

Специализированный стенд для тестирования операционных устройств TEST_OY

1. Структура стенда TEST_OY

Стенд представляет собой проект с именем TEST_OY, выполненный в САПР QUARTUS II с использованием языка VHDL. Все файлы проекта находятся в одноименной папке. Структура стенда приведена на Рис. 1.

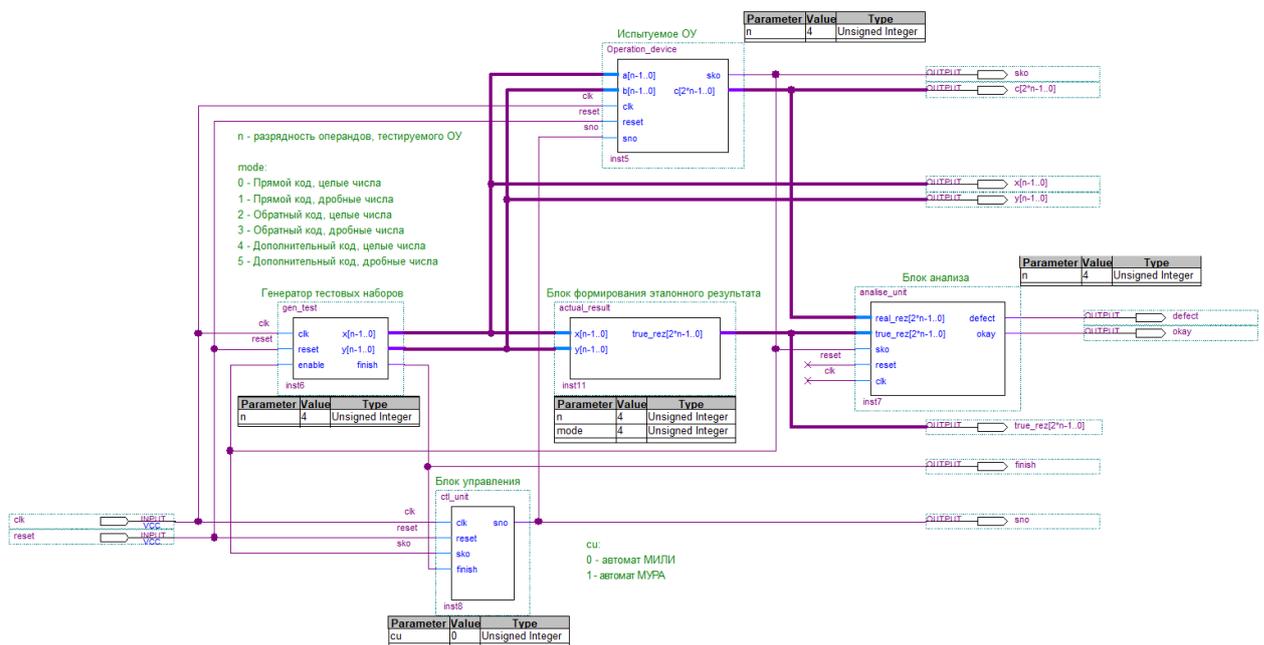


Рис. 1 Структура специализированного стенда TEST_OY

Испытуемое операционное устройство, созданное студентами, входит в состав стенда в виде компоненты **Operation_device**. Для задания разрядности ОУ используется настраиваемый параметр n . Помимо испытуемого устройства в состав стенда входят следующие блоки: генератор тестовых наборов **gen_test**, блок формирования эталонного результата **actual_result**, блок анализа результата **analyse_unit** и блок управления **ctrl_unit**. Для описания структурных частей стенда используются процессы с одноименными названиями.

Генератор тестовых наборов **gen_test** представляет собой двоичный счетчик, который формирует последовательные двоичные наборы, начиная с начального и до конечного. Начальное и конечное значения задаются в виде констант `start_patern` и `stop_patern`, которые легко можно изменить. По умолчанию начальное значение представляет комбинацию из всех нулей. Конечное значение – комбинацию из всех единиц. То есть, по умолчанию генератор перебирает все возможные значения, обеспечивая тем самым исчерпывающее тестирование операционного устройства. Параметр `n` задает разрядность формируемых операндов. После того, как генератор сформирует последний тестовый набор, он выработает сигнал `finish`, который остановит работу блока управления.

Блок **actual_result** формирует эталонный результат `true_rez`, который зависит не только от значений операндов - множимого и множителя, но и от формата и кода, в котором они представлены. Для этого используется параметр `mode`. Варианты задания параметра `mode` приведены на схеме и в файле `TEST_OY.vhd`.

