**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

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

**Кафедра информационной безопасности**

отчет

**по лабораторной работе №3**

**по дисциплине «Электроника и схемотехника»**

**Тема: Использование кнопочной клавиатуры и**

**DIP-переключателей**

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

Санкт-Петербург

2024

**Цель работы**

Ознакомиться с принципом работы светодиодов и кнопок на плате Tang Nano 9K, изучить процесс управления состоянием светодиодов с использованием переключателей и кнопок. Разработать модуль для управления светодиодами, реализующий функционал инверсии состояния светодиодов при нажатии кнопок, провести симуляцию модуля в среде GTKWave и протестировать работу на плате

**Задание**

Реализовать управление светодиодами с помощью переключателей и кнопок. Переключатели должны включать и выключать светодиоды, а кнопки — инвертировать текущее состояние светодиодов. Провести симуляцию работы модуля в среде GTKWave и протестировать работоспособность на плате Tang Nano 9K.

**Используемые аппаратные и программные средства**

— Плата Tang Nano 9K

— Шесть светодиодов

— Две кнопки

— Среда разработки Gowin EDA

— Среда для симуляции GTKWave

— Язык описания оборудования Verilog

**Ход работы**

* 1. **Реализация модуля**

Модуль предназначен для управления тремя светодиодами () на основе состояния переключателей () и кнопки (). На вход модуля поступают тактовый сигнал , три значения переключателей, а также сигнал кнопки, отвечающей за изменение состояния светодиодов. При запуске модуля инициализируется переменная , отвечающая за состояние инверсии светодиодов. Эта переменная изменяет свое значение при каждом нажатии одной из кнопок.

Основная логика работы модуля заключается в следующем:

1. **Инициализация переменных**

При запуске модуля происходит инициализация:

* Переменная , которая определяет режим инверсии светодиодов, изначально устанавливается в значение 0 (режим без инверсии).
* Переменная используется для подавления дребезга кнопки.
* Временная переменная хранит предыдущее состояние кнопки для отслеживания нажатий.
* Переменная задаёт начальное состояние всех светодиодов.

1. **Антидребезг кнопки**

При каждом фронте тактового сигнала () реализуется обработка дребезга кнопки.

* Переменная увеличивается на каждом такте.
* Если значение превышает установленный порог ( тактов), и состояние кнопки изменилось (нажатие), переменная меняет свое значение. Это переключает режим инверсии светодиодов.
* После этого таймер сбрасывается.

1. **Управление светодиодами**

В зависимости от значения переменной , светодиоды отображают:

* Инвертированное состояние переключателей ), если .
* Прямое состояние переключателей (), если .

Состояние светодиодов обновляется в переменной , а затем передаётся на выход через оператор .

1. **Назначение кнопки**

Кнопка позволяет инвертировать состояние всех светодиодов, мгновенно изменяя их поведение.

1. **Назначение переключателей**

Переключатели определяют, какие именно светодиоды включены или выключены. Их состояние напрямую влияет на светодиоды в зависимости от режима инверсии.

Полный исходный код модуля приведен в приложении 1.

* 1. **Реализация тестбенча**

В тестбенче создаётся экземпляр модуля , обеспечивающий проверку корректности его работы. Для проведения тестирования генерируется тактовый сигнал с частотой 100 МГц, позволяющий симулировать условия реальной работы устройства.

Симуляция проводится на протяжении 1000 наносекунд, а результаты сохраняются в файл для последующего анализа.

**Основные этапы работы тестбенча**

1. Генерация тактового сигнала:

Тактовый сигнал генерируется с периодом 10 нс (частота 100 МГц) с использованием бесконечного цикла, который инвертирует значение каждые 5 нс.

1. Инициализация начальных условий:

В начальный момент времени устанавливаются значения переключателей () и кнопки ().

1. Тестовые сценарии:

* Поочерёдно проверяется работа светодиодов в различных состояниях переключателей.
* Выполняется тестирование работы инверсии при нажатии кнопки .

1. Сохранение результатов симуляции:

Для анализа поведения модуля результаты симуляции сохраняются в формате VCD в файл .

Полный исходный код тестбенча приведен в приложении 2.

* 1. **Разводка**

При разводке модуля для переключателей были выбраны пины с 79 по 81. Данные пины располагаются на плате последовательно (рисунок 2) , что упрощает сборку схемы. Для кнопки был выбран пин 3 (рисунок 3).

