

Московский государственный университет
имени М. В. Ломоносова

Физический факультет

О. Ю. Волков

ПРАКТИКУМ ПО РАДИОЭЛЕКТРОНИКЕ
ЦИФРОВЫЕ СХЕМЫ

Методическое пособие для студентов
третьего курса, осенний семестр

Москва – 2012

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, 2012. — 22 с.

Представлено переработанное описание цикла задач по цифровой электронике с введением ряда новых элементов подготовки студентов к выполнению задач, технологии выполнения и в формы отчета по результатам. Пособие ориентировано на этап тестирования в осеннем семестре 2012/2013 учебного года методов создания электронных схем на программируемых логических интегральных схемах. Материал пособия в равной мере адресован студентам и преподавателям, принимающим участие в проекте, и служит основой для последующего тематического развития цифрового раздела практикума по радиоэлектронике.

ВОЛКОВ Олег Юрьевич

Практикум по радиоэлектронике: цифровые схемы

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

Аннотация

Пособие составлено для студентов физического факультета МГУ, изучающих основы работы цифровых схем в практикуме по радиоэлектронике. Программа состоит из 5 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 задача выполняется на макетной плате с микросхемой К155ЛА3. Задачи предназначены для знакомства с логическими элементами и конструированием простейших схем на их основе, освоении элементов булевой алгебры. Третья, четвертая и пятая задачи выполняются на макетной плате с микросхемой XC9572XL, являющейся программируемой логической интегральной схемой.

Студенты последовательно изучают различные подходы цифровой схемотехники — от использования элементарной логики до проекта для ПЛИС. Третья задача совмещается с освоением схемотехнического описания проекта. Четвертая и пятая задачи помогают освоить язык программирования VHDL, научиться записывать на нем логические операции, простые логические конструкции, а так же изучить работу более сложных элементов — триггеров, счетчиков, регистров и схем на их основе. В процессе изучения языка VHDL вводится понятие симуляции работы ПЛИС, что позволяет тестировать и отлаживать сложные схемы за минимальное время.

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	10
3.2	Контрольные вопросы.	11
4	Логические схемы и триггеры на VHDL.	12
4.1	Практическая часть	12
4.2	Контрольные вопросы.	14
5	Счетчики, регистры, формирователи кодов	15
5.1	Практическая часть	15
5.2	Контрольные вопросы.	17
	Приложения	18
	Первый запуск ISE	18
	Программирование ПЛИС через схемотехническое описание	19
	Программирование ПЛИС на языке VHDL	20
	Работа с симулятором ISim	21
	Литература	22

Задача 1

Логические элементы.

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

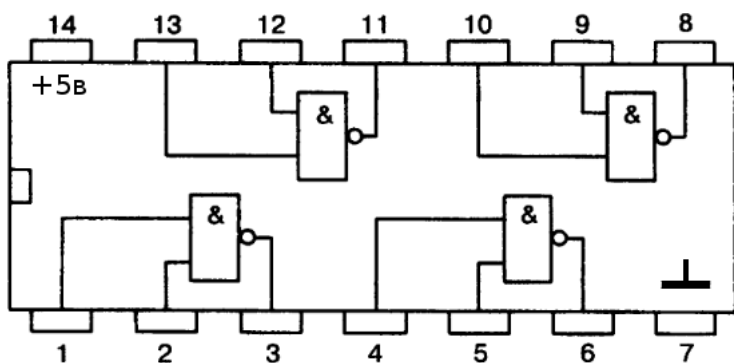


Рис. 1.1: Корпус микросхемы К155ЛА3

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] (задача 6) и в [2] (стр. 92-100, 111-127).

