеки
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- после architecture Behavioral of ....., но перед begin
SIGNAL n : STD_LOGIC_VECTOR(3 downto 0) := "0000";
SIGNAL p : INTEGER := 0;

-- после architecture .... и begin, но перед end Behavioral;
process (Clk)
begin
  if(Clk'event and Clk = '1') then
    if(bot1 = '0') then
      if(p = 0) then
        n <= n + 1;
      end if;
      p <= 10000;
    elsif(p > 0) then
      p <= p - 1;
    end if;
    led <= n;
  end if;
end process;
```

Записать таблицу всех возможных состояний счетчика, начиная с "0000".

2. Собрать реверсивный счетчик на N . Описать на VHDL схему, увеличивающую значение на 1 при нажатии на кнопку 1 и уменьшающую на 1 при нажатии на кнопку 2 (с устранением "дребезга контактов"). Сброс счетчика в положение "0000" должен осуществляться по нажатию на кнопку 3. Результат счета должен отображаться в двоичном коде на светодиодах 1–4.
3. Собрать 4-х разрядный регистр сдвига. Описать на VHDL схему, представляющую собой регистр сдвига, у которого кнопка 1 является тактовым сигналом (с устранением "дребезга контактов"), кнопка 2 — входом переноса, кнопка 3 должна подключать вход переноса вместо кнопки 2 к выходу последнего разряда регистра, а кнопка 4 — к инверсному выходу старшего разряда. Результат должен отображаться на светодиодах 1–4.

5.2. Контрольные вопросы и задания

1. Чем отличаются синхронные и асинхронные счетчики?
2. Что понимается под счетчиком по модулю n ? Что такое реверсивный счетчик?
3. Приведите схему счетчика, построенного на D-триггерах.
4. Что такое сдвиговый и кольцевой регистр?
5. Приведите схему сдвигового и кольцевого регистра, построенного на D-триггерах.
6. Опишите сдвиговый и кольцевой регистр на VHDL.

Задача 6.

Формирователи кода, коррелятор

В задаче изучаются формирователи детерминированных и случайных бинарных последовательностей, а так же схемы анализа битового потока данных — коррелятора. Упражнения выполняются на языке VHDL для макетной платы с ПЛИС XC9572XL. Реализация схем, входящих в задачу на дискретной логике рассмотрена в [1] (лабораторная работа 9) и в [2] (стр. 393-413).

Для каждого упражнения необходимо создать собственный проект. Работа каждого упражнения проверяется в симуляторе ISim. Порядок работы с симулятором описан в приложении на стр. 25.

6.1. Практическая часть

Параметры: N — количество элементов кода Баркера (упражнение 1).

1. Описать на VHDL схему, формирующую один из возможных кодов Баркера для заданного количества элементов кода N .

$N = 3$	1 1 -1
$N = 4$	1 1 -1 1
$N = 5$	1 1 1 -1 1
$N = 7$	1 1 1 -1 -1 1 -1
$N = 11$	1 1 1 -1 -1 -1 1 -1 -1 1 -1
$N = 13$	1 1 1 1 1 -1 -1 1 1 -1 1 -1 1

Тактирование схемы осуществляется от генератора, расположенного на макетной плате. Выход подключить к светодиоду 1. На

светодиод 2 вывести сигнал, соответствующий первому символу кода. Результат продемонстрировать в симуляторе ISim.

2. Описать на VHDL схему, формирующую M-последовательность. Тактирование схемы осуществляется от генератора, расположенного на макетной плате. Нажатие и удержание кнопки 1 должно приостанавливать счет. Выход 4-х младших разрядов подключить к светодиодам. Результат формирования последовательности продемонстрировать в симуляторе ISim. Найти период повторения последовательности.
3. Описать на VHDL схему коррелятора, считающую количество совпадающих бит в 13-ти разрядном сдвиговом регистре с кодом Баркера. Результат в виде двоичного кода вывести на светодиодную линейку. Нажатие кнопок 1-4 должно приводить к сдвигу информации в регистре, с заполнением младшего разряда логическим "0", "1", значением старшего разряда (циклический сдвиг) и псевдослучайным значением соответственно. Построить в тетради функцию автокорреляции кода Баркера и функцию корреляции кода Баркера с псевдослучайной последовательностью.

6.2. Контрольные вопросы и задания

1. Что такое код Баркера? Основные свойства радиосигнала, промодулированного кодом Баркера.
2. Что такое M-последовательность? Ее свойства. Привести алгоритм синтеза M-последовательности с периодом повторения 255.
