

Федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский государственный
электротехнический университет «ЛЭТИ»
им. В.И. Ульянова (Ленина)»
Кафедра Информационной безопасности

ОТЧЁТ
по лабораторной работе №4
по дисциплине «Электроника и схемотехника»
Тема: «Использование семисегментного индикатора»

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 четырёхразрядного семисегментного индикатора модели 4042BS-1, изучить процесс управления выводом символов с помощью анодов и катодов, разработать модуль-драйвер для работы с индикатором, провести его симуляцию в среде GTK Wave и протестировать вывод заданных значений на индикатор с использованием платы Tang Nano 9K.

Задание

Придумать выводимое на индикатор значение (статичное или динамическое), осуществить симуляцию модуля-драйвера индикатора, подключить индикатор к плате и отладить его работу.

Используемые аппаратные и программные средства

- Плата Tang Nano 9K
- Четырёхразрядный семисегментный индикатор 4042BS-1
- Среда разработки Gowin EDA
- Среда для симуляции GTKWave
- Язык описания оборудования Verilog

ХОД РАБОТЫ

1.1 Реализация модуля

Модуль в качестве входа принимает тактирующий сигнал, а в качестве выходов аноды (знаки) и катоды (сегменты) индикатора.

При старте модуля инициализируется массив, хранящий состояния сегментов для вывода на индикатор цифр десятиричной системы счисления. Также объявляются и инициализируются счётчик задержки переключения между знаками (delay) индикатора и счётчик (digit index), хранящий номер активного знака.

Далее на повышении тактирующего сигнала в случае, если счётчик задержки не равен нулю, он декрементируется. В противном случае инкрементируется счётчик (digit index), хранящий номер активного знака. Так происходит смена текущего знака при выводе.

Каждый раз при изменении digit index в переменную active digit записываются состояния сегментов из массива цифр в соответствии с тем, какую цифру необходимо вывести на соответствующий разряд индикатора.

Полный исходный код модуля приведен в приложении 1.

1.2 Реализация тестбенча

В тестбенче создается экземпляр модуля indicator. Затем производится генерация тактирующего сигнала с частотой 100 МГц и симуляция модуля на протяжении 1000 наносекунд. Результат симуляции сохраняется в файле indicator_tb.vcd.

Полный исходный код тестбенча приведен в приложении 2.

1.3 Симуляция

На рисунке 2 отражена симуляция текущей программы, которая выводит на индикатор значения 3363. А на рисунке 1 отражена симуляция программы, которая выводит на индикатор значения 1234. Вторая симуляция предназначена для лучшего понимания работы программы.

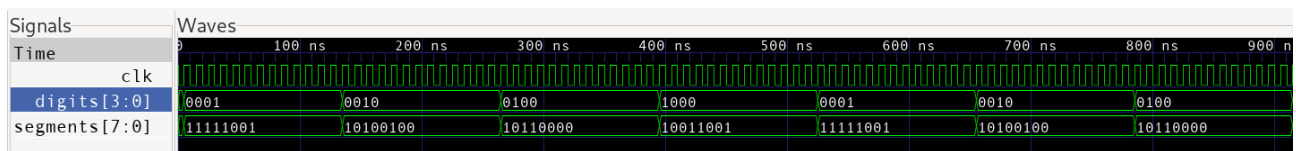


Рисунок 1 — Симуляция GTK Wave для вывода 1234

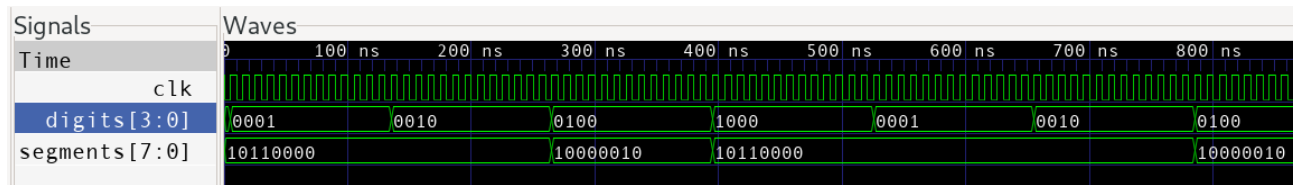


Рисунок 2 — Симуляция GTK Wave для вывода 3363

1.4 Разводка

При разводке модуля для анодов индикатора были выбраны пины с 72 по 75 из 1 банка, так как данные выходы имеют напряжение 3.3 вольта, что необходимо для повышенной яркости сегментов индикатора. Данные пины располагаются на плате последовательно (рисунок 3), что упрощает сборку схемы.