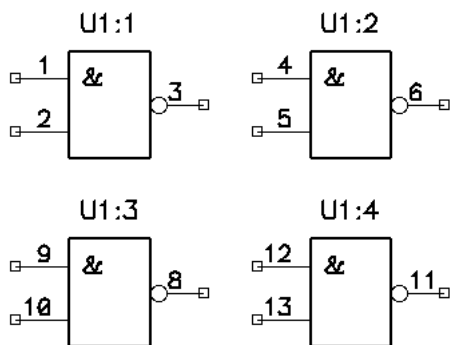


Рис. 1.2: Условные обозначения элементов микросхемы К155ЛА3

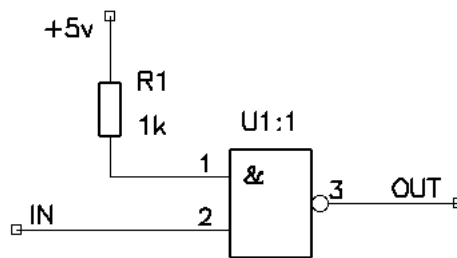


Рис. 1.3: Схема для тестирования свойств элемента микросхемы К155ЛА3

1.1 Практическая часть

1. Изучение работы логического элемента 2И-НЕ (рис. 1.2). Измерение уровня логических "0" и "1" на выходе элемента. Проверка работоспособности всех четырех элементов микросхемы путем подачи на входы выбранного элемента комбинации логических "0" и "1". Измерение напряжения на выходе логического элемента производится при помощи осциллографа с открытым входом.
2. Снятие передаточной характеристики (рис. 1.3). На вход схемы (точка IN) подается сигнал с генератора треугольного напряжения, размахом 5 В и смещением 2,5 В. К входу и выходу (точки IN и OUT) подключается двухлучевой осциллограф. Необходимо 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. Определение пороговой величины сопротивления на ТТЛ входе (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

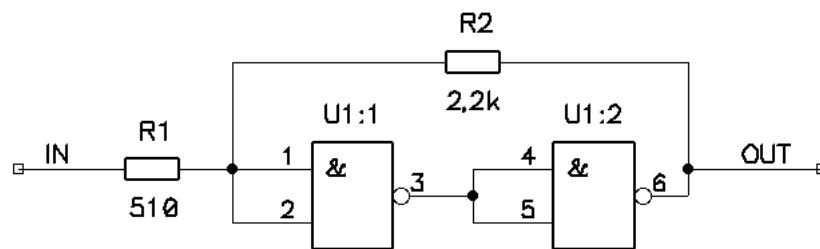


Рис. 1.4: Схема триггера Шмитта на элементах микросхемы К155ЛА3

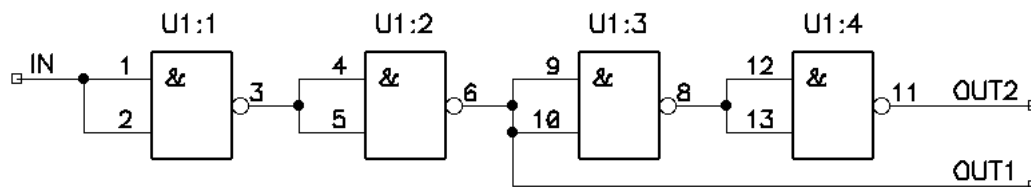


Рис. 1.5: Схема для тестирования времени переключения элемента микросхемы К155ЛА3

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

4. Триггер Шмитта (рис. 1.4). Схема подключается к генератору и осциллографу аналогично упражнению 2. Получить и зарисовать в тетради форму напряжений на входе и выходе, а также передаточную характеристику схемы (петлю гистерезиса). Сравнить с характеристиками элемента 2И-НЕ.
5. Определение времени задержки переключения 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.

Задача посвящена знакомству с простыми схемами, получаемыми на логических элементах. Теория рассмотрена в [1] (задача 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. Простейший автоколебательный мультивибратор (рис. 2.1). Рассчитать времязадающую цепочку мультивибратора R1C1 по формуле

$$R1 C1 \simeq \frac{T_0}{3} \equiv \frac{1}{3f_0} \quad (2.1)$$

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.

