**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В. И. Ульянова (Ленина)**

**Кафедра информационной безопасности**

отчет

**по лабораторной работе №1**

**по дисциплине «Электроника и схемотехника»**

Тема: **Установка Gowin EDA, Icarus Verilog (iverilog), Verilog Virtual Processor (vvp), GTKWave. Симуляция тестового проекта**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3363 |  | Минко Д. А. |
| Студент гр. 3363 |  | Гончаренко О. Д. |
| Студент гр. 3363 |  | Овсейчик Н. И. |
| Преподаватель |  | Рыбин В. Г. |

**Цель работы**

Целью данной работы является установка, требуемого программного обеспечения: Gowin EDA, Icarus Verilog (iverilog), Verilog Virtual Processor (vvp), GTKWave, и симуляция тестового проекта.

**Ход работы**

1. Установка программного пакета для симуляции

Были загружены и установлены компилятор Icarus Verilog, среда исполнения Verilog Virtual Processor и визуализатор временных диаграмм GTKWave. Теперь можно приступать к проведению временной симуляции Verilog-проектов.

1. Установка Gowin EDA Education

Установка и настройка программы Gowin EDA для разработки на ПЛИС компании Gowin выполнена. Были установлены необходимые драйверы, порты были подготовлены к использованию. После настройки COM-портов потребовалась перезагрузка системы. Теперь можно приступать к разработке для ПЛИС в среде Gowin EDA.

1. Тестовый проект runner

Открыт и отредактирован проект "runner.gprj" в программе Gowin EDA. Изучены файлы "runner.v" и "runner\_tb.v", где в первом установлено переключение светодиодов каждые 4.500.000 тактов для симуляции, а во втором настроен тестбенч для работы с тактирующим сигналом, сигналом сброса и генерации выходного файла симуляции.

1. Компиляция модуля runner и его тестбенча с помощью Icarus Verilog

После открытия терминала в папке проекта была выполнена команда "iverilog -o ./compiled ./src/runner.v ./src/runner\_tb.v" для компиляции файлов runner.v и runner\_tb.v. Выходной файл был назван compiled.

