

Лабораторная работа №2

Построение логических схем

Инструментарий и требования к работе:

Работа выполняется в среде моделирования Logisim-evolution и Icarus Verilog 10 и новее (желательно 12).

Ссылки: [logisim-evolution v3.8.0](#) и [iverilog](#) (раздел "Инструменты для работы на Verilog").

Описание работы

Работа состоит из двух частей: моделирования в Logisim и описания схемы на SystemVerilog. В заданиях задана одна и та же схема.

Схема

Собрать схему "Синхронный стек". Размер (число ячеек) – 5. Одна ячейка хранит 4 бита. У стека нет состояния “занятости” (занято/свободно). Синхронизация (CLK) должна работать по высокому уровню.

Схема должна обрабатывать следующие команды (подаются через вход команд COMMAND):

0 – пор (нет операции)

1 – push (положить значение ячейки на вершину стека)

2 – pop (снять значение ячейки с вершины стека)

3 – get (получить значение ячейки по индексу, относительно вершины стека)

Отдельно должен быть вход для инициализации (RESET) – асинхронно переводит схему в начальное состояние и обнуляет состояние внутренней памяти.

Базовый вариант (easy): у схемы должны быть отдельно вход (I_DATA) и выход (O_DATA) данных. Усложненный вариант (normal): у схемы должен быть входо-выход (IO_DATA) данных.

Задание 1. Logisim

Собрать описанную схему на Logisim. В репозитории выданы проекты с прототипом схемы. Модифицировать схемы main и уже размещённые контакты в проекте запрещено.

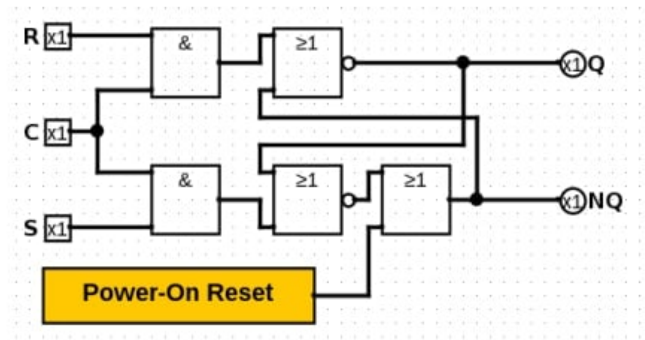
Требуемая схема должна быть реализована на подсхеме *stack*, подсхема *main* должна содержать только один экземпляр подсхемы *stack* и элементы ввода и вывода, для тестирования собранной схемы. Необходимо использовать дополнительные подсхемы (например, подсхема ячейки памяти) для реализации подсхемы *stack*. Файл с проектом (всеми подсхемами) должен называться *stack_logisim_(easy/normal).circ*. Если реализуется базовый вариант (отдельно вход и отдельно выход данных, то нужно дополнять проект с суффиксом "*stack_logisim_easy.circ*", иначе – "*stack_logisim_normal.circ*").

В задании можно использовать только полевые транзисторы и следующие логические элементы: NOT, OR, AND, NOR, NAND, XOR, XNOR. Соответственно все триггеры, мультиплексоры и пр. собираются на логических элементах в виде подсхем. Также стоит собрать часто повторяющиеся элементы в отдельные подсхемы.

Направление всех логических элементов: Восток. Исключение – элементы выхода, для них направление Запад. Входы-выходы располагаются снизу. Все входы и выходы подсхем – pin (контакт).

Все входы, выходы и подсхемы должны быть названы (иметь заполненный label) и адекватно называться.

Для тестирования reset полезно использовать элемент POR (в русской локализации сигнал сброса). Сигнал reset, в частности, нужен для багфикса симуляции logisim при сборке триггеров (если проблема ещё возникает).

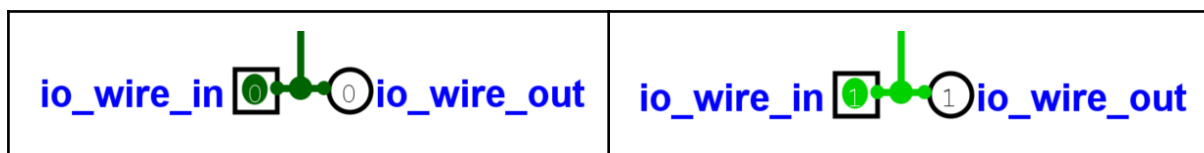


Помимо описанных выше элементов можно использовать разветвители (splitter), датчики (probe), тоннели (tunnel), передаточный вентиль (transmission gate) (он же cmos в verilog).

