Система удалённого тестирования HDL кода на FPGA

Автор : Сыщиков Никита Сергеевич

Группа P3131

Принимающий :

Обляшевский Севастьян Александрович

Оглавление

[Общее описание проекта : 3](#_Toc193093335)

[Идея: 3](#_Toc193093336)

[Структура: 3](#_Toc193093337)

[Серверная часть : 3](#_Toc193093338)

[Клиентская часть: 3](#_Toc193093339)

[Предполагаемый технологический стек: 3](#_Toc193093340)

[Описание протокола общение между клиентской частью и серверной : 4](#_Toc193093341)

[Описание системы команд: 5](#_Toc193093342)

[Описание серверной части 6](#_Toc193093343)

[Сопоставление этапов разработки проекта с лабораторными работами 5-8: 7](#_Toc193093344)

[Лабораторная работа 5: 7](#_Toc193093345)

[Лабораторная работа 6: 7](#_Toc193093346)

[Лабораторная работа 7: 7](#_Toc193093347)

[Лабораторная работа 8: 7](#_Toc193093348)

# Общее описание проекта :

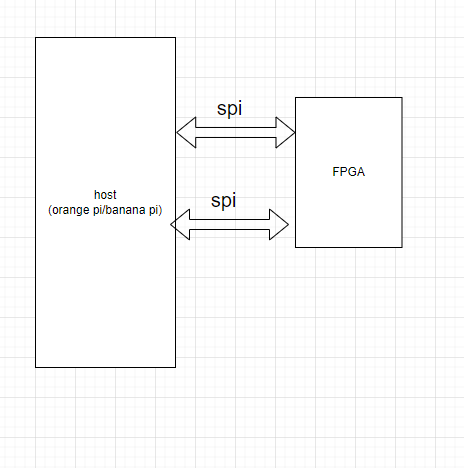
## Идея:

реализовать систему, позволяющую проводить удалённое тестирование hdl кода с использованием FPGA. Пока только на получение данных. В идеале встроить lua для создания тестовых воздействий на тестируемый дизайн и более интеллектуального описания сценария.

## Структура:

### Серверная часть :

Консольное приложение, реализующее такие возможности, как:

* + - * 1. Получение тестового задания от клиентской части по udp, хранение тестовых заданий.
        2. Исполнение тестовых заданий, включающее в себя различные этапы, смотреть рисунок n.
        3. Отправка по запросу клиентской части собранных данных.

### Клиентская часть:

Программа с графическим интерфейсом, реализующая такие возможности, как:

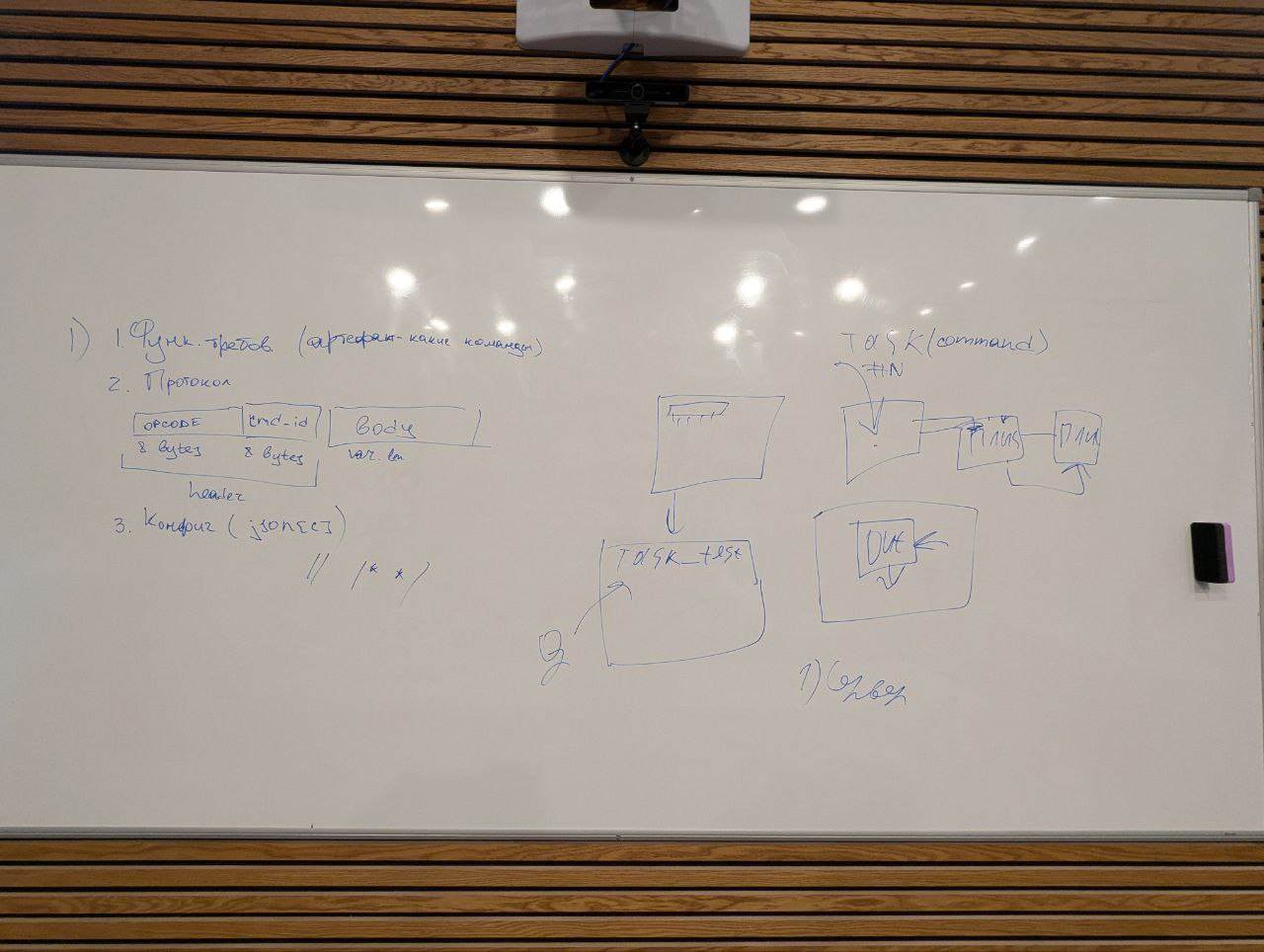
1. Создание тестового сценария, формирование тестового задания и передача его на серверную часть.
2. Управление работой тестового задания на серверной части по udp (постановка задания на исполнение, запуск, остановка, отмена, ограниченная реконфигурация);
3. Получение собранных данных от серверной части по udp, их хранение, показ пользователю.