2. 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:

$$T_0 = \tau_1 + \tau_2 \simeq R1 C1 + R2 C2. \quad (2.2)$$

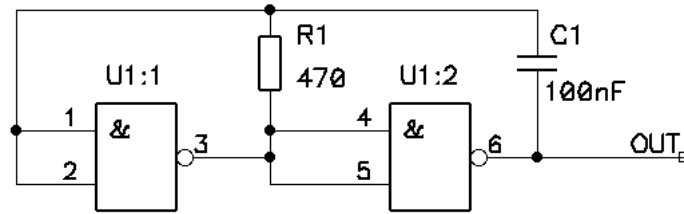


Рис. 2.1: Простейший мультивибратор.

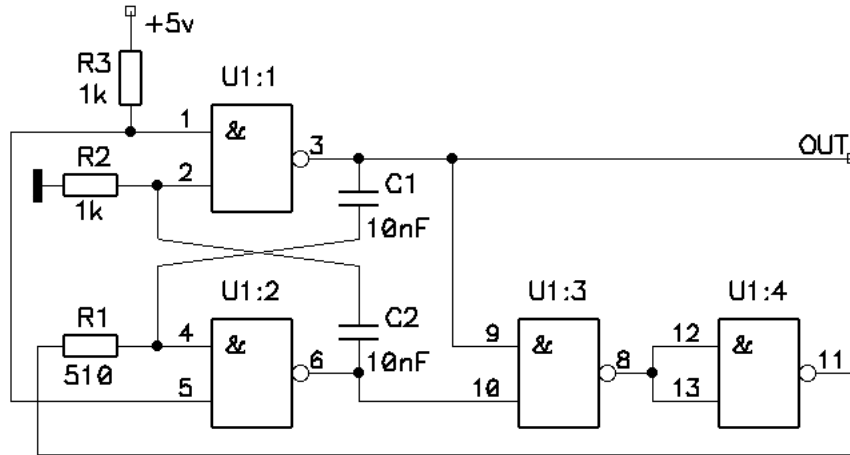


Рис. 2.2: Мультивибратор с заданной скважностью.

Скважность оценивается соотношением:

$$s = \frac{T_0}{\tau_1} \simeq 1 + \frac{R2 C2}{R1 C1}. \quad (2.3)$$

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

3. Ждущий мультивибратор (рис. 2.3) запускается фронтом отрицательного импульса (приходом логического 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 тоже время, τ_1 должно превышать удвоенное время задержки распространения сигнала через один элемент микросхемы $\tau_{\text{задержки}}$. Таким образом, при расчете можно использовать выражение

$$\tau_{\text{задержки}} \ll R1C1 \ll \tau. \quad (2.4)$$

Длительность импульса τ , формируемого ждущим мультивибратором задается цепочкой $R2C2$. На вход ждущего мультивибрато-

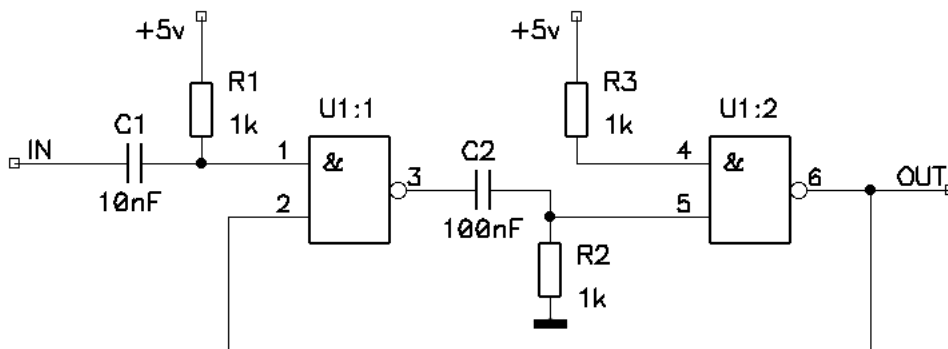


Рис. 2.3: Ждущий мультивибратор.

