

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

Институт информатики и кибернетики

Кафедра радиотехники

Отчет по лабораторной работе

"РАЗРАБОТКА ЦИФРОВЫХ УСТРОЙСТВ НА БАЗЕ ПЛИС"

Студент: Согонов Е.А.

Преподаватель: Корнилин Д.В.

Группа: 6364-120304D

Самара 2022

СОДЕРЖАНИЕ

1	Часть I	2
1.1	Создание проекта и описание устройства с помощью VHDL	2
1.2	Симуляция устройства	2
1.3	Синтез цифрового устройства	4
1.3.1	Временные ограничения	4
1.4	РЕАЛИЗАЦИЯ	5
1.5	Индивидуальные задания	5
	ОТВЕТЫ НА ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ	6

1. Часть I

1.1. Создание проекта и описание устройства с помощью VHDL

По методике, описанной в методических указаниях к лабораторной работе, в программном пакете Vivado v2016.4 был создан проект.

Для описания портов и логики схемы используется файл с описанием устройства на языке VHDL *fdc.vhd*, содержимое которого можно увидеть ниже:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL; --подключили библиотеку для использования типа std_logic

entity fdc is --Декларация entity
    Port (
        D_0, D_1: in STD_LOGIC; --описание портов, в данном случае типа in
        C : in STD_LOGIC; --описание портов, в данном случае типа in
        Q : out STD_LOGIC);--описание портов, в данном случае типа out
end fdc;

-- архитектурное тело
architecture Behavioral of fdc is --Архитектура Behavioral для интерфейса fdc
    signal x_0, x_1 : STD_LOGIC; --внутренние сигналы x_0 и x_1, это нужно для того,
    --чтобы можно было во время отладки посмотреть состояние и внутренних сигналов тоже

begin
    process (C) --процесс в неявном виде
    begin
        if rising_edge(C) then
            X_0 <= D_0; --присвоение значений сигналам
            X_1 <= D_1;
        end if;
    end process;

    process (C) --процесс в явном виде, в списке чувствительности только C (то есть блок работает только при изменении
    --в списке чувствительности находится перечисление сигналов, которые активируют процесс)
    begin
        if rising_edge(C) then -- проверка прихода переднего фронта сигнала C
            Q <= X_0 and X_1; -- это выполняется, когда приходит передний фронт (rising_edge) сигнала C
        end if;
    end process;

end Behavioral;
```

Данный код описывает в одном модуле всю схему, представленную на рисунке 1. Первый блок *process (C)* описывает логику D-триггеров DD1 и DD2. Второй блок *process (C)* – логику конъюнктора DD3 и D-триггера DD4.

1.2. Симуляция устройства

Следующий этап - симуляция устройства. Для этого был создан файл симуляции (Test bench) *tb_sim.vhd* на языке VHDL, содержимое которого можно увидеть ниже:

```

--тестовый файл
--подключение библиотек
library IEEE;
use IEEE.STD_LOGIC_1164.ALL; --подключение пакета std_logic_1164 из библиотеки ieee для
--использования типа данных std_logic

entity tb_sim is
end tb_sim;

architecture Behavioral of tb_sim is

COMPONENT fdc -- декларация компонента для UUT
    PORT (
        C : in STD_LOGIC;
        D_0 : in STD_LOGIC;
        D_1 : in STD_LOGIC;
        Q : out STD_LOGIC);
END COMPONENT;
signal D_0 : std_logic := '0'; --декларация сигналов и присвоение им значения "0"
signal D_1 : std_logic := '0';
signal C : std_logic := '0';
signal Q : std_logic := '0';

begin --внутри архитектуры
    uut: fdc --конкретизация компонента для Unit Under Test (UUT)
    PORT MAP ( --подключение выводов
        C => C,
        D_0 => D_0,
        D_1 => D_1,
        Q => Q);

    clock: process --создание тактового сигнала(процесс без списка чувствительности)
    begin
        C <= '0'; wait for 5 ns;
        C <= '1'; wait for 5 ns;
    end process;