## Предполагаемый технологический стек:

Серверная часть – C++, FPGA инструментарий

Клиентская часть – C++, qt (Графическая библиотека), altera ide

## Описание протокола общение между клиентской частью и серверной :



Обмен между серверов и клиентским приложение будет происходить по UDP

Будет передаваться прошивка , маска для waveform .

Структура json – тестового сценария

Scenario name: string

Test duration : number - in ms

STOP\_scenario\_pin: number - пин по которому будет останвливаться сценарий(когда он high)

Name of fpga firmware (rbf): string – имя .rbf файла прошивки FPGA

Name of data file: string – имя файла, куда будут собираться данные собранные с fpga

Pin scan Enable : [64] - массив из 64 элементов, где каждый элемент содержить true или false , обозначающий будет ли данный “пин” читаться при сборе waveform.

Pin naming : [{“bus\_name”:string, “pins”: [], “isLSB”:bool}] - группировка пинов в шины , которые могуть быть как lsb так и msb.

## Описание команд:

Help - помощь по командам.

Fire <Имя сценария > - прошивка FPGA  
Start <Имя сценария> - запуск тестового сценария

List scenarios – вывод списка сценариев, доступных на сервере

List firmware – вывод всех прошивок

Print <Имя сценария> - вывод информации о сценарии

Show datafile <Имя файла с данными, собранными с fpga> -вывод определённого файла , полученного после тестирования, где хранятся собранные значения с fpga.

List datafiles - вывод списка всех доступных файлов с данными от fpga.

Change firmware <Имя сценария> <имя новой прошивки> - изменяет прошивку в сценарии.

Pin2busconcat [-lsb|msb] <Имя сценария> <имя шины> {перечисление пинов через запятую} – конкатенация пинов(может быть и один) в одну шину .

Clean buses <Имя сценария> - очистка информации о всех шинах в сценарии.

# Описание серверной части

Конечному пользователю будет предоставлена прошивка – обёртка

https://github.com/LeMaker/WiringBP

## Сопоставление этапов разработки проекта с лабораторными работами 5-8:

## Лабораторная работа 5:

Тема лабораторной работы : работа с коллекциями , с потоками ввода-вывода , параметризация классов.

Что сделаю : основную логику работы, прошивки , сбора данных , обработки команд .

## Лабораторная работа 6:

Тема лабораторной работы : клиент – серверная архитектура, протоколы , сокеты, tcp/udp , передача данных по сети , сериализация и десериализация полей и методов.

Что сделаю : клиентское приложение, которое подключается к серверу , может кидать ему команды и отправлять json сценарии , прошивки и получать файл с данными от плис.

## Лабораторная работа 7:

Тема лабораторной работы : многопоточность , использование базы данных.

Что сделаю : в базу данных кидать сценарии , их состоянии (хз зачем). В многопотоке можно реализовать одновременную обработку сценария(тяжеловесная штуа) и обработку запросов от клиента/клиентов. Два клиента могут ставить свои задачи в очередь исполнения, но одновременно выполняется только один сценарий (лочится).

## Лабораторная работа 8:

Тема лабораторной работы : Графическая оболочка

Что сделаю : напишу QT интерфейс для клиента