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

АННОТАЦИЯ

Данная работа включает в себя 5 рисунка и 5 листинга. Количество страниц в работе — 13.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В данной практической работе требуется создать проект САПР Vivado, создать модули, описывающий синхронизатор, параметрический фильтр дребезга, создать модуль верхнего уровня и тестовый модуль на языке Verilog HDL. Протестировать работу модулей и убедиться в их правильности. Затем подготовить отчёт о проделанной работе и защитить её.

1 ПРОЕКТИРОВАНИЕ МОДУЛЕЙ

1.1 Создание модуля Синхронизатора

Синхронизатор — модуль, необходимый для выравнивания частоты. Модуль синхронизатора имеет 2 входа для входного значения in и тактового генератора clk, а также 1 выход для выходного сигнала out. Модуль состоит из двух последовательно соединённых регистров a, принимающий на свой вход входное значение in, и b, принимающий на вход выход регистра a и подающий значение на выход out. С каждым приходом восходящего фронта регистр b запоминает значение регистра a и передаёт его на выход out, а регистр a запоминает новое входное значение (Листинг 1.1).

Листинг 1.1 — Модуль Синхронизатора

```
`timescale 1ns / 1ps


module synhro (
    input clk, in,
    output out
    );

reg a, b;

always@(posedge clk)
begin
    b <= a;
    a <= in;
end

assign out = b;

endmodule
```

На основе написанного модуля САПР Vivado генерирует RTL-схему, представленную на Рисунке 1.1.

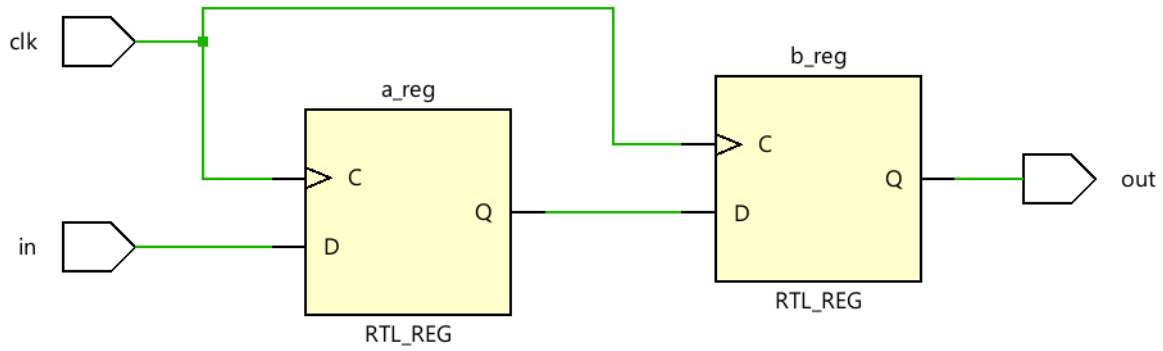

Рисунок 1.1 – RTL-схема модуля Синхронизатора

1.2 Создание модуля Счётчика

Для реализации фильтра дребезга необходимо создать модуль счётчика. Он был реализован в Практической работе №6. Каждый раз, когда приходит восходящий фронт, на выход подаётся остаток от деления суммы текущего состояния и значения параметра step на значение параметра modul. В данной работе модуль был модернизирован. У модуля Счётчика появилось 2 новых входа: разрешающий вход ce и вход сброса re. Теперь значение счётчика изменяется только когда подаётся разрешающее значение, а если будет подана 1 на вход сброса, то счётчик примет 0 значение (Листинг 1.2).

Листинг 1.2 — Модуль Счётчика

```
`timescale 1ns / 1ps

module counter #(step = 1, modul = 2)(
        input clk, re, ce,
        output reg [$clog2(modul) - 1:0] cnt
    );

initial
    cnt = 0;

always@(posedge clk)
    if (re)
        cnt = 0;
    else if (ce)
        begin
            if(step < 0 && cnt == 0 )
                cnt = modul - 1;
            else
```

Продолжение Листинга 1.2

```
                cnt <= (cnt + step) % modul;
        end


endmodule
```

На основе написанного модуля САПР Vivado генерирует RTL-схему, представленную на Рисунке 1.2.