ра 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. Конструирование функций НЕ, ИЛИ, "Исключающее ИЛИ" на элементах 2И-НЕ. Упражнение выполняется в тетради. Для каждой схемы привести таблицу истинности.
5. Логическая схема по заданной функции (функция задается преподавателем). Упражнение выполняется в тетради. Заданная функция упрощается с применением карт Карно. Приводится таблица истинности.

2.2 Контрольные вопросы.

1. Булева алгебра. Коммутативность. Идемпотентность. Ассоциативность. Дистрибутивность. Комплементность. 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.

Задача посвящена изучению принципов работы простейших триггеров и их построению на основе логических элементов. Работа триггеров рассмотрена в [1] (задача 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 с средой ISE, программатором и макетной платой, на примере готового проекта, поставляемого разработчиками макетной платы (стр. 18). 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. 19.

3.1 Практическая часть

1. RS-триггер (рис. 3.1). Собрать схему. Входы \bar{R} и \bar{S} подключить к кнопкам 1 и 2. Выходы — к светодиодам 1 и 2. Записать таблицу истинности.
2. Синхронный RS-триггер (рис. 3.2). Собрать схему. Входы R, S и C подключить к кнопкам 1 – 3. Выходы — к светодиодам 1 и 2. Найти состояния хранения. Записать таблицу истинности.

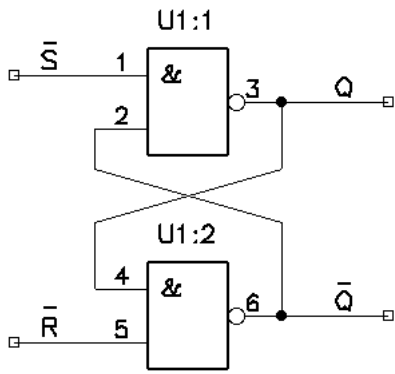


Рис. 3.1: RS-триггер

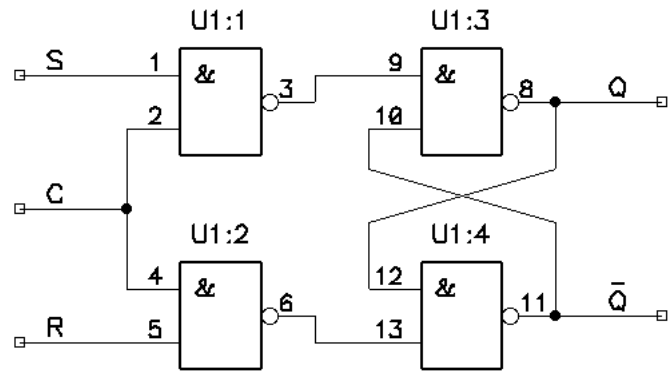


Рис. 3.2: Синхронный RC-триггер

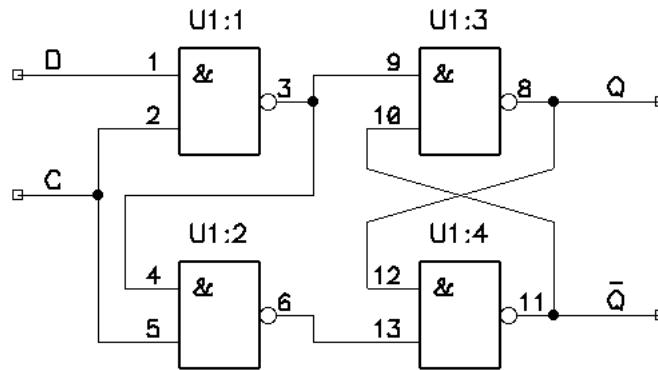


Рис. 3.3: D-триггер

3. D-триггер (рис. 3.3). Входы D и C подключить к кнопкам 3 и 4, выходы — к светодиодам 3 и 4. Записать таблицу истинности.
4. D-триггер, управляемый фронтом. Открыть сохраненный ранее проект RS-триггера. Не разбирая RS-триггера, разработать и собрать D-триггер, управляемый фронтом. Вход D подключить к кнопке 3, вход C — к выводу RS триггера. Выходы D-триггера — к светодиодам 3 и 4. Записать таблицу истинности.

