|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий Кафедра Вычислительной Техники (ВТ)

**ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 8**

«Простейшие устройства ввода и индикации»» по дисциплине

«Архитектура вычислительных машин и систем»

|  |  |
| --- | --- |
| Выполнил студент группы ИВБО-01-22 | Зырянов М.А. |
| Принял ассистент кафедры ВТ | Дуксина И.И. |
| Практическая работа выполнена | « » 2023 г. |
| «Зачтено» | « » 2023 г. |

Москва 2023

1

**АННОТАЦИЯ**

2

Данная работа включает в себя 6 рисунка и 5 листинга. Количество страниц в работе — 16.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_bookmark0)

1. [СОЗДАНИЕ МОДУЛЕЙ В САПР VIVADO 5](#_bookmark1)
   1. [Описания модуля счетчика в САПР Vivado 5](#_bookmark2)
   2. [Описание модуля синхронизатора в САПР Vivado 6](#_bookmark3)
   3. [Описание фильтра дребезга контактов в САПР Vivado 7](#_bookmark4)
   4. [Реализация кнопки в САПР Vivado 10](#_bookmark5)
   5. [Реализация верификатора в САПР Vivado 12](#_bookmark6)
2. [ВЕРИФИКАЦИЯ МОДУЛЕЙ 14](#_bookmark7)

[ЗАКЛЮЧЕНИЕ 15](#_bookmark8)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 16](#_bookmark9)

# ВВЕДЕНИЕ

Одним из промежуточных итогов изучения языка Verilog HDL и САПР Vivado, в частности является создание простейших устройств ввода, которые смогут взаимодействовать с физическими объектами, в данном случае с кнопкой. В данной работе требуется создать модуль, описывающий условную работу с кнопкой, производимой пользователем, при этом используя знания из предыдущих работ, а именно универсальный счетчик. Так же для корректной работы кнопки в условиях реальной среды потребуется создать модуль описывающий синхронизатор. Так же, предварительной проверки корректности работы кнопки требуется создать тестовый модуль.

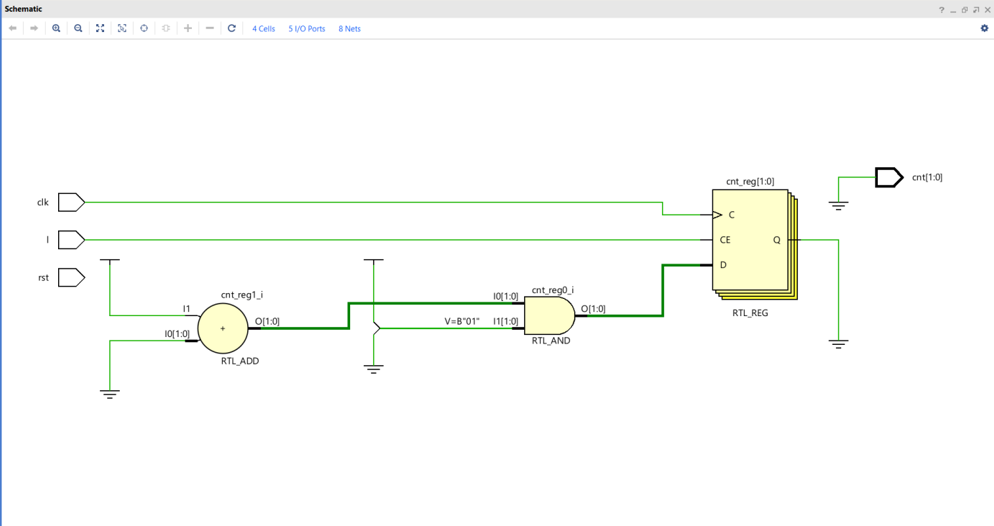
# СОЗДАНИЕ МОДУЛЕЙ В САПР VIVADO

* 1. **Описание модуля счетчика в САПР Vivado**

Модуль описывающий счетчик уже был описан в практической работе №6.

Принцип его работы был так же описан в данной практической работе.

Код представлен в Листинге 1.1, а RTL-схема на Рисунке 1.1.