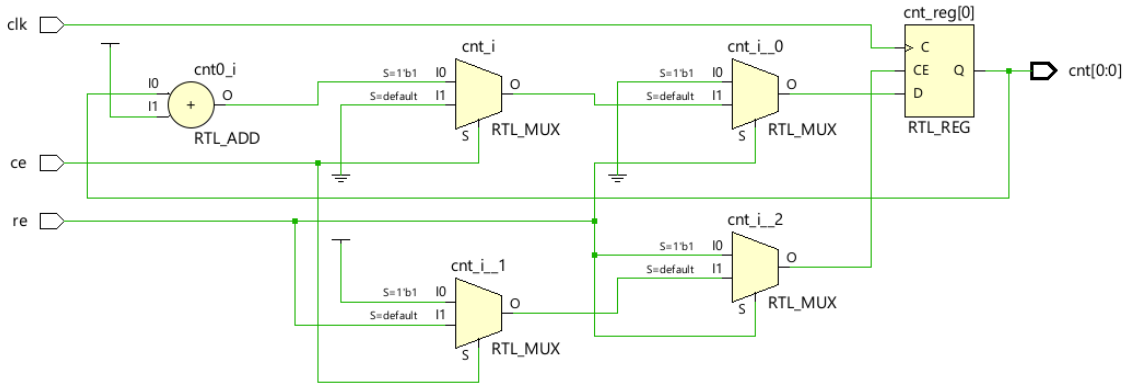

Рисунок 1.2 – RTL-схема модуля Счётчика

1.3 Создание модуля Фильтра дребезга

Фильтр дребезга, предназначен для того, чтобы фильтровать дребезг, возникающий при зажимания контакта, и передавать сигнал только после полного зажатия контакта. Модуль является параметрический и принимает параметр enable, отвечающий за разрешающий вход Фильтра дребезга. Также модуль имеет 2 входа для входного значения in и тактового генератора clk и 2 выхода out для выходного значения и outen для выходного разрешающего значения.

Модуль Фильтра дребезга подключён к 2 модулям: модуль Синхронизатора и модуль Счётчика. Модуль Счётчика считает количество тактов, которые пришли, когда разрешающий вход был равен 1. Модуль Синхронизатора выполняет свой функционал и подаёт свой результат на регистр out_reg с разрешающим входом, куда приходит функция И со всеми битами выходного значения счётчика, который запоминает значение, которое будет подано на выход

out. На выход outen подаётся результат функции И со всеми битами выходного значения счётчика, разрешающим входом и выходом модуля Синхронизатора (Листинг 1.3).

Листинг 1.3 — Модуль Фильтра дребезга

```
`timescale 1ns / 1ps

module filter #(enable = 1)(
    input clk, in,
    output reg out, outen
    );

wire outsynhro;
wire [2 - 1:0] cnt;

synhro sch(
    .clk(clk),
    .in(in),
    .out(outsynhro)
);
counter #(.step(1), .modul(4)) c(
    .clk(clk),
    .re(out~^outsynhro),
    .ce(enable),
    .cnt(cnt)
);

always@(posedge clk)
    if(cnt[1] & cnt[0] & enable)
        out <= outsynhro;

always@(posedge clk)
    outen <= cnt[1] & cnt[0] & enable & outsynhro;


endmodule
```

На основе написанного модуля САПР Vivado генерирует RTL-схему, представленную на Рисунке 1.3.

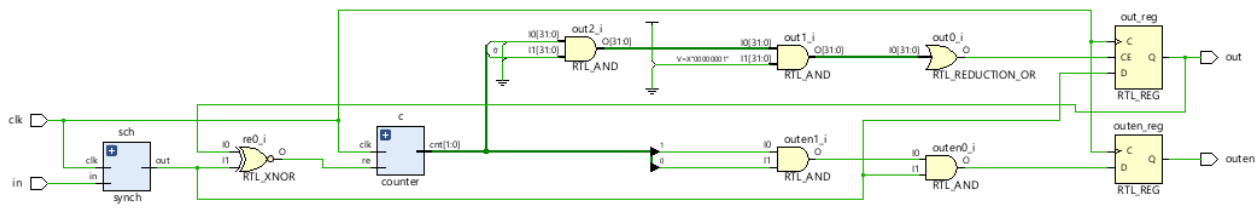

Рисунок 1.3 – RTL-схема модуля Фильтра дребезга

1.4 Создание модуля Верхнего уровня

Данный модуль в данной работе необходим для изменения состояния лампочки. Модуль имеет 2 входа для тактового генератора clk и входа кнопки button_in и 1 выход led_out. К данному модулю подключён модуль фильтра дребезга, в параметр enable которого подаётся значение 1, на вход clk тактовый генератор, на вход in значение входа button_in, а выход outen передаёт значение на регистр out_signal_enable. Каждый раз, как регистр out_signal_enable переходит в состояние 1 значение выхода led_out меняется на противоположное (Листинг 1.4).