Для катодов были выбраны пины с 79 по 86 из 3 банка. Выбор обусловлен тем, что они располагаются на плате последовательно (рисунок 3), что упрощает сборку схемы, и для данных выводов напряжение не имеет значения.

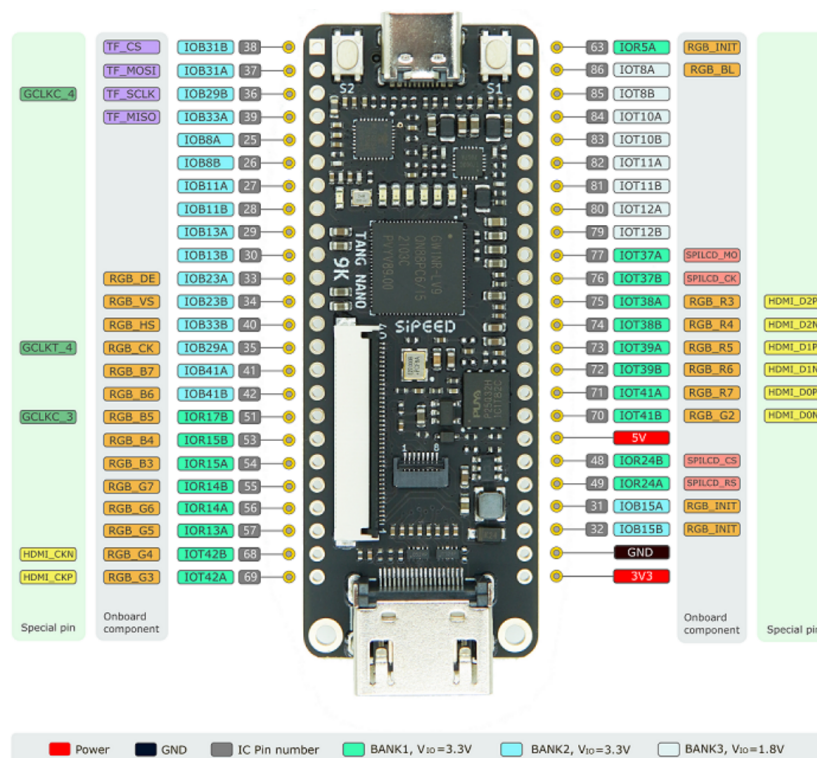


Рисунок 3 — Пины платы

Тактирующий сигнал находится на 52 пине в соответствии с документацией к плате.

Полная разводка модуля приведена на рисунке 4.