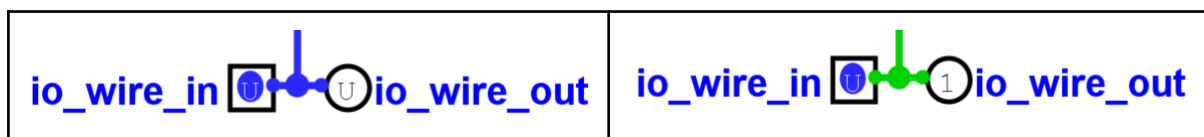
Пояснение про входо-выход

Входо-выходы на схеме можно подключить к pin (контакту) и probe (датчику). Это позволит задавать значение через pin, когда провод используется как вход, и наблюдать значение, когда провод используется как выход.

В случае записи контакт, подсоединённый к inout проводу, ставится в 0 или 1.



В случае чтения данных с провода необходимо установить на контакте высокоимпедансное состояние (значение U).



Задание 2. SystemVerilog

Собрать описанную схему на SystemVerilog.

Задание 2 делится на 2 подзадания. Оба подзадания реализуют описанную ранее схему двумя разными способами. К обоим заданиям написан тестирующий модуль (testbench), при запуске которого будет проводиться тестирование ваших описанных модулей.

В интерфейсе всех ваших модулей сначала объявляются выходы, затем входы-выходы, затем входы.

Если реализуется базовый вариант (отдельно вход и отдельно выход данных, то модули называются с суффиксом "_easy", иначе – "_normal"). В репозитории выданы прототипы модулей и тестовых модулей. Модифицировать модули (название и интерфейс) в шаблонах запрещено.

Задание 2.1. Описание через структурную модель

Для реализации стека нужно использовать только встроенные примитивы: транзисторы (pmos, nmos, cmos) и логические элементы (not, and, nand, or, nor, xor, xnor).

Модуль, реализующий стек, должен называться *stack_structural_[easy/normal]*, тестирующий модуль – *stack_structural_tb*. Файл, содержащий все модули этого задания должен называться *stack_structural.sv*, файл с *stack_structural_tb* – *stack_structural_tb.sv*.

Задание 2.2. Описание через поведенческую модель

Для реализации стека нужно использовать операторы поведенческого моделирования (always, initial, assign, операторы, ...) и управляющие конструкции (case, if, ...).

Модуль, реализующий стек, должен называться *stack_behaviour_[easy/normal]*, тестирующий модуль – *stack_behaviour_tb*.
Файл, содержащий все модули этого задания должен называться *stack_behaviour.sv*, файл с *stack_behaviour_tb* – *stack_behaviour_tb.sv*.

Полезные материалы по SystemVerilog:

https://books.ifmo.ru/book/2667/arhitektura_evm:_uchebno-metodicheskoe_posobie_po_laboratornym_rabotam..htm

https://mk.cs.msu.ru/images/e/ee/VLSI_Verilog_2017.pdf

https://kpfu.ru/portal/docs/F2032376552/Verilog_Examples_D.pdf

Модификация ППА

Без модификации.

Содержание отчета

1. Минититульник (ФИО, группа и название работы в любом формате).
2. Ссылка на репозиторий.
3. Версия logisim-evolution и iverilog.
4. *Описание работы схем.* Что за схема вам дана, как она работает (примеры работы схем/модулей в виде таблицы истинности / временной диаграммы и подобное).

Отдельно описать особенности сборки на Logisim, отдельно – на SystemVerilog.

При описании работы схемы может быть полезно приложить формулы/таблицы/временные диаграммы или небольшие рисунки для иллюстрации пояснений.

Порядок сдачи работы

1. Выполнить работу.

2. Оформить миниотчет в формате pdf.
3. Загрузить файл миниотчета и файлы со схемами и кодом в выданный вам репозиторий в корень (не надо плодить папки, это неудобно для автотестов). Можно заводить свои дополнительные файлы для описания схем на SystemVerilog, все схемы на Logisim должны находиться в одном Logisim-проекте.
4. Запустить автотесты. Подробнее: [Автотесты на GitHub - comp-arch-course \(gitbook.io\)](https://gitbook.io/comp-arch-course)

В репозиторий необходимо загружать файл с миниотчётом в формате pdf, названным в формате **“Фамилия_Имя_Группа_НомерРаботы.pdf”**. Номер группы – только 2 последние цифры группы, номер работы – порядковый номер лабораторной работы: 1, 2 и т.д.