3. Опишите на языке VHDL модулятор кодом Баркера на 11 передаваемой информации. В качестве источника информации используйте кнопку 1, выход — светодиод 1. При нажатии кнопки должен передаваться инверсный код Баркера, а при отпущенной — прямой. (Состояние кнопки оценивать перед началом передачи)

очередной кодовой последовательности, а не при передаче каждого символа кода.)

4. Опишите на языке VHDL устройство, представляющее собой 4-разрядный сдвиговый регистр, тактируемый частотой 1 Гц (при частоте тактового генератора 100 мГц). В качестве источника информации — состояние кнопки 1. Выходы — на светодиоды 1-4.
5. Опишите на языке VHDL 4-разрядный счетчик, считающий с частотой 1 Гц в прямом направлении если управляющая кнопка отпущена и в обратном — если нажата.
6. Описать на VHDL формирователь 4-разрядных случайных чисел с периодом повторения не менее 1000. Каждое следующее число должно быть случайным полностью, а не полученное путем сдвига на 1 бит с последующим добавлением 1 случайного бита.

Приложения

Первый запуск ISE

1. Запустить программу ISE. Закрывать окно Tip of the Day, сняв галочку Show Tips at Startup. Закрывать окно Xilinx License Error, нажав ОК. Подождать запуска менеджера лицензий.
2. В окне Xilinx License Configuration Manager выбрать закладку Manage Xilinx License, нажать кнопку Copy License и подключить файл лицензий `"/opt/Xilinx/Xilinx.lic"`, закрыть все окна, относящиеся к менеджеру лицензий.
3. Открыть тестовый проект (File – Open Project). Имя файла `"/home/student/Desktop/ISE_test/XC9572XLVQ44.xise"`.
4. Запустить программу iMPACT, проигнорировав предупреждение и отказавшись от опроса.
5. Двойным щелчком мыши вызвать режим сканирования оборудования Boundary Scan. Выбрать инициализацию канала (File – Initialize Chain). В окне Auto Assign Configuration Files Query Dialog поставить галочку и нажать Yes. В окне Assign New Configuration Files выбрать соответствующий файл проекта `"/home/student/Desktop/ISE_test/XC.jed"`. Закрывать окно Device Programming Properties нажав ОК.
6. Выбрать режим очистки Erase. Убедиться что 4 светодиода горят в пол накала.
7. Выбрать режим программирования Programm. 3 светодиода должны погаснуть, один гореть в четверть накала. При нажатии на кнопки состояние светодиодов должно изменяться.

Программирование ПЛИС через схемотехническое описание

1. Запустить ISE и выбрать режим создания нового проекта New project. В окне New Project Wizard задать имя проекта (Name) и выбрать тип проекта (Top-level sources type) — Schematic. Нажать Next. Выбрать параметры микросхемы: Family — XC9500XL CPLDs, Device — XC9572XL, Package — VQ44, Speed — "-10".
2. Создать новый лист схемы (Project – New Source, тип Schematic). Разместить на нем необходимые библиотечные элементы (Add – Symbol). Выполнить необходимые соединения элементов (Add – Wire). Входы и выходы вывести на левый и правый край схемы соответственно, подсоединив к ним коннекторы (Add I/O Marker). Переименовать цепи, идущие к коннекторам в соответствии с назначением сигналов (двойной щелчок по коннектору, Category: Nets).
3. Составить ucf файл (Project – New Source, тип Implementation Constraints File), ставящий в соответствие названия цепей схемы и номеров выводов ПЛИС. Пример содержимого:

```
NET "BOT1" LOC = "P12" ;
NET "BOT2" LOC = "P13" ;
NET "BOT3" LOC = "P14" ;
NET "BOT4" LOC = "P16" ;
NET "GCK1" LOC = "P43" ;
NET "LED1" LOC = "P5" ;
NET "LED2" LOC = "P6" ;
NET "LED3" LOC = "P7" ;
NET "LED4" LOC = "P8" ;
```
4. Выполнить компиляцию схемы (Process – Implement Top Module) и исправить ошибки.
5. Осуществить программирование ПЛИС, для этого, запустить программу iMPACT (Tools – iMPACT), выполнить сканирование (Boundary Scan), инициализацию канала (File – Initialize Chain), в окне Assign New Configuration Files выбрать соответствующий файл проекта, очистить и запрограммировать ПЛИС (Erase, Programm).

Программирование ПЛИС на языке VHDL

1. Запустить ISE и выбрать режим создания нового проекта New project. В окне New Project Wizard задать имя проекта (Name) и выбрать тип проекта (Top-level sources type) — HDL. Нажать Next. Выбрать параметры микросхемы: Family — XC9500XL CPLDs, Device — XC9572XL, Package — VQ44, Speed — "-10". Предпочитаемый язык (Preferred Language) — VHDL.