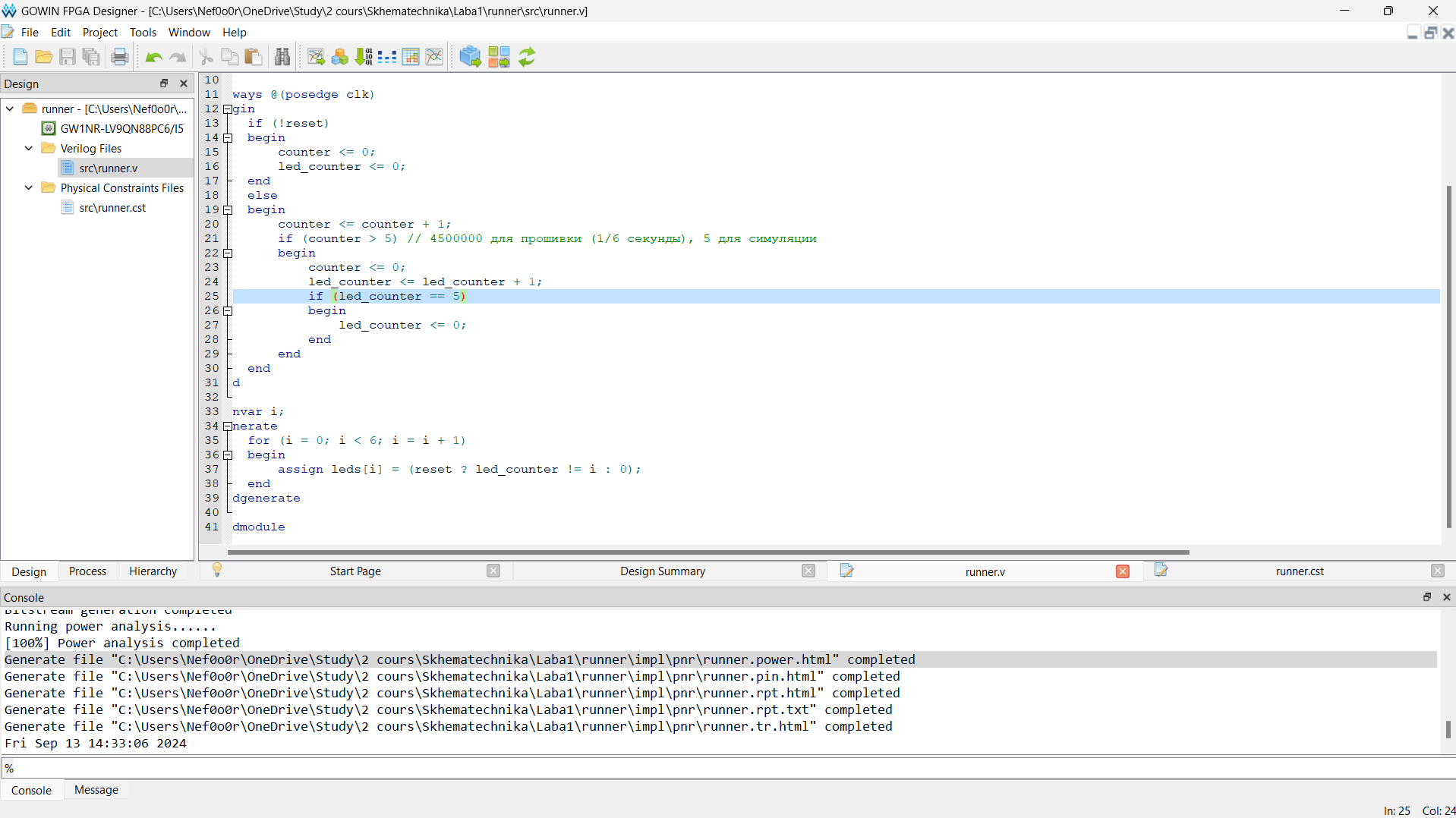


Рисунок 1 – Код открытый в Gowin

1. Симуляция модуля с помощью VVP

Далее была выполнена команда "vvp ./compiled" для проведения симуляции с использованием Verilog Virtual Processor. Результаты симуляции были записаны в файл runner\_out.vcd в папке с проектом, описывающий изменения в значениях переменных на протяжении симуляции.

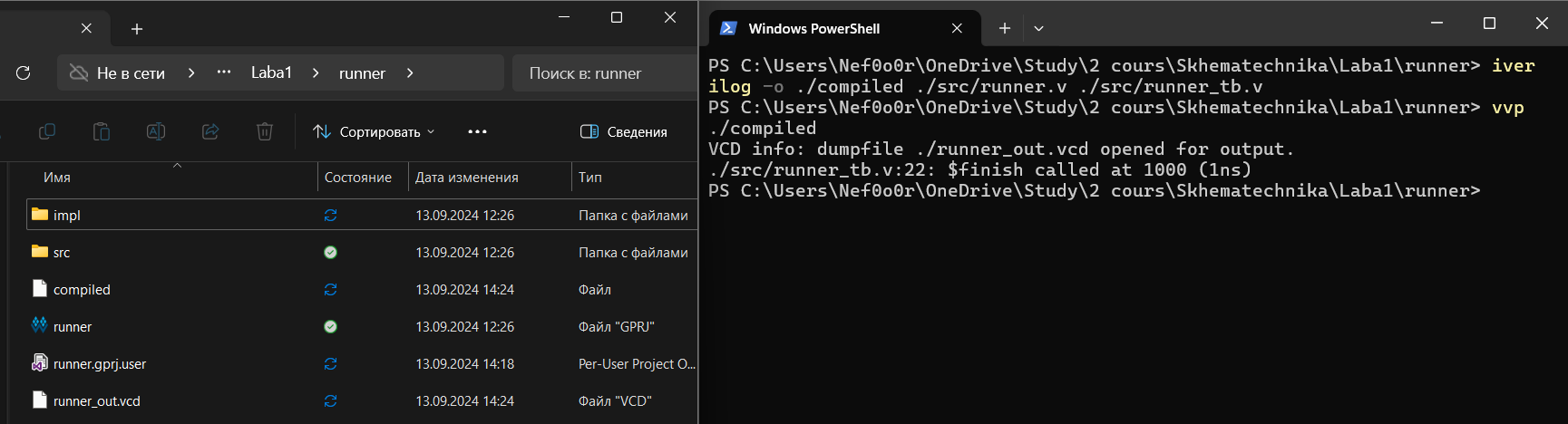


Рисунок 2 – Компиляция и симуляция модуля runner

1. Просмотр полученной временной диаграммы в GTKWave

После открытия программы GTKWave из папки gtkwave64 в подпапке bin, была создана новая вкладка через команду Open New Tab в меню File, после чего был выбран ранее полученный файл данных vcd. В окне SST был выбран модуль runner\_tb для отображения его переменных в нижнем окне. Сигналы были перетащены в основное окно справа для визуализации. Для просмотра всей длительности симуляции были использованы кнопки меню сверху или колесико мыши с зажатой клавишей Ctrl.

Для представления сигнала leds в двоичном формате было выполнено нажатие правой кнопкой мыши на сигнал в меню Signals, а затем выбран формат Binary в меню Data Format. В результате стало заметно, что ноль бегает по битам массива leds, что означает включение одного из диодов на реальной плате.

