

# Мини-отчёт

Рудаков Максим, М3137

11 ноября 2023 г.

<https://github.com/skkv-itmo2/itmo-comp-arch-2023-circuit-Massering>

## 1 Логический уровень

Я решил начать именно с Logisim'a, чему впоследствии был очень рад. Модули организованы по смыслу. Я бы выделил 3 главных модуля: память MEMORY (хранит 5 ячеек по 4 бита), голова HEAD (хранит положение +1 от вершины стека), а сам стек stack отвечает за логику происходящего. Отмечу, что положение вершины стека изменяется по спаду синхронизации, всё остальное работает по высокому уровню. Сделать normal также не составило труда, я использовал подтягивающие резисторы (кажется, работало бы без них) и передаточный вентиль (большое спасибо Вам за презентацию).

## 2 Структурный уровень

В общем это просто переписанный логический уровень. Признаюсь, пока я делал этот уровень, мне пришлось поправлять логический, потому что важный для реализации модуль kostyl (16 "НЕ" подряд) оказался непереносимым. Ещё я получил неудовольствие писать некоторые модули (mod\_5), потому что Verilog плохо работает с массивами модулей. Входовыходы работают принципиально так же, как в Logisim'e, но тут я использовал два транзистора: nmos и pmos.

## 3 Поведенческий уровень

После проделанной работы на предыдущих уровнях я откровенно боялся приступать к этому уровню, но, к моему удивлению, сделать его оказалось просто. Переход индекса вершины стека я попытался сделать так же, по спаду синхронизации, однако какая-то шутка в тестах, которую я не понял: "#0.1 всё мне ломала, так что теперь я не уверен, что behaviour работает по высокому уровню. Стек составляет двумерный массив reg. Вершину стека решил пикануть и хранить в integer. Для входовывода использую отдельный массив reg, который присоединяю к IO\_DATA. Этот массив переходит в высокоимпедансное состояние по спаду синхронизации, чтобы обеспечить доступ к поступающим значениям.