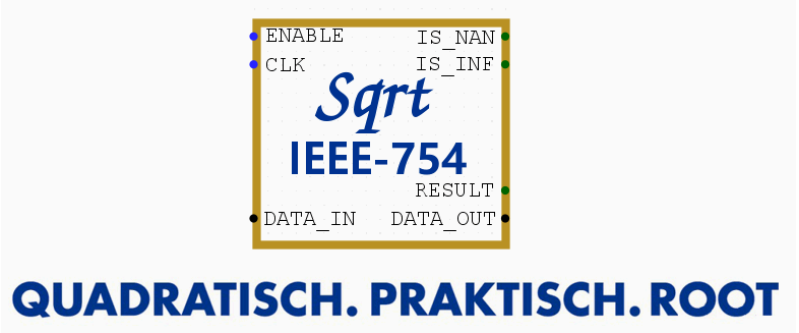
Лабораторная работа №2

Схемы



Инструментарий и требования к работе

ПО	Logisim-evolution 3.9.0 и Icarus Verilog 12
Чтобы работа была принята на проверку, необходимо сделать как минимум одну из реализаций.	

Задание

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

Схема

Собрать схему "Квадратный корень из half precision". На вход по шине данных подаётся число, записанное в 16-ричной побитовой форме. Над ним схема в столбик *итерационно* вычисляет значение квадратного корня и выдаёт ответ на шину данных. Округление к 0.

В случае получения NaN в результате операции с не-NaN, должен получится тихий NaN с остальными битами мантиссы 0 и битом знака 1.

Если в результате получается специальное значение (не-число или бесконечность), то помимо значения, на соответствующий выход должна подаваться 1.

Принцип работы

Вход CLK – вход синхронизации.

ENABLE – вход, отвечающий за начало и продолжение работы. Гарантируется, что исполнение схемы начинается с ENABLE = 0.

На первом такте после перехода ENABLE 0 -> 1 схема берёт с шины данных IO_DATA входное значение и начинает итеративно считать корень.

В процессе вычисления текущее (промежуточное) значение результата должно выводиться на шину данных (со второго такта).

Когда корень досчитался, то выход RESULT должен быть установлен в 1, выходы IS_NAN/IS_PINF/IS_NINF установлены в соответствии с полученным результатом. Это состояние должно сохраняться до сброса.

В случае установки ENABLE = 0 происходит остановка расчётов, сброс внутреннего состояния, ожидание подачи данных на шину данных.

Гарантируется, что при CLK = 1 значение ENABLE и входное значение на шине данных не меняются.

Задание 1. Logisim

Собрать описанную схему на Logisim. В репозитории выдан проект с прототипом схемы, который вы должны использовать. Модифицировать схему *main* и менять порядок уже размещённых контактов на схеме *sqrt* запрещено.

В задании можно использовать только передаточный вентиль (transmission gate), полевые транзисторы (transistor) и базовые логические элементы. Из вспомогательных элементов: разветвители (splitter), датчики (probe), тоннели (tunnel), константы (constant, power, ground). Соответственно, все триггеры, мультиплексоры и пр. собираются

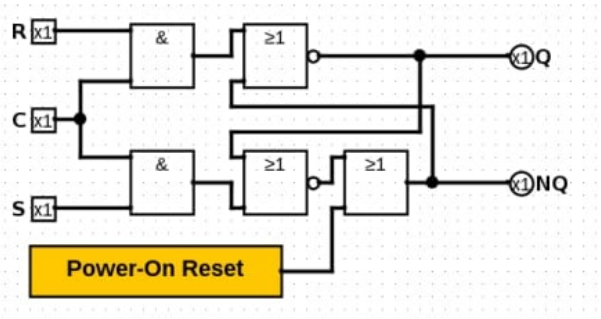
самостоятельно в виде подсхем. Также стоит собрать часто повторяющиеся элементы в отдельные подсхемы.

Направление всех логических элементов: Восток (East). Исключение – элементы НЕ, если иное размещение не мешает читать схему. Входо-выходы располагаются снизу. Все входы и выходы ваших подсхем – контакты (pin).

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

Если возникает проблема с функционированием триггеров, то для багфикса симуляции Logisim следует использовать элемент “сигнал сброса” (POR). Обратите внимание, что в Verilog POR не нужен.

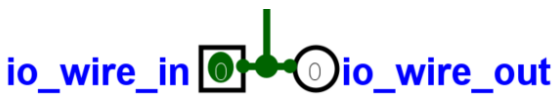
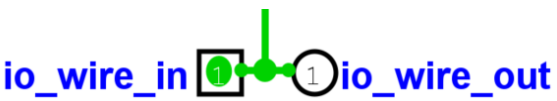
Пример:



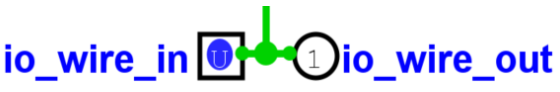
Пояснение про входо-выход

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

В случае записи контакт, подсоединённый с 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 в поведенческой модели**.** Обратите внимание, что поведение схем на Logisim и Verilog должно быть эквивалентно.

Шаблон файла размещён в репозитории.

Для реализации нужно использовать операторы поведенческого моделирования (`always`, `initial`, `assign`, операторы, …) и управляющие конструкции (`case`, `if`, …). Типы данных: `reg`, `wire`. Допустимы функции, события.

В дополнение к реализованному модулю необходимо написать testbench, показывающий, что модуль описан и работает корректно.

В автотестах на GitHub из **корня репозитория**:

build	`iverilog -g2012 -o sqrt2_tb.out sqrt2_tb.sv`
sim	`vvp sqrt2_tb.out`

Репозиторий и работа с шаблонами

Файлы в `template_dont_edit` не модифицируем. Сделайте копию шаблонов в корень репозитория и правите их.

В `sqrt2_tb.sv` приведён код, позволяющий записывать значение на каждом такте в csv файл (возможно, будет полезен).

Logisim тестируется вручную :(

Полезные материалы

- *презентация с лекций, лист Информация*
- *“Справочная информация о Verilog”*

Возможно, будет полезно: https://www.youtube.com/watch?v=U7S0Dkkcrbk