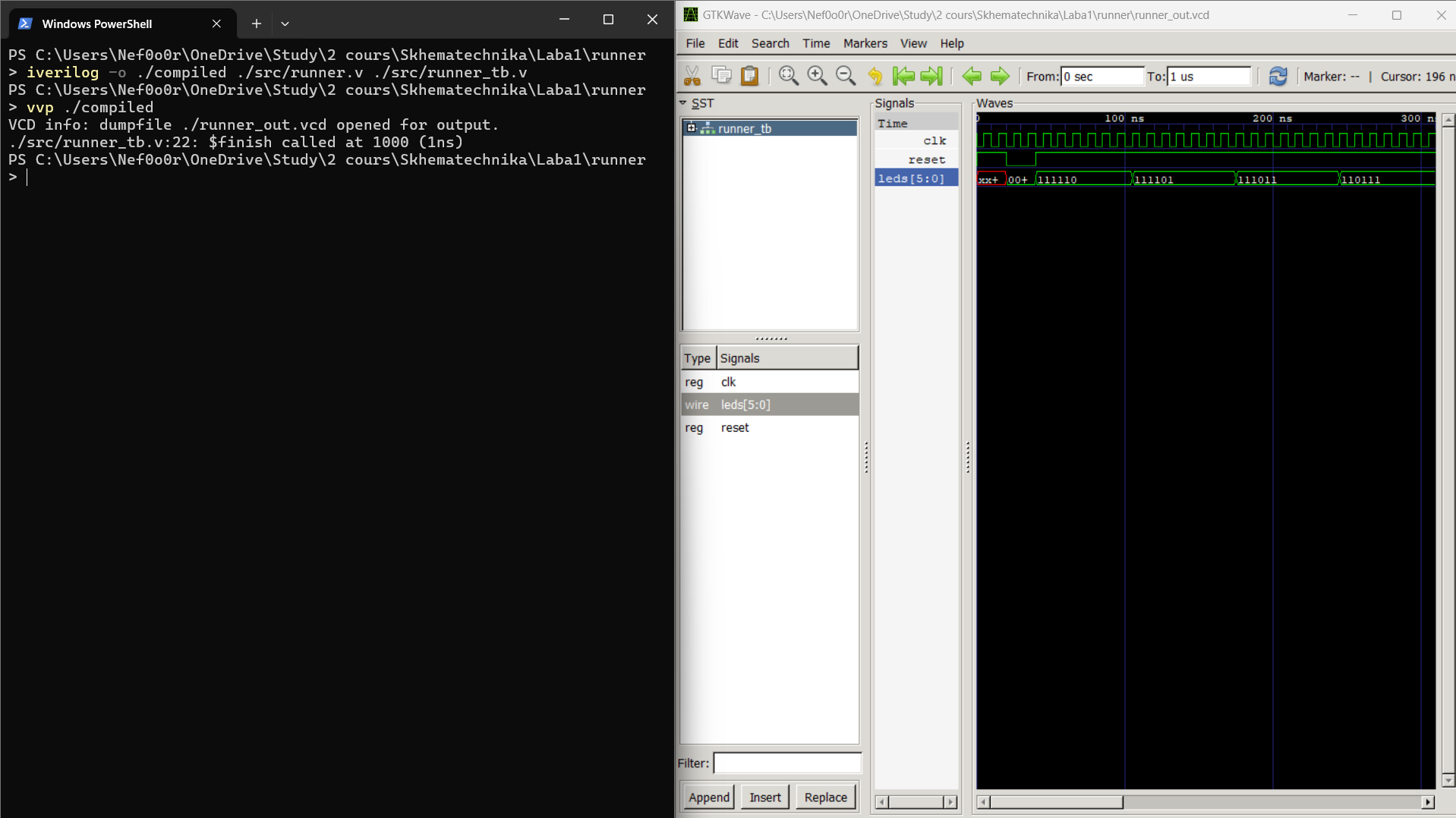


Рисунок 3 – Временная диаграмма, полученная в GTKWave

Сделана блок-схема алгоритма работы данной программы (рис. 4).

Изображение выглядит как текст, снимок экрана, Самоклеющийся листок, Шрифт

Автоматически созданное описание

Рисунок 4 – Блок-схема кода

**Вывод**

В результате выполнения данной работы было установлено, требуемое программное обеспечение: Gowin EDA, Icarus Verilog (iverilog), Verilog Virtual Processor (vvp), GTKWave, и реализована симуляция тестового проекта.

ИСХОДНЫЙ КОД

module runner

(

input clk, // Тактирующий сигнал

input reset, // Сигнал сброса (1 по умолчанию, 0 при сбросе)

output [5:0] leds // Выходы светодиодов

);

reg [22:0] counter; // Счётчик для переключения светодиодов

reg [2:0] led\_counter; // Состояние

always @(posedge clk)

begin

if (!reset)

begin

counter <= 0;

led\_counter <= 0;

end

else

begin

counter <= counter + 1;

if (counter > 5) // 4500000 для прошивки (1/6 секунды), 5 для симуляции

begin

counter <= 0;

led\_counter <= led\_counter + 1;

if (led\_counter == 5)

begin

led\_counter <= 0;

end

end

end

end

genvar i;

generate

for (i = 0; i < 6; i = i + 1)

begin

assign leds[i] = (reset ? led\_counter != i : 0);

end

endgenerate

endmodule