**Рисунок 1.1 – RTL-схема счетчика**

*Листинг 1.1 – Модуль счетчика*

`timescale 1ns / 1ps

module counter #( STEP = 1,

MODULE = 2,

DIRECTION = 0,

SIZE = 2,

EDGE = 0)

( input clk, rst, l,

output reg [SIZE - 1:0] cnt

);

initial cnt <= 0;

*Продолжение листинга 1.1*

always @(rst) cnt <= 0;

always @(posedge clk) begin

if (EDGE == 0 || EDGE == 2)

begin if (l)

begin

if (~DIRECTION) cnt = (cnt + STEP) % MODULE;

else cnt = (cnt - STEP) % MODULE; end

end end

always @(negedge clk) begin

if (EDGE == 1 || EDGE == 2)

begin if (l)

begin

if (~DIRECTION)

cnt = (cnt + STEP) % MODULE;

else

cnt = (cnt - STEP) % MODULE;

end end

end endmodule

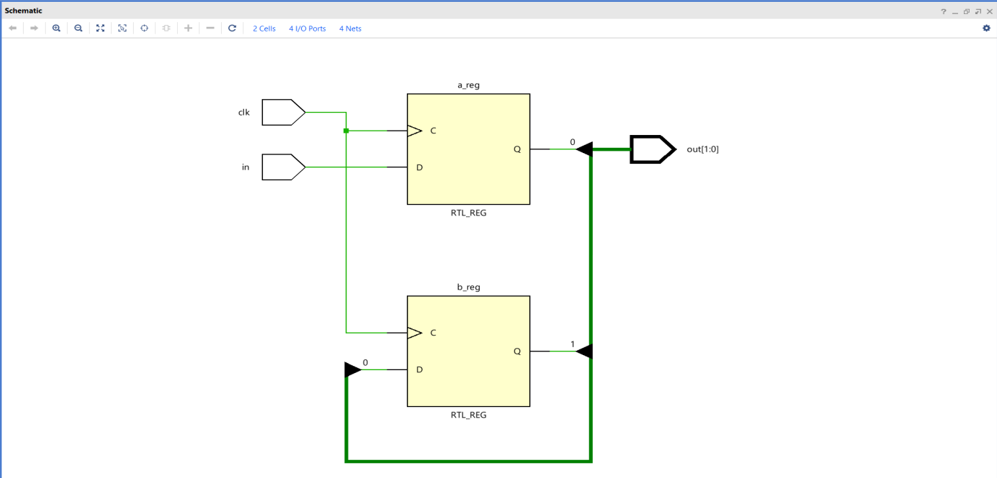
## Описание модуля синхронизатора в САПР Vivado

Используя указания строиться синхронизатор. Для него создается модуль в «design sources».

На него подаются две однобитные шины «clk», отвечающий за входные синхроимпульсы и «in», отвечающий за входные сигналы, так же потребуются два одноразрядных регистра «a», «b» и двухбитная шина «out», отвечающая за выходные сигналы.

Регистру «b» будет присваиваться значение регистра «a», которому присваивается значение «in», на переднем фронте синхросигнала. Шине «out» присваивается значение регистра «b» с помощью команды «assign».

Код представлен в Листинге 1.2, а RTL-схема на Рисунке 1.2.



**Рисунок 1.2 – RTL-схема синхронизатора**

*Листинг 1.2 – Модуль синхронизатора*

`timescale 1ns / 1ps module synchronizer(

input clk, in, output wire [1:0] out

);

reg a = 0, b = 0;