    tb : process --процесс тоже без списка чувствительности, используем wait
    begin
        D_0 <= '0'; D_1 <= '0'; wait for 50 ns;
        D_0 <= '1'; wait for 20 ns;
        D_1 <= '1'; wait for 80 ns;
        D_1 <= '0'; wait for 40 ns;
        D_1 <= '0'; wait for 40 ns;
        D_0 <= '1'; wait for 12 ns;
        D_1 <= '1'; wait for 17 ns;
        D_0 <= '0'; D_1 <= '0'; wait for 50 ns;
    end process;

end Behavioral;

```

После запуска симуляции были получены временные диаграммы, изображенные на рисунке 1.

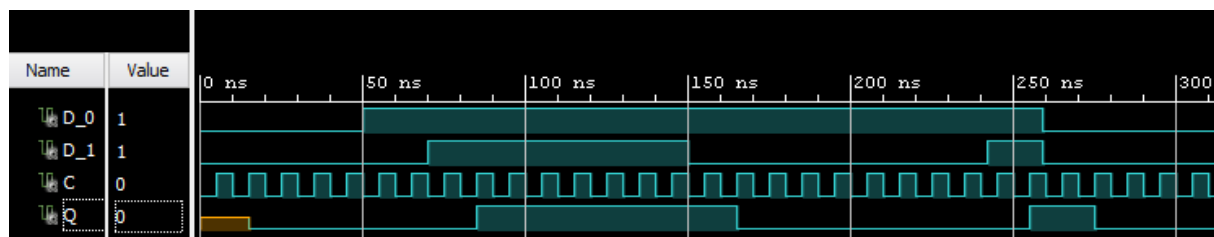


Рисунок 1 — Результат симуляции

1.3. Синтез цифрового устройства

В этом этапе необходимо было провести тщательный временной анализ, необходимый для обнаружения узких мест и обеспечения эффективной и надежной работы устройства. В результате синтеза была получена следующая схема (рисунок 2)

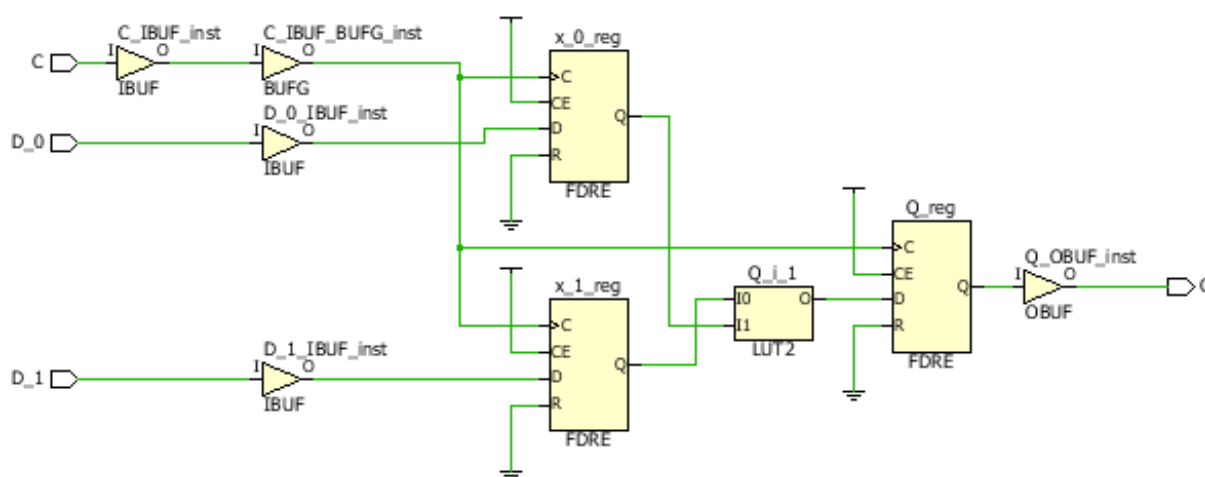


Рисунок 2 — Синтезированная схема

От принципиальной схемы, изображенной в методических указаниях в начале работы (рисунок 3), полученная в результате синтеза схема отличается наличием буферных каскадов IBUF и OBUF (вероятно, это сокращения от input и output buffer), по всей видимости необходимых для согласования напряжений внутри и снаружи микросхемы, согласования сопротивлений. Так же отличием можно считать элемент BUFG, который нужен, как я понял, для того, чтобы использовать глобальный тактовый сигнал.