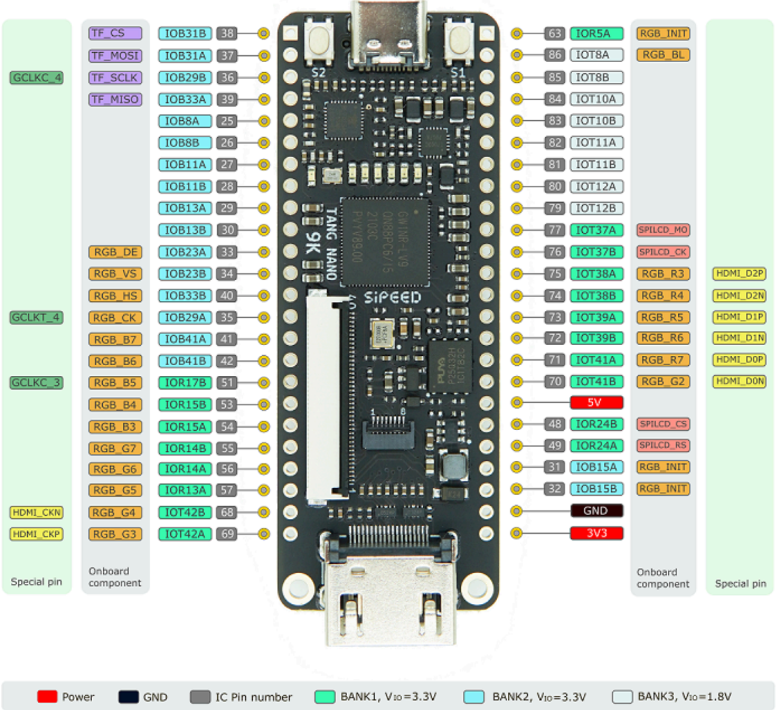


Рисунок 2 — Пины платы

Тактирующий сигнал находится на 52 пине, массив светодиодов leds находятся с 10 – 11 и 13. в соответствии с документацией к плате.

Полная разводка модуля приведена на рисункке 3.