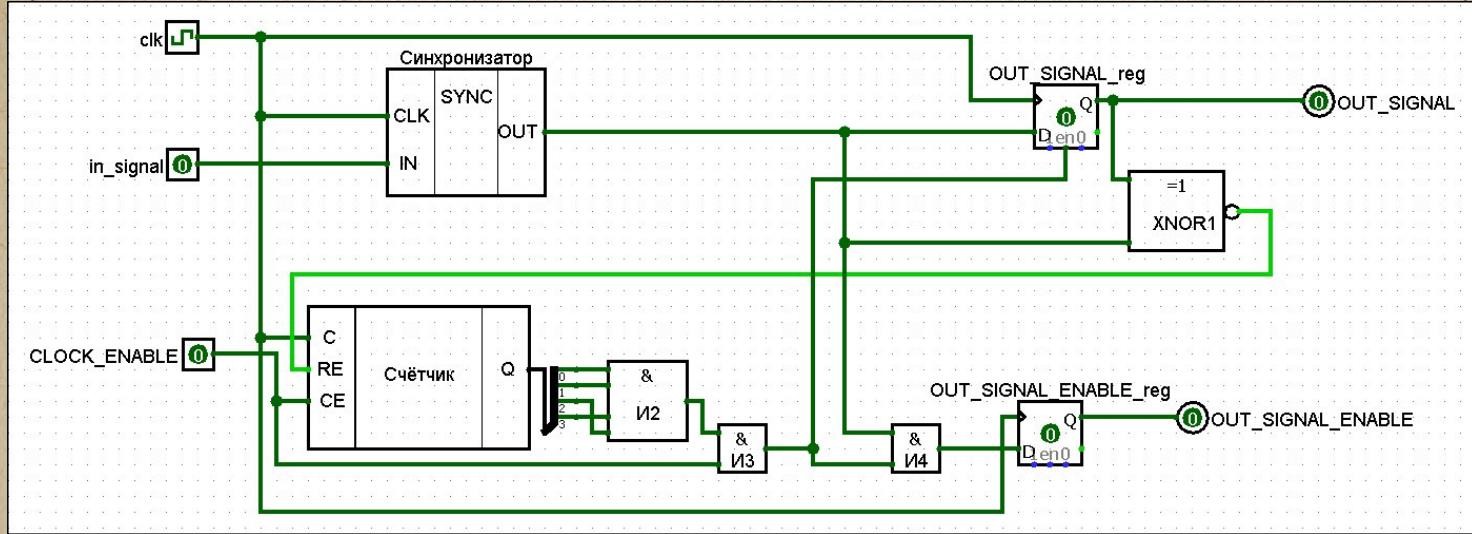
always @(posedge clk) begin

b <= a; a <= in; end

assign out = {b, a}; endmodule

## Описание фильтра дребезга контактов в САПР Vivado

Данный модуль был построен на основе схемы, показанной в методических указаниях к данной работе. Схема представлена на Рисунке 1.3.



**Рисунок 1.3 – Схема фильтра дребезга контактов**

Для описания фильтра дребезга контактов потребуется модуль в «design sources».

В нем будет девять однобитных шин: «clk», отвечающий за синхроимпульсы,

«CLOCK\_ENABLE», в качестве разрешающего сигнала на запись, «OUT\_SIGNAL», отвечающий за выходные данные, «in\_signal», отвечающий за входные данные,

«and1», «and2», «and3», «XNOR», все они отвечают за комбинационную логику в данном модуле. Еще понадобится двухбитная шина «sync», отвечающая за хранение сигнала синхронизатора, трехбитная шина «cnt», отвечающая за счет счетчика, два регистра на один бит «reg\_OUT\_SIGNAL\_ENABLE», «reg\_OUT\_SIGNAL», отвечающие за хранение промежуточных результатов.

Начинается все с инициализации регистров в нулевых значениях. Далее переносим параметризованный синхронизатор на два бита: шаг равен единице, направление как у универсального, размер на три бита, на сброс поступает команда с «XNOR», на счетчик подаются значения «cnt», на разрешающий сигнал поступает

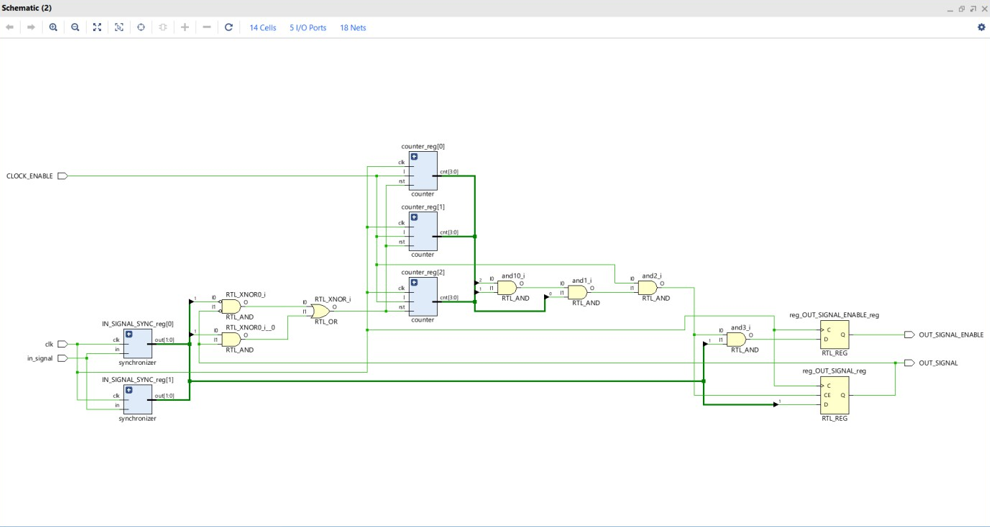
«CLOCK\_ENABLE». Шине «and1» присваивается значение конъюнкции значений счетчика «cnt [2:0]», шине «XNOR» присваивается значение «ИЛИ-НЕ по модулю» значений синхронизатора и регистра выходного сигнала («sync[1]»,

