

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
- Три светодиода
- Одна кнопка
- Среда разработки Gowin EDA
- Среда для симуляции GTKWave
- Язык описания оборудования Verilog

Ход работы

1.1. Реализация модуля

Модуль *led_controller* предназначен для управления тремя светодиодами (*leds*) на основе состояния переключателей (*switches*) и кнопки (*up*). На вход модуля поступают тактовый сигнал *clk*, три значения переключателей, а также сигнал кнопки, отвечающей за изменение состояния светодиодов. При запуске модуля инициализируется переменная *invert*, 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 *invert*, которая определяет режим инверсии светодиодов, изначально устанавливается в значение 0 (режим без инверсии).
- Переменная *cooldown* используется для подавления дребезга кнопки.
- Временная переменная *temp_button* хранит предыдущее состояние кнопки для отслеживания нажатий.
- Переменная *to_leds* задаёт начальное состояние всех светодиодов.

2. Антидребезг кнопки

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

- Переменная *cooldown* увеличивается на каждом такте.
- Если значение *cooldown* превышает установленный порог (2^{22} тактов), и состояние кнопки изменилось (нажатие), переменная *invert* меняет свое значение. Это переключает режим инверсии светодиодов.
- После этого таймер *cooldown* сбрасывается.

3. Управление светодиодами

В зависимости от значения переменной *invert*, светодиоды отображают:

- Инвертированное состояние переключателей (*~switches*), если *invert* = 1.

- Прямое состояние переключателей (*switches*), если *invert* = 0.

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

4. Назначение кнопки

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

5. 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.

1.2. Реализация тестбенча

В тестбенче создаётся экземпляр модуля *led_controller*, 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 *led_controller_tb_out.vcd* для последующего анализа.

Основные этапы работы тестбенча

1. Генерация тактового сигнала:

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

2. Инициализация начальных условий:

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

3. Тестовые сценарии:

- Поочерёдно проверяется работа светодиодов в различных состояниях переключателей.

- Выполняется тестирование работы инверсии при нажатии кнопки *up*.

4. Сохранение результатов симуляции:

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

Полный исходный код тестбенча приведен в приложении 2.

1.3. 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).

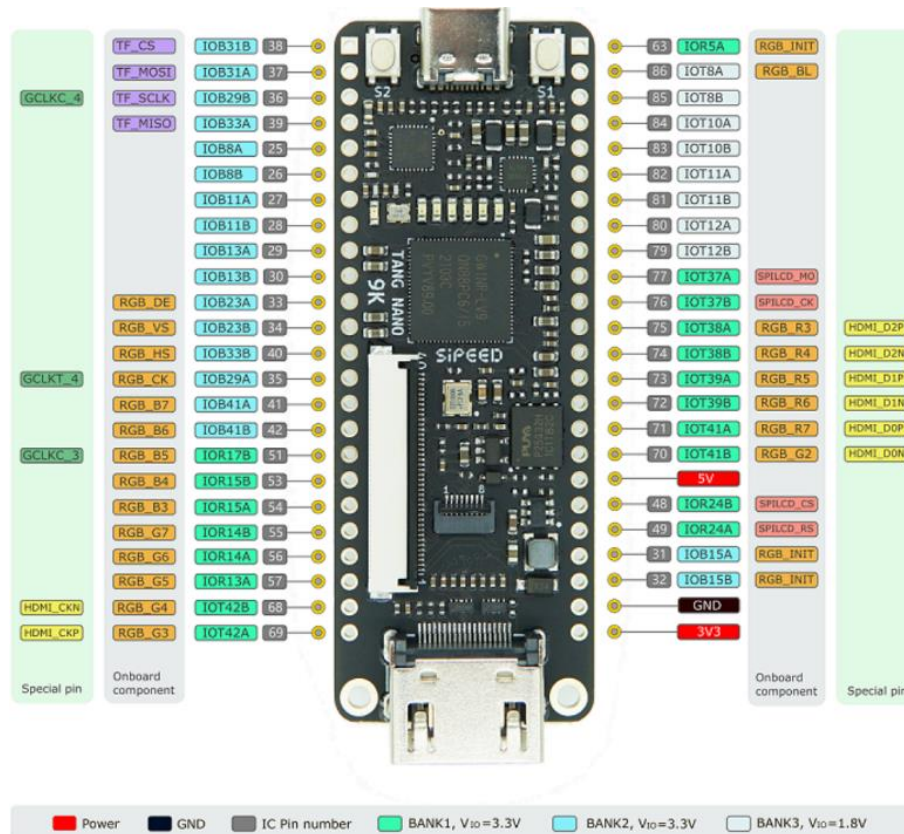


Рисунок 2 — Пины платы

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 рисунке 3.