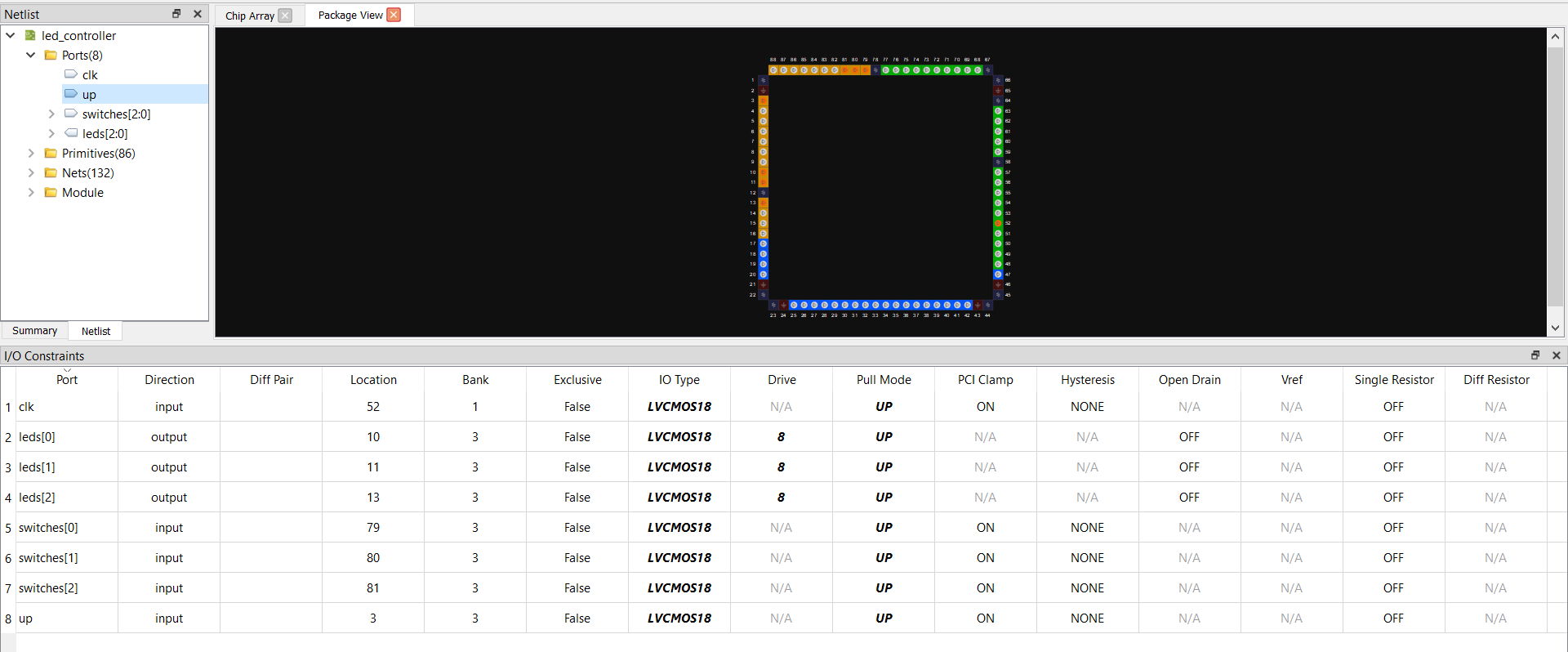


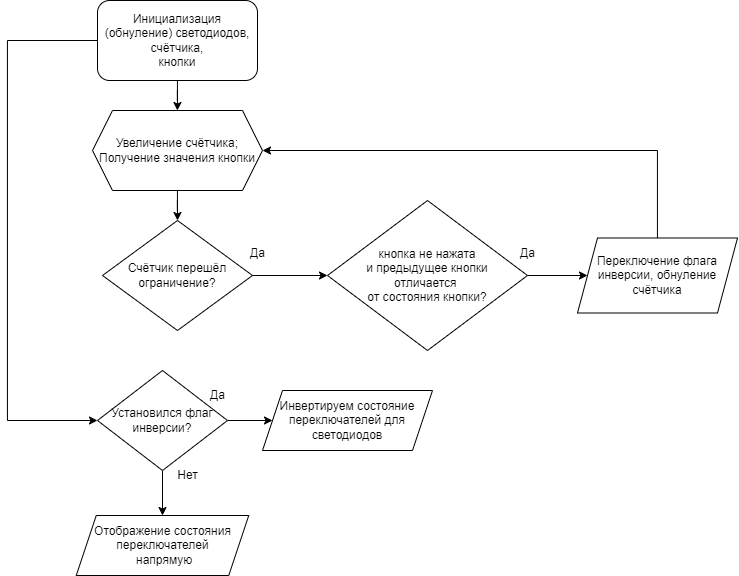
Рисунок 3 — Разводка модуля

**1.5 Запуск на плате**

**Вывод**

В ходе работы был разработан модуль управления светодиодами с использованием кнопок и переключателей на плате Tang Nano 9K. Модуль корректно реализует переключение состояний светодиодов в зависимости от положения переключателей и состояния кнопок. При нажатии кнопки активируется режим инверсии, что позволяет менять отображение состояния светодиодов. В процессе работы были изучены принципы управления светодиодами с использованием логических сигналов и функции инверсии.

**ПРИЛОЖЕНИЕ 1 — БЛОК — СХЕМА ПРОГРАММЫ**

****

**ПРИЛОЖЕНИЕ 2 — ИСХОДНЫЙ КОД МОДУЛЯ**

**Файл Lab\_3.v:**

module led\_controller

(

input wire clk,

input wire [2:0] switches, // Три переключателя

input up, // Кнопка для инверсии состояния

output [2:0] leds // Три светодиода

);

reg invert = 0; // Флаг для инверсии состояния светодиодов

reg [31:0] cooldown; // Таймер для обработки дребезга

reg temp\_button;

reg [2:0] to\_leds;

initial

begin

to\_leds = 3'b111;

cooldown <= 0;

temp\_button <= 1;

end

// Логика инверсии состояния при нажатии кнопки

always @(posedge clk) begin

cooldown <= cooldown + 1;

temp\_button <= up; //значение кнопки

if(cooldown > 4194304)

begin

if(up && temp\_button != up)

begin

invert <= ~invert; // Переключаем флаг инверсии

cooldown <= 0; //

end

end

end

// Управление светодиодами в зависимости от флага инверсии и состояния переключателей

always @(posedge clk) begin

if (invert) begin

to\_leds <= ~switches; // Инвертируем состояние переключателей для светодиодов

end

else begin

to\_leds <= switches; // Отображаем состояние переключателей напрямую

end

end

assign leds = to\_leds;

endmodule

**ПРИЛОЖЕНИЕ 2 — ИСХОДНЫЙ КОД ТЕСТБЕНЧА**

**Файл indicator\_tb.v:**

`timescale 1ns/1ps

module led\_controller\_tb;

reg clk; // Тактовый сигнал

reg [2:0] switches; // Переключатели

reg up; // Кнопка инверсии

wire [2:0] leds; // Светодиоды

// Экземпляр тестируемого модуля

led\_controller uut (

.clk(clk),

.switches(switches),

.up(up),

.leds(leds)

);

// Генерация тактового сигнала (100 МГц)

initial begin

clk = 0;

forever #5 clk = ~clk; // Период 10 нс

end

// Основной блок тестирования

initial begin

// Открытие файла для записи результатов

$dumpfile("led\_controller\_tb\_out.vcd");

$dumpvars(0, led\_controller\_tb);

// Инициализация входов

switches = 3'b000;

up = 1;

// Тест 1: Проверка начального состояния

#20 switches = 3'b101; // Устанавливаем переключатели

#20 up = 0; // Нажимаем кнопку

#10 up = 1; // Отпускаем кнопку

#100; // Ждём обработки инверсии

// Тест 2: Инверсия состояния

#50 switches = 3'b010; // Изменяем состояние переключателей

#20 up = 0; // Нажимаем кнопку для инверсии

#10 up = 1; // Отпускаем кнопку

#100;

// Тест 3: Включение всех светодиодов

#50 switches = 3'b111;

#50;

// Завершение симуляции

$finish;

end

endmodule