«reg\_OUT\_SIGNAL»), шине «and2» значение конъюнкции разрешающего сигнала на запись и первой конъюнкции («CLOCK\_ENABLE», «and1»), на шину «and3» значение конъюнкции второй конъюнкции и значение синхронизатора («and2»,

«sync[1]»), шине «OUT\_SIGNAL\_ENABLE», значение регистра выходного синала

(«reg\_OUT\_SIGNAL»). На восходящем фронте синхросигнала, если «and2» равен единице, регистру выходных значений («reg\_OUT\_SIGNAL») присваивается значение первого элемента шины синхронизации («sync»). Регистру дозволения записи выходного сигнала («reg\_OUT\_SIGNAL\_ENABLE») всегда присваивается значение «and3».

Код модуля описан в Листинге 1.3, а RTL-схема представлена на Рисунке 1.4.



**Рисунок 1.4 – RTL-схема дребезга контактов**

*Листинг 1.3 – Модуль дребезга контактов*

`timescale 1ns / 1ps

module filter(

input clk, CLOCK\_ENABLE, in\_signal,

output OUT\_SIGNAL, OUT\_SIGNAL\_ENABLE

);

wire [1:0] sync;

wire and1, and2, and3;

reg reg\_OUT\_SIGNAL = 0, reg\_OUT\_SIGNAL\_ENABLE = 0; wire [2:0] cnt;

wire XNOR;

synchronizer IN\_SIGNAL\_SYNC\_reg[1:0] (.clk(clk),

.in(in\_signal),

.out(sync));

*Продолжение листинга 1.3*

counter #(.STEP(1),

.MODULE(16),

.DIRECTION(0),

.SIZE(3),

.EDGE(0))

counter\_reg[2:0](.clk(clk),

.rst(XNOR),

.l(CLOCK\_ENABLE),

.cnt(cnt));

assign and1 = cnt[2] && cnt[1] && cnt[0]; assign and2 = CLOCK\_ENABLE && and1;

always @(posedge clk) begin

if (and2) reg\_OUT\_SIGNAL <= sync[1]; end

assign XNOR = (~sync[1] && ~reg\_OUT\_SIGNAL)

|| (sync[1] && reg\_OUT\_SIGNAL); assign and3 = and2 && sync[1];

always @(posedge clk) begin

reg\_OUT\_SIGNAL\_ENABLE <= and3; end

assign OUT\_SIGNAL\_ENABLE = reg\_OUT\_SIGNAL\_ENABLE; assign OUT\_SIGNAL = reg\_OUT\_SIGNAL;

endmodule

## Реализация кнопки в САПР Vivado

Для реализации кнопки в САПР Vivado потребуется модуль в «design sources». В нем будут две параметрические переменные «range» - время работы и

«waitfor» - время ожидания; три однобитные шины: «clk» - синхросигнал,

«sig\_out\_enable» - разрешение записи выходных данных, «sig\_out\_in\_input» - промежуточные результаты вычислений; три одноразрядных регистра: «sig\_in» - входной сигнал, «sig\_out» - выходной сигнал с кнопки; один регистр «sig\_out\_in» - регистр входных сигналов.

Регистр «sig\_out» инициализируется с нулевым значением. Далее циклически данному регистру присваиваются случайные значения, с промежутками в одну

единицу времени. После окончания цикла регистру выходных значений кнопки («sig\_out») присваивается значение единицы. Далее программа останавливается на значение параметра «waitfor» единиц времени, после чего идет повторение цикла. После цикла регистру выходного значения кнопки («sig\_out») присваивается нулевое значение, и программа опять переходит в режим ожидания на «waitfor» единиц времени.

При каждом изменении значения шины промежуточных результатов вычислений («sig\_out\_in\_input») регистру входных сигналов («sig\_out\_in») присваивается значение шины промежуточных результатов вычислений («sig\_out\_in\_input»). На каждом переднем фронте шины разрешения записи выходных данных («sig\_out\_enable») значение регистра выходных данных («sig\_out») инвертируется.