3.2 Контрольные вопросы.

1. Какие виды триггеров вы знаете. Каково отличие триггера, управляемого по фронту, от управляемого по уровню сигнала?
2. Составить схему D-триггера, управляемого по фронту на основе логических элементов и триггеров, управляемых уровнем.
3. Составить таблицу истинности JK триггера, имеющего входы предустановки \bar{R} и \bar{S} .

Задача 4

Логические схемы и триггеры на VHDL.

Задача посвящена повторению упражнений, связанных с работой простых логических функций и триггеров. Работа схем подробно описана в [1] (задачи 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. 20.

4.1 Практическая часть

1. Получение функций 2И-НЕ, 2ИЛИ, "Исключающее ИЛИ". На языке VHDL функции описываются следующим образом:

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

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. RS-триггер, синхронный RS-триггер (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. Ждущий мультивибратор, D-триггер, 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;

```

Подключить ждущий мультивибратор к кнопке 1 и светодиоду 1. Описать самостоятельно 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. JK-триггер. Описать делитель частоты тактового генератора, чтобы получить период повторения 3 сек. Выход подключить к светодиоиду 1. Описать JK-триггер. Вход С 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 в языке VHDL отличаются сигналы (SIGNAL), переменные (VARIABLE) и константы (CONSTANT)? Где и когда они используются?
2. Чем отличаются между собой и в каких случаях используются следующие операции:

(a) =

(b) <=

(c) :=

В каких случаях можно вместо операции (b) использовать операцию (c)?

3. Что означает выражение

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

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

Задача 5

Счетчики, регистры, формирователи кодов

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] (задачи 8, 9) 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. Для двух последних упражнений необходимо выполнить симуляцию проекта в ISim. Порядок работы с симулятором описан в приложении на стр. 21.

5.1 Практическая часть

Параметры: N — максимальное число пересчета реверсивного счетчика (упражнение 2), N_k — количество элементов кода Баркера (упражнение 4).

1. Построение счетчика на 16. На VHDL схема, считающая число нажатий на кнопку 1 (со схемой устранения "дребезга контактов") и отображающая результат в двоичном коде на светодиодах 1–4 описывается следующим образом:


```

-- после описания IEEE библиотеки
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 истинности счетчика, начиная с состояния '0000' до '1111'.

2. 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. Коэффициент пересчета счетчика N . Сброс счетчика в положение '0000' должен осуществляться по нажатию на кнопку 3. Результат счета должен отображаться в двоичном коде на светодиодах 1–4.
3. 4-разрядный регистр сдвига. Описать на VHDL схему, представляющую собой регистр сдвига. Кнопка 1 является тактовым сигналом, 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.
4. Формирователь кода Баркера. Описать на VHDL схему, формирующую один из возможных кодов Баркера для заданного количества элементов кода N_k .

$N_k = 3$	1 1 -1
$N_k = 4$	1 1 -1 1
$N_k = 5$	1 1 1 -1 1
$N_k = 7$	1 1 1 -1 -1 1 -1
$N_k = 11$	1 1 1 -1 -1 -1 1 -1 -1 1 -1
$N_k = 13$	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.

5. M-последовательность и датчик случайных чисел. 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.

5.2 Контрольные вопросы.

1. Чем отличаются синхронные и асинхронные счетчики? Что понимается под счетчиком по модулю n ? Что такое реверсивный счетчик? Приведите схему счетчика, построенного на D-триггерах.
2. Что такое сдвиговый регистр? Что такое кольцевой регистр? Приведите схему сдвигового и кольцевого регистра, построенного на D-триггерах. Опишите сдвиговый и кольцевой регистр на VHDL.
3. Что такое код Баркера? Основные свойства радиосигнала, промодулированного кодом Баркера. Что такое M-последовательность? Ее свойства. Привести алгоритм синтеза M-последовательности с периодом повторения 255.

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 очистки 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>