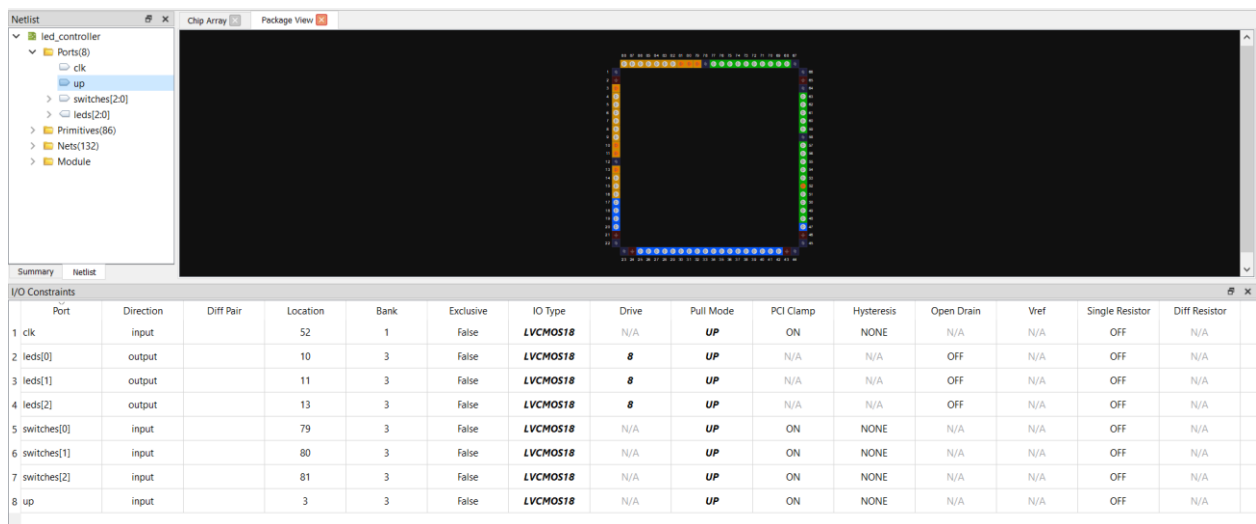


Рисунок 3 — Разводка модуля

1.5 Запуск на плате

При запуске программы соответственно подключаем GND, вставляем провода в пины, которые отвечают за светодиоды и подключаем к земле.

Начальное состояние платы (рис.4)

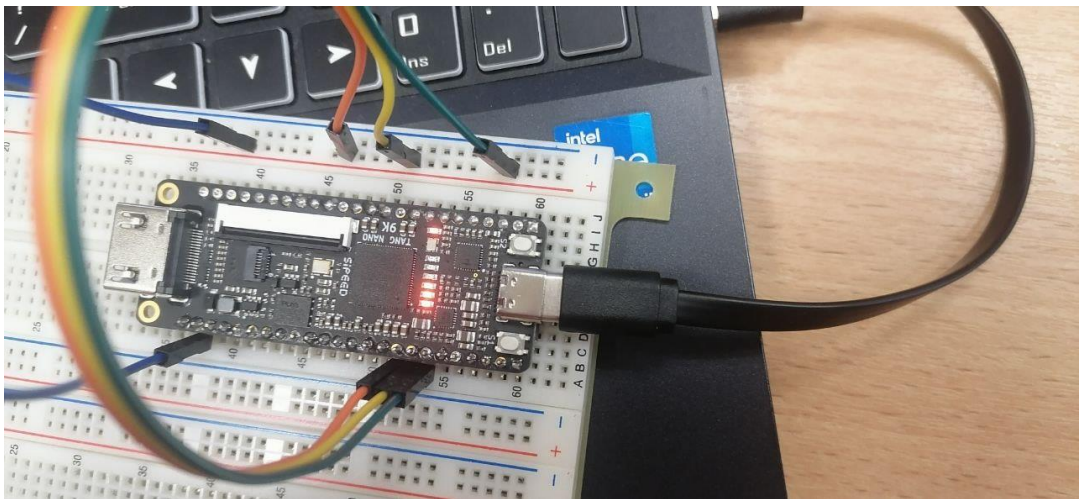


Рисунок 4 – Начальное состояние платы

Далее вытащим два провода. Так как наши провода – это реализация DIP – переключателей, наши два светодиода, согласно проводам, потухнут (рис.5)