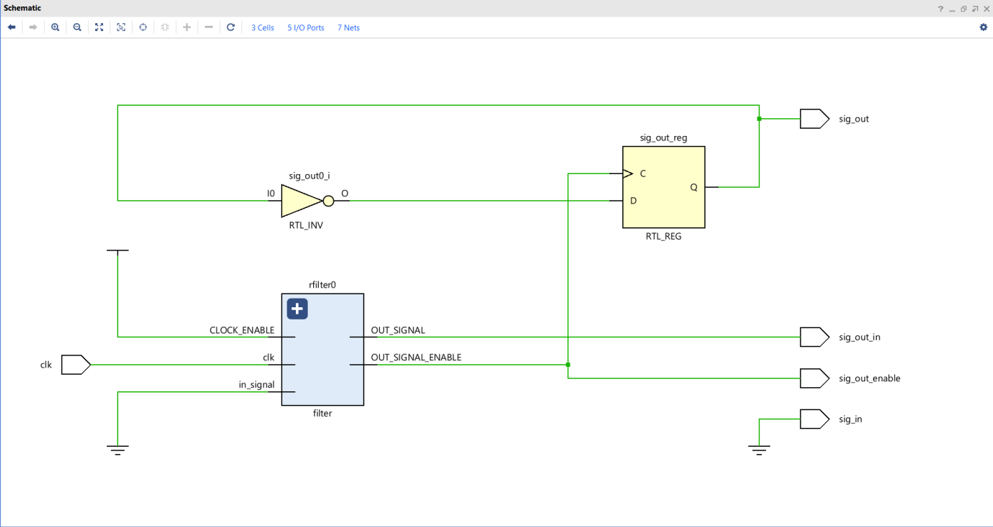
Далее идет инициализация модуля фильтра дребезга контактов. На его вход

«clk» подается «clk» данного модуля, на «CLOCK\_ENABLE» единица, на

«in\_signal» «sig\_in», на «OUT\_SIGNAL\_ENABLE» «sig\_out\_enable», на

«OUT\_SIGNAL» «sig\_out\_in\_input».

Код модуля описан в Листинге 1.3, а RTL-схема представлена на Рисунке 1.5.



**Рисунок 1.5 – RTL-схема кнопки**

*Листинг 1.4 – Модуль кнопки*

`timescale 1ns / 1ps

module Button #(range = 0, waitfor = 0) (

input clk,

output reg sig\_in,

output wire sig\_out\_enable, output reg sig\_out = 0, output reg sig\_out\_in

);

wire sig\_out\_in\_input; initial begin

repeat (range) begin sig\_in = $random; #1;

end

sig\_in = 1; #waitfor;

repeat (range) begin sig\_in = $random; #1;

end

sig\_in = 0; #waitfor;

repeat (range) begin sig\_in = $random; #1;

end

sig\_in = 1; #waitfor;

repeat (range) begin sig\_in = $random; #1;

end

sig\_in = 0; end

always @(sig\_out\_in\_input) sig\_out\_in <= sig\_out\_in\_input;

always @(posedge sig\_out\_enable) sig\_out = ~sig\_out;

filter rfilter0(.clk(clk),

.CLOCK\_ENABLE(1),

.in\_signal(sig\_in),

.OUT\_SIGNAL\_ENABLE(sig\_out\_enable),

.OUT\_SIGNAL(sig\_out\_in\_input)); endmodule;

## Реализация верификатора в САПР Vivado

Для реализации верификатора в САПР Vivado, требуется создать модуль в

«simulation sources» [3], данный модуль будет называться «testbench».

В модуле будет один одноразрядный регистр «clk» - синхросигнал, равный при инициализации значению 0, и три однобитный шины: «sig\_in» - входной сигнал,

«sig\_out\_enable» - разрешение записи выходного сигнала, «sig\_out\_in» - выходной сигнал. Генерируем такты путем инвертирования регистра «clk» каждые 10 единиц времени, с помощью блока «always». Инициализируем модуль управления с параметрами «range» равного одной сотне и «waitfor» равного тремстам, на «clk» подаем «clk» данного модуля, на «sig\_in» подаем «sig\_in» данного модуля, на

«sig\_out\_enable» подаем «sig\_out\_enable» данного модуля, на «sig\_out» подаем

«sig\_out» данного модуля, на «sig\_out\_in» подаем «sig\_out\_in» данного модуля.