2. Создать новый модуль описания (Project – New Source, тип VHDL Module). Нажать Next. Указать имена цепей ввода-вывода (Port Name) и направление Direction. Цепи, подключаемые к кнопкам — in, к светодиодам — out. В результате получится шаблон на языке VHDL, в котором необходимо вписать необходимый функционал.
3. Составить ucf файл (Project – New Source, тип Implementation Constraints File), ставящий в соответствие названия цепей схемы и номеров выводов ПЛИС.
4. Выполнить компиляцию схемы (Process – Implement Top Module) и исправить ошибки.
5. Для получения статистической информации о работе синтезатора вызвать Project – Design Summary/Reports.
6. Для визуального контроля работы синтезатора существует два варианта представления: как реализован проект на уровне описания цепей сигналов и сохранения их состояний в регистрах (Tools – Schematic Viewer – RTL); как транслятор ISE распорядился ресурсами микросхемы (Tools – Schematic Viewer – Technology).
7. Осуществить программирование ПЛИС, для этого, запустить программу iMPACT (Tools – iMPACT), выполнить сканирование (Boundary Scan), инициализацию канала (File – Initialize Chain), в окне Assign New Configuration Files выбрать соответствующий файл проекта, очистить и запрограммировать ПЛИС (Erase, Programm).

Работа с симулятором ISim

1. Запустить ISE, открыть проект или создать новый.
2. Создать новый модуль описания (Project – New Source, тип VHDL Test Bench). В результате получится шаблон работы с симулятором на языке VHDL, в котором необходимо вписать необходимый функционал.
3. В левой части окна войти в панель Design, переключить вид на симуляцию View: Simulation и активизировать файл проекта, который предполагается симулировать. При этом, в нижней части панели Design появится пункт ISim Simulator и в нем два подпункта: Behavioral Check Syntax — для проверки синтаксиса и Simulate Behavioral Model — для запуска ISim. Необходимо выполнить их последовательно.
4. В окне ISim выполнить View – Zoom – To Full View для получения обзорного вида на процесс симуляции.

Литература

1. И.Т.Трофименко, Е.В.Лебедева, Н.С.Седлецкая. Практикум по радиоэлектронике. Учеб. пособие. — М. Изд-во МГУ, 1997.
2. Бойт К. Цифровая электроника. 2007 г.
3. И.Е. Тарасов. Разработка цифровых устройств на основе ПЛИС Xilinx с применением языка VHDL Горячая Линия - Телеком, 2005, ISBN: 5-93517-242-9.
4. ГОСТ Р 50754-95 Язык описания аппаратуры цифровых систем VHDL. Описание языка.
5. Сайт по VHDL: <http://allhdl.ru/vhdl.php>

Пособие составлено для студентов физического факультета МГУ, изучающих основы работы цифровых схем в практикуме по радиоэлектронике. Программа состоит из 6 задач. Выполнение каждой задачи начинается с допуска. Список вопросов приводится в конце описания каждой задачи. Для ответа на контрольные вопросы необходимо чтение дополнительной литературы.

Первая и вторая задачи выполняются на макетной плате и предназначены для знакомства с дискретными элементами, на примере микросхемы к155ла3. В первой — логика работы элемента 2И-НЕ, передаточные характеристики, время задержки распространения сигнала, схема триггера Шмитта. Вторая задача посвящена схемам, содержащим времязадающие цепочки (различные мультивибраторы).

Дальнейшая работа студента переносится на макетную плату с установленной на ней программируемой логической интегральной схемой (ПЛИС) XC9572XL. Третья задача знакомит студента с языком схемотехнического описания ПЛИС на примере построения схем простейших триггеров. Четвертая задача показывает преимущество использования языка VHDL, как более оптимального способа создания современных электронных схем.

Пятая и шестая задачи дают студенту возможность более подробно изучить язык VHDL, освоить симулятор ISim, использовать полученные навыки для выполнения более сложных задач, таких как счетчики, регистры, формирователи кода.

Автор выражает благодарность техническому директору ООО "ЛОИС" А.Л. Ампелонову, сотрудникам физического факультета МГУ В.М. Гурьеву, В.Ф. Марченко, И.В. Румянцеву, Н.А. Сухаревой, А.П. Сухорукову, С.Н. Цысарю и студентам 3-го курса, принявшим участие в тестировании новой методики проведения занятий в практикуме по радиоэлектронике.

ISBN 978-5-8279-0104-4



9 785827 901044