Так же каждый из D-триггеров на синтезированной схеме имеет неиспользуемый заземленный вход CE (clock enabled)

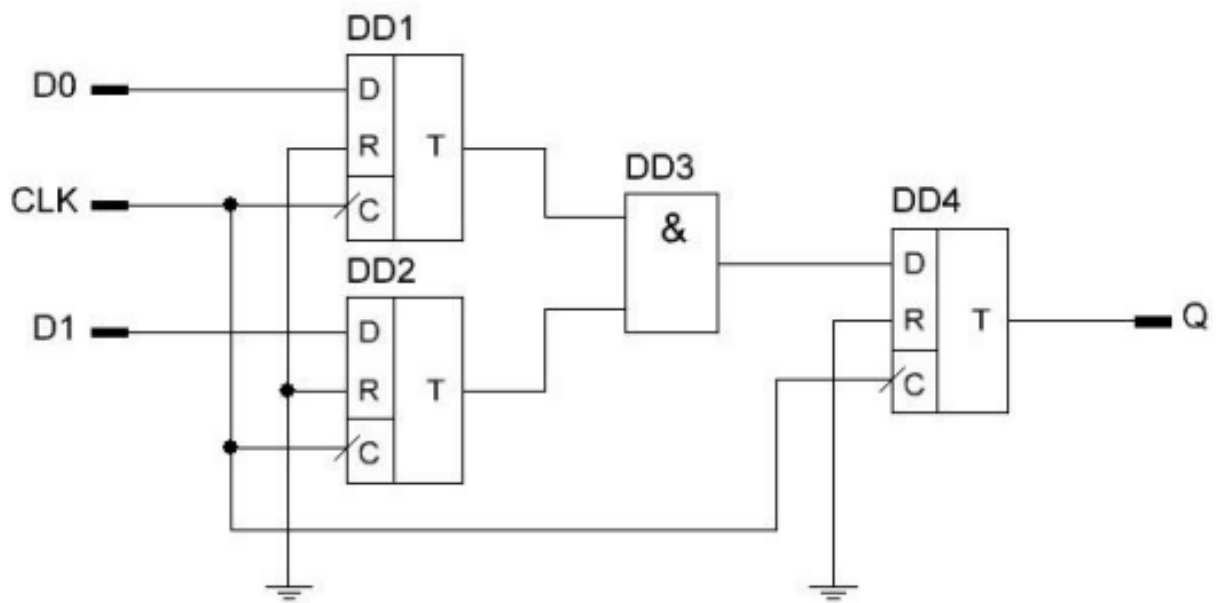


Рисунок 3 — Принципиальная схема из методических указаний к лабораторной работе

1.3.1. Временные ограничения

Тут по моему скромному мнению нужно дохуя почитать харриса или глянуть лекции, потому что я лично понимаю нихуя и только на пальцах в результате все это говно надо связать с рисунком 4

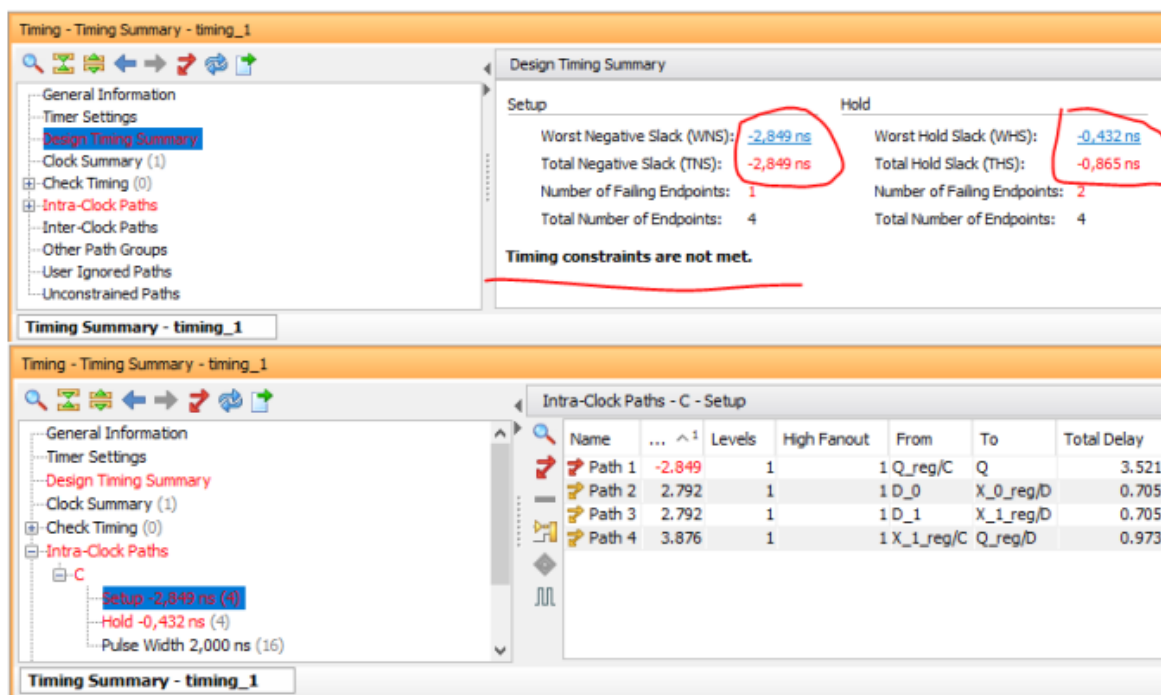


Рисунок 4 — хуевременные задержки

1.4. РЕАЛИЗАЦИЯ

Про автоматическую расстановку портов что то нахуй, или еще что. можно ебануть ту картинку с морским боем

1.5. Индивидуальные задания

1. Выясните, какая максимальная частота работы схемы возможна при заданных временных ограничениях.

2. Выберите ПЛИС с другими параметрами (опции -2 или -3 в названии) и определите, какая максимальная частота возможна для них.

3. Установите ручную позицию вывода Clock на E3 и заставьте Vivado автоматически разместить все остальные выводы.

4. Вернитесь к этапу моделирования и повторите его используя «Run Post-Synthesis Functional Simulation», «Run Post-Synthesis Timing Simulation», «Run Post-Implementation Functional Simulation» и «Run Post-Implementation Functional Simulation». Сравните полученные результаты с результатами, полученными в разделе 2.