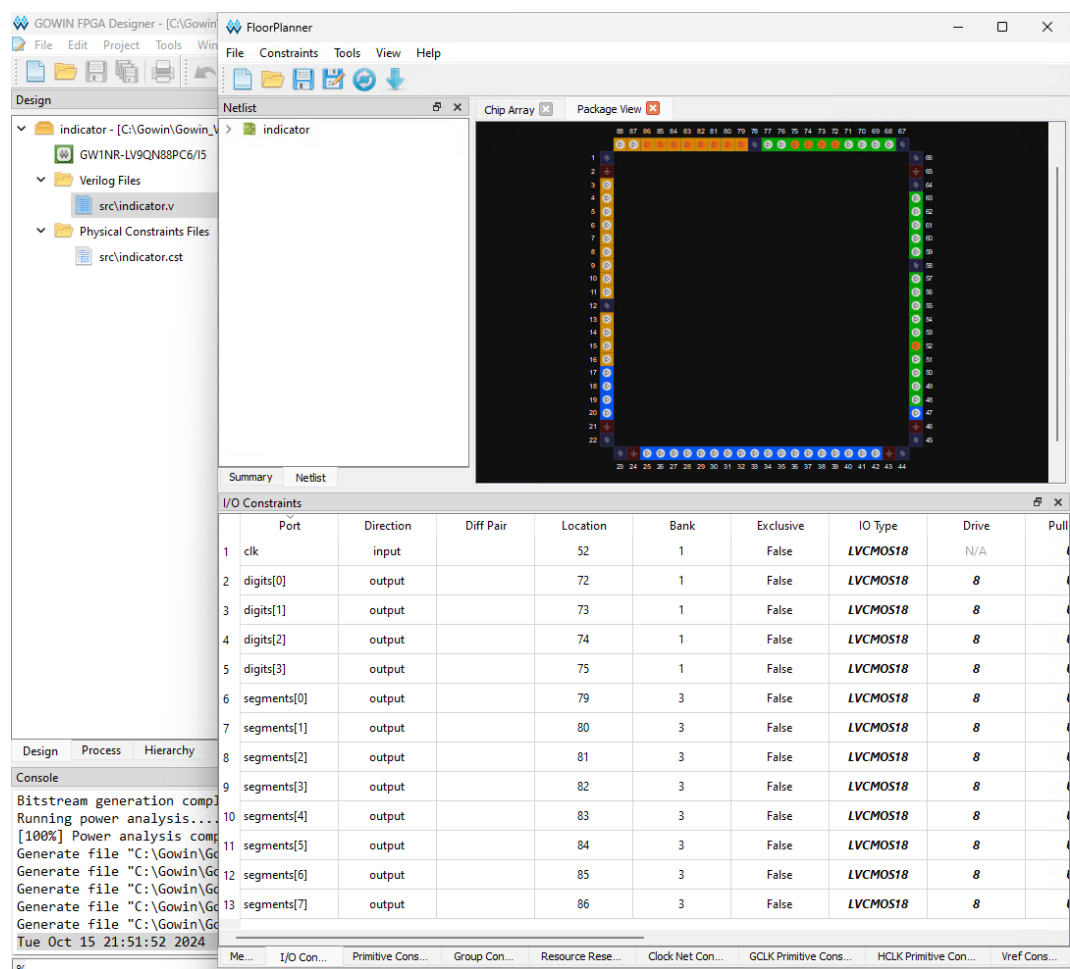


Рисунок 4 — Разводка модуля

1.5 Запуск на плате

Вывод

В ходе работы был разработан модуль-драйвер для управления четырёхразрядным семисегментным индикатором модели 4042BS-1 на базе платы Tang Nano 9K. Модуль корректно реализует циклическое переключение разрядов и вывод соответствующих символов за счёт управления анодами и катодами. Программа использует счётчик для смены активного разряда и выводит данные на сегменты индикатора.

В процессе работы были изучены принципы управления многозначными семисегментными индикаторами и особенности циклического обновления разрядов. Также был получен опыт симуляции и тестирования работы разработанного модуля в среде Gowin EDA.

ПРИЛОЖЕНИЕ 1 — ИСХОДНЫЙ КОД МОДУЛЯ

Файл indicator.v:

```
module indicator
(
    input clk,
    output [3:0] digits,
    output [7:0] segments
);

reg [7:0] DIGITS [0:10];

initial begin
    DIGITS[0] = 'b11000000; // 0: ABCDEF
    DIGITS[1] = 'b11111001; // 1: BC
    DIGITS[2] = 'b10100100; // 2: ABDEG
    DIGITS[3] = 'b10110000; // 3: ABCDG
    DIGITS[4] = 'b10011001; // 4: BCFG
    DIGITS[5] = 'b10010010; // 5: ACDFG
    DIGITS[6] = 'b10000010; // 6: ACDEFG
    DIGITS[7] = 'b11111000; // 7: ABC
    DIGITS[8] = 'b10000000; // 8: ABCDEFG
    DIGITS[9] = 'b10010000; // 9: ABCDFG
    DIGITS[10] = 8'b0; // ALL
end

localparam DELAY = 12;

reg [3:0] delay;
reg [1:0] digit_index;
reg [7:0] active_segment;

initial begin
    delay = 0;
    digit_index = 3;
end

always @(posedge clk) begin
    if (!delay) begin
        delay <= DELAY;
        digit_index <= digit_index + 1'b1;
    end
end
```

```

        end
        else delay <= delay - 1'b1;
end

always @(*) begin
    case (digit_index)
        0: active_segment = DIGITS[3];
        1: active_segment = DIGITS[3];
        2: active_segment = DIGITS[6];
        3: active_segment = DIGITS[3];
        default: active_segment = DIGITS[10];
    endcase
end

assign digits = 4'b0001 << digit_index;
assign segments = active_segment;

endmodule

```


ПРИЛОЖЕНИЕ 2 — ИСХОДНЫЙ КОД ТЕСТБЕНЧА

Файл indicator_tb.v:

```
`timescale 1ns / 1ns

module indicator_tb();

reg clk;
wire [3:0] digits;
wire [7:0] segments;

indicator uut(.clk(clk), .digits(digits), .segments(segments));

// Генерация тактового сигнала с периодом 10 нс (частота 100 МГц)
initial begin
    clk = 0;
    forever #(5) clk = ~clk;
end

// Завершение симуляции через 1000 нс
initial #1000 $finish;

// Сохранение результата симуляции в файл VCD
initial begin
    $dumpfile("indicator_tb_out.vcd");
    $dumpvars(0, indicator_tb);
end

endmodule
```