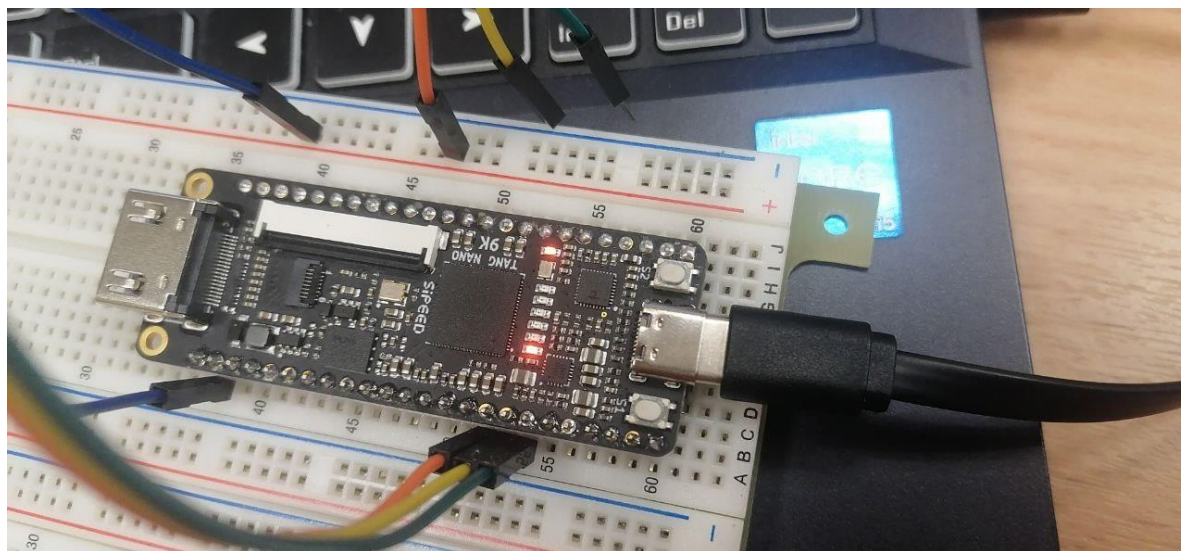


Рисунок 5 – Реализация DIP – переключателей на плате

Соответственно, после нажатия кнопки *up* работает флаг invert, и наше состояние светодиодов поменяется в противоположную сторону (если горел один и два потухли, то теперь первый потух, а остальные два загорелись) (рис.6)