Листинг 1.4 — Модуль Верхнего уровня

```
`timescale 1ns / 1ps

module ledcontr (
    input button_in, clk,
    output reg led_out = 0
    );

wire out_signal_enable;

suppression #(.enable(1'b1)) sp(
    .clk(clk),
    .in(button_in),
    .out(),
    .outen(out_signal_enable)
);

always@(posedge out_signal_enable)
    led_out = ~led_out;

endmodule
```

На основе написанного модуля САПР Vivado генерирует RTL-схему, представленную на Рисунке 1.4.

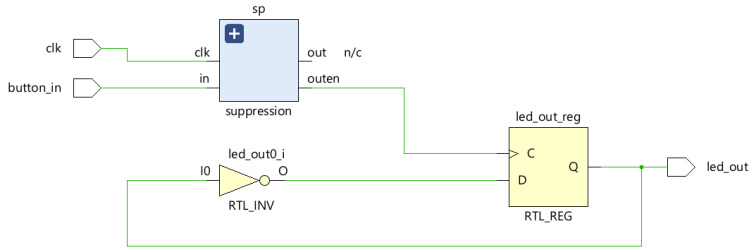

Рисунок 1.4 – RTL-схема модуля Верхнего уровня

1.5 Создание тестового модуля

Для запуска и проверки работоспособности модулей нужен симулирующий тестовый модуль. В него будет подключён модуль Верхнего уровня. Тестовый модуль будет иметь цепь led, отвечающий за лампочку, тактовый генератор и регистр кнопки. На регистр кнопки будут подаваться случайные значения, созданные при помощи функций $srandom, $urandom_range и $random (Листинг 1.3).

Листинг 1.3 — Тестовый модуль testbench.v

```
`timescale 1ns / 1ps

module testbench();

wire led;
reg clk = 0;
reg button = 0;

led_controller contlr(.button_in(button), .clk(clk), .led_out(led));

initial
begin
    #50
    $srandom(33985);
    repeat(2)
    begin
        repeat($urandom_range(50,0))
        begin
            button = $random;
            #3;
        end
        button = 1;
        #200;

        repeat($urandom_range(50,0))
        begin
            button = $random;
            #3;
        end
        button = 0;
        #200;
    end
end

always #10 clk = ~clk;

endmodule
```

2 ВРЕМЕННЫЕ ДИАГРАММЫ

На временной диаграмме будут показаны изменения значений clk — тактовый генератор, button — кнопки и led — лампочки (Рисунок 2.1).

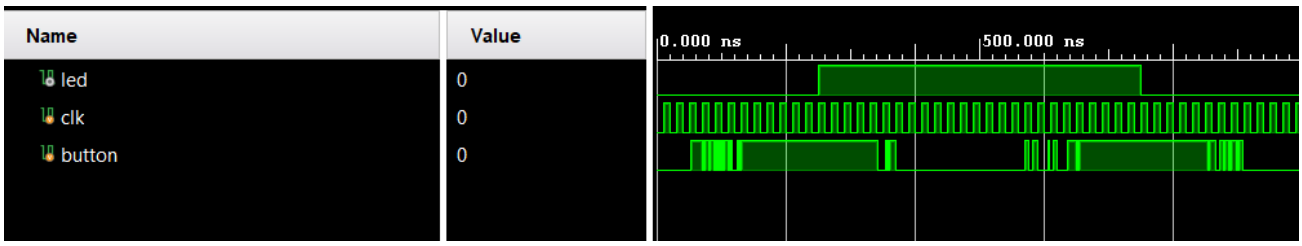

Рисунок 2.1 – Временная диаграмма

На временной диаграмме можно заметить, что значение led меняется не каждый раз, когда button принимает значение 1, а только после того, как кнопка была полностью нажата и фильтр дребезга убедился, что кнопка действительно нажата.

ЗАКЛЮЧЕНИЕ

Для заданной варианта в САПР Vivado был создан модули, описывающий синхронизатор, параметрический фильтр дребезга, создать модуль верхнего уровня и тестовый модуль на языке Verilog HDL. Осуществлена симуляция работы данных модулей, а также проведена успешная проверка работы модулей. Для каждого модуля была сгенерирована его RTL-схема.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методические указания по ПР № 1 — URL: https://onlineedu.mirea.ru/mod/resource/view.php?id=405132 (Дата обращения: 10.12.2023).

2. Методические указания по ПР № 2 — URL: https://onlineedu.mirea.ru/mod/resource/view.php?id=409130 (Дата обращения: 10.12.2023).

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).

4. Тарасов И.Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования. — М.: Горячая линия — Телеком, 2021. — 538 с.: ил.

5. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие / Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).