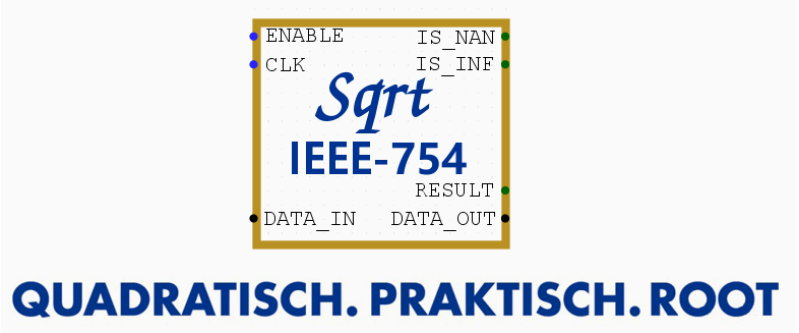
Лабораторная работа №3

Схемы с памятью



Инструментарий и требования к работе

ПО	Icarus Verilog 12
Чтобы работа была принята на проверку, необходимо сделать как минимум одну из реализаций и тестовое окружение.	

Задание

Работа состоит из трёх частей: описания схемы на Verilog в структурной и поведенческой реализациях, описание тестового окружения для проверки работоспособности реализаций. В заданиях задана одна и та же схема. Обратите внимание, что поведение обеих реализаций должно быть внешне одинаково. Шаблоны размещены в репозитории.

Схема

Собрать схему "Квадратный корень из half precision". На вход по шине данных подаётся 16-битное число. Над ним схема в столбик *итерационно* вычисляет значение квадратного корня и выдаёт ответ на шину данных. Итерационно – таким образом, что её несложно масштабировать до, например, single precision без принципиальной модификации кода (реализация в духе СКНФ/СДНФ по таблице истинности принята не будет). 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 синхронизации 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.

Структурная реализация

Реализовать описанную схему на SystemVerilog в структурной модели**.**

В задании можно использовать только базовые логические примитивы (не операторы!) и примитивы транзисторы (**.mos**). Типы

данных: `reg`, `wire`. Из вспомогательного: константы (constant, `supply0`/`supply1`), непрерывное присваивание (`assign`). Соответственно, все триггеры, мультиплексоры и пр. описываются самостоятельно в виде модулей. Также стоит собрать часто повторяющиеся элементы в отдельные модули.

Поведенческая реализация

Реализовать описанную схему на SystemVerilog в поведенческой модели**.**

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

Тестовое окружение

В дополнение к реализованному модулю необходимо написать testbench, показывающий, что модуль описан и работает корректно. Тестирующий модуль не должен знать о внутреннем устройстве тестируемого модуля, например, о том, через сколько тактов будет установлен RESULT = 1, промежуточные значения проверять не следует.

Если все тесты пройдены успешно, то симуляция должна завершиться с кодом 0, иначе – кодом 1:

`$fatal(2, "message")`	Завершение симуляции с кодом 1
`$finish`	Завершение симуляции с кодом 0

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

build (behav.)	`iverilog -g2012 -D BEHAVIOUR -o sqrt2_tb_b.out sqrt2_tb.sv`

sim (behav.)	`vvp sqrt2_tb_b.out`
build (struct.)	`iverilog -g2012 -D STRUCTURAL -o sqrt2_tb_s.out sqrt2_tb.sv`
sim (struct.)	`vvp sqrt2_tb_s.out`

При проверке работы будут запускаться закрытые тесты.

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).

Полезные материалы

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

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