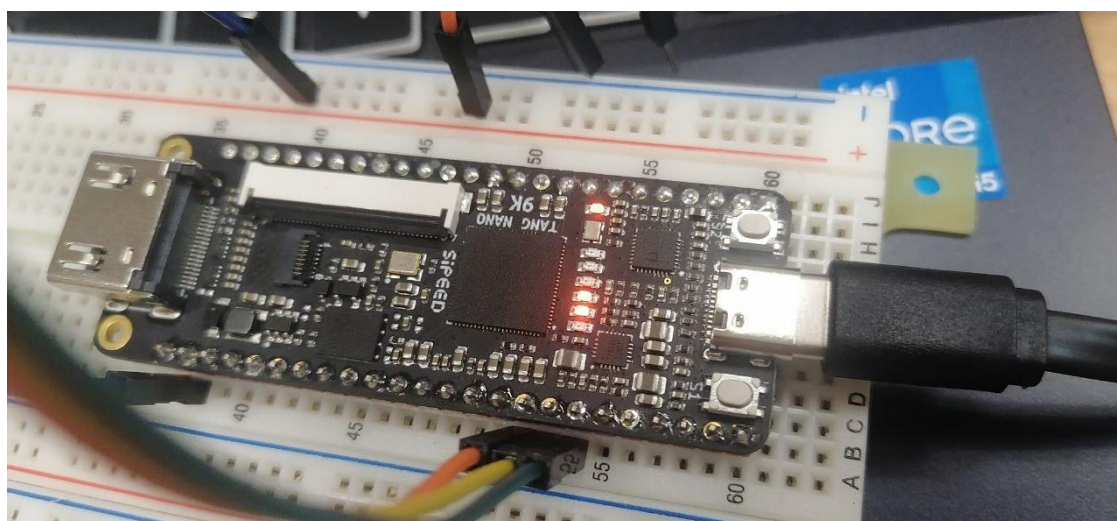
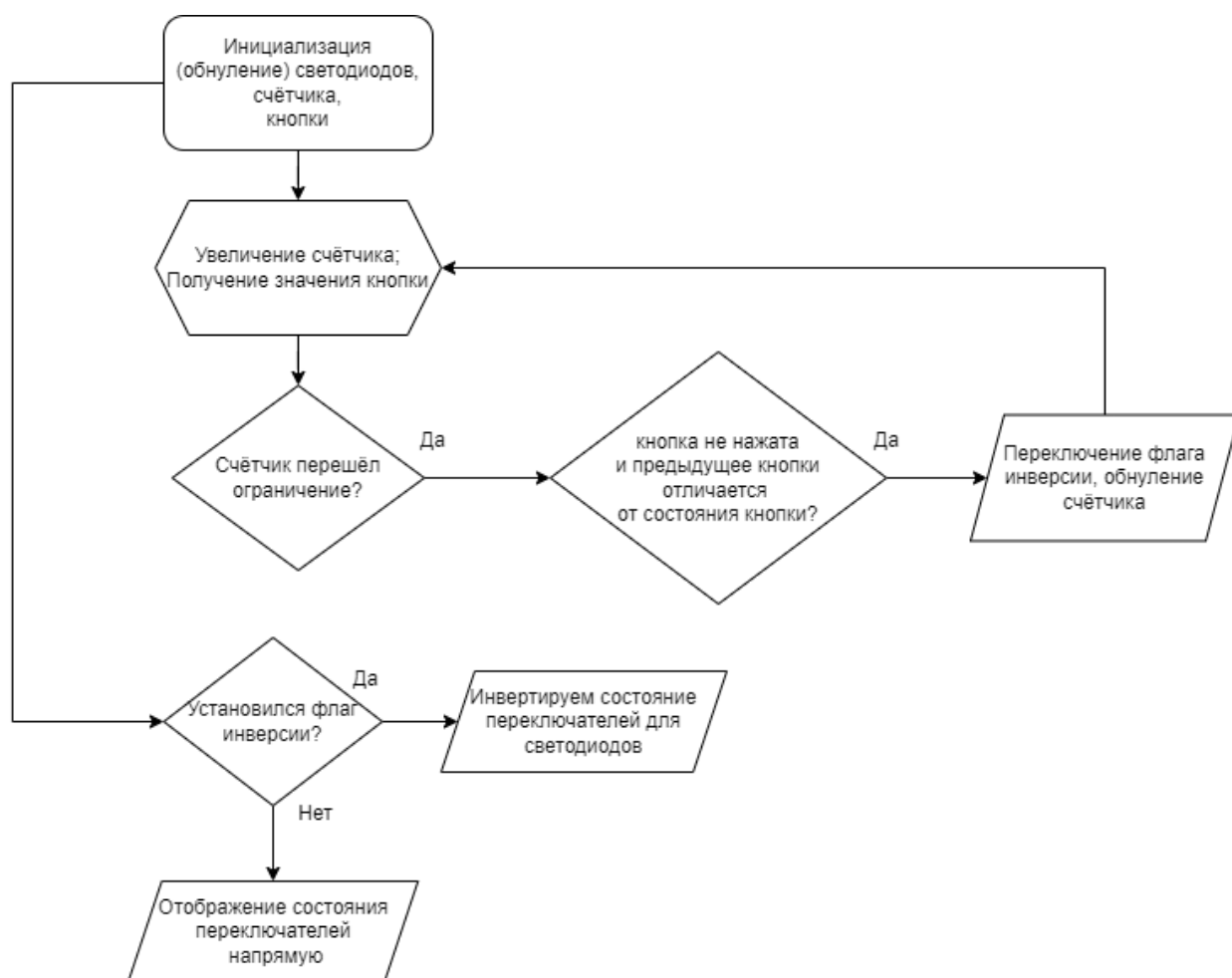


Рисунок 6 – Изменение состояния светодиодов после нажатия кнопки

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
    состояние
    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
```