5. Повторите пункт 4, задав другую частоту тактового сигнала.

6. Добавьте к наблюдаемым сигналам внутренние сигналы (выходы элемента «И» и триггеров первой ступени) и посмотрите, как выполняются временные соотношения

ОТВЕТЫ НА ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. Какие два подхода к описанию аппаратуры вы знаете? Какой подход вы использовали в лабораторной работе?
2. Какие языки описания аппаратуры (HDL) вы знаете? Какой язык вы использовали в лабораторной работе?
3. Какой программный пакет и какую ПЛИС вы использовали в лабораторной работе?
4. Опишите стандартный процесс проектирования цифровых устройств.
5. Вспомните таблицу переходов D-триггера и таблицу истинности логического вентиля И.
6. Для чего нужна симуляция устройства на поведенческом уровне?
7. Для чего необходимо задавать временные задержки?
8. На каком этапе проектирования цифровых устройств строится топология проекта и подключаются внешние выводы?
9. Опишите назначение строк кода в файле `fdc.vhd`.
10. Опишите назначение строк кода файла симуляции `tb_sim.vhd`.
11. Нарисуйте временную диаграмму работы вашего устройства и сравните ее с диаграммой, полученной в результате симуляции.
12. Что такое ограничения проектирования и для чего они нужны?
13. Какие временные ограничения вы задавали в вашем проекте?