Код представлен в Листинге 1.5.

*Листинг 1.5 – Модуль верификатора*

`timescale 1ns / 1ps module testbench();

reg clk = 0;

wire sig\_in, sig\_out\_enable, sig\_out, sig\_out\_in;

always #10 clk = ~clk;

Button #(.range(100), .waitfor(300))

button(.clk(clk),

.sig\_in(sig\_in),

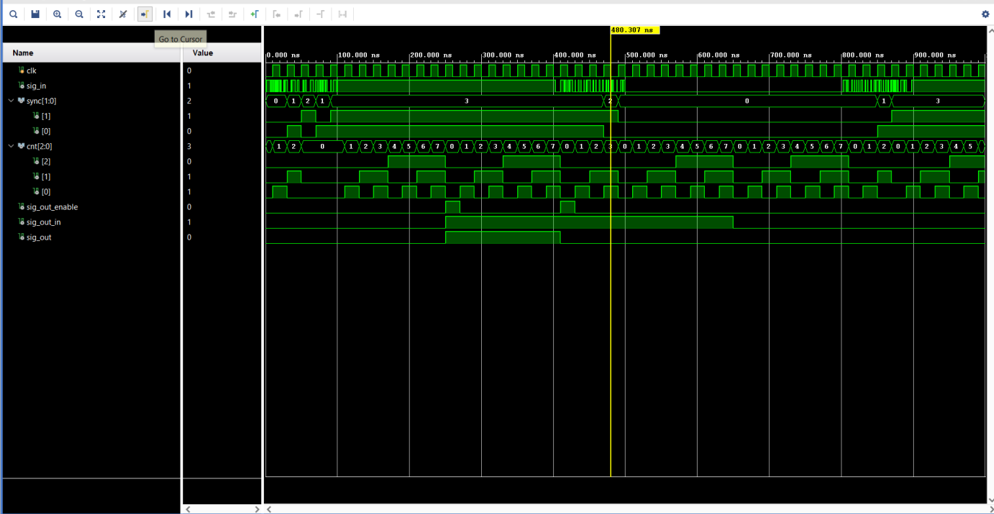
.sig\_out\_enable(sig\_out\_enable),

.sig\_out(sig\_out),

.sig\_out\_in(sig\_out\_in)); endmodule

# ВЕРИФИКАЦИЯ МОДУЛЕЙ

Произведем верификацию модулей посредством временных диаграмм (Рисунок 2.6).



**Рисунок 2.6 – Временная диаграмма автоматов**

Как видно, переключение происходит при длительном чтении сигнала с кнопки, после чего поступает разрешающий на запись сигнал и идет дальнейшее чтение приходящей информации.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной практической работы были созданы и реализованы модули синхронизатора, фильтра дребезга контактов, кнопки, так же был реализована проверкана корректность на языке описания аппаратуры Verilog HDL.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методические указания по ПР № 1 — URL: https://online- edu.mirea.ru/mod/resource/view.php?id=405132 (Дата обращения: 19.09.2023).
2. Методические указания по ПР № 2 — URL: https://online- edu.mirea.ru/mod/resource/view.php?id=409130 (Дата обращения: 19.09.2023).
3. Смирнов С.С. Информатика [Электронный ресурс]: Методические указания по выполнению практических и лабораторных работ / С.С. Смирнов — М., МИРЭА

* Российский технологический университет, 2018. — 1 электрон. опт. диск (CD- ROM).

1. Тарасов И.Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования / И.К. Тарасов — Горячая Линия - Телеком, 2022.

* 538 стр. — ISBN 978-5-9912-0802-4.

1. Методические указания по ПР № 5 — URL: https://online- edu.mirea.ru/mod/resource/view.php?id=529522 (Дата обращения: 30.11.2023).
2. Методические указания по ПР № 6 — URL: https://online- edu.mirea.ru/mod/resource/view.php?id=413203 (Дата обращения: 06.12.2023).
3. Методические указания по ПР № 7 — URL: https://online- edu.mirea.ru/mod/resource/view.php?id=413206 (Дата обращения: 30.11.2023).
4. Методические указания по ПР № 8 — URL: https://online- edu.mirea.ru/mod/resource/view.php?id=413209 (Дата обращения: 06.12.2023).