Блок **analyse_unit** сравнивает результат, формируемый испытуемым устройством `real_rez`, с эталонным результатом `true_rez`, вырабатываемым блоком **actual_result**. Анализ выполняется по положительному фронту синхросигнала `clk`, после формирования испытуемым устройством сигнала конца операции `sko`. Если хотя бы один раз результат не совпадет с эталонным, то в этом случае будет сформирован единичный сигнал `defect`, который не изменит своего значения до конца тестирования. Сигнал `окау`, формируется каждый раз после формирования сигнала конца операции. Его единичный уровень свидетельствует о совпадении результата с эталонным.

Для формирования сигнала начала операции `spo` используется модуль **ctl_unit**. Этот модуль формирует единичный сигнал либо асинхронно по сигналу `reset`, либо по положительному фронту синхросигнала `clk` всякий раз после появления на его входе сигнала конца операции до тех пор, пока не

будет сформирован генератором **gen_test** сигнал finish. Тип управляющего автомата МИЛИ или МУРА можно задать при помощи параметра CU.

Все файлы, связанные с тестируемым устройством, находятся в папке MY_OY. Тестовые наборы, включающие рассмотренные в разделе **Ошибка!** **Источник ссылки не найден.** примеры, содержатся в файле test_oy. Наборы дополнены примерами с множителем, равным минус единице, и с множителем, равным нулю. Первый из них требует максимального времени выполнения операции, так все разряды множителя равны единице. Для n-разрядного множителя время выполнения операции составит 2n тактов. Второй пример требует минимального времени выполнения операции – n тактов. Файл test_oy предназначен для автономного тестирования операционного устройства.

Также в папке MY_OY находится файл test_for_stand, предназначенный для исчерпывающего тестирования операционного устройства. В нем активными являются только два сигнала clk и reset. Первый из них используется для тактирования стенда и испытуемого устройства. Второй для установки их в начальное состояние.

2. Принцип работы стенда TEST_OY

Стенд TEST_OY работает следующим образом. По единичному внешнему сигналу reset стенд вместе с тестируемым устройством, установится в начальное состояние. При этом генератор установится в соответствии с параметром start_patern в состояние, соответствующее первому тестовому набору. После снятия сигнала reset блок управления **ctrl_unit** выработает первый сигнал начала операции sno, который запустит операционное устройство, вычисляющее результат для первого тестового набора. После того, как результат будет получен, испытуемое устройство сформирует сигнал конца операции sko. В свою очередь этот сигнал разрешит блоку анализа **analise_unit** сравнить полученный результат с эталонным, блоку **gen_test**

сформировать следующий тестовый набор, а блоку **ctrl_unit** - следующий сигнал **spo**. Причем все эти действия будут выполняться одновременно по положительному фронту синхросигнала **clk**. Аналогичным образом будут формироваться все последующие тестовые наборы и анализироваться полученные результаты до тех пор, пока генератором **gen_test** не будет сформирован сигнал **finish**, означающий завершение тестирования.

3. Подключение испытуемого устройства к стенду

Архитектурное тело стенда содержит один компонент - испытуемое операционное устройство. Все остальные блоки стенда, разобранные в предыдущей части, представлены процессами с одноименными названиями. Это процессы формирования тестовых наборов, управления, анализа и вычисления эталонного результата.

Для подключения операционного устройства к стенду рекомендуется использовать конфигурацию. Студенту лишь необходимо снабдить своё устройство параметром **n**, не задавая ему конкретного значения, и сохранить название портов, совпадающие с названиями портов операционного устройства специализированного стенда **TEST_OU**. Когда все эти мероприятия будут выполнены, необходимо скопировать файлы с описанием своего операционного устройства в папку **MY_OU**, после чего добавить файлы к проекту.

Затем следует создать файл конфигурации, в котором надо указать, что в стенде в качестве компоненты **operation_device** следует использовать устройство, разработанное студентом. Шаблон файла конфигурации содержится в папке с проектом. Файл **config** позволяет задать используемую в процессе моделирования конфигурацию проекта. В нашем случае надо указать, что в качестве компонента **Operation_device** будет использоваться операционное устройство, разработанное студентом. Подставляем нужный **entity**, подставляем нужное архитектурное тело.

Объявляем файл TEST_OY модулем верхнего уровня проекта и выполняем компиляцию проекта. Для корректной работы блока формирования эталонного результата, перед выполнением компиляции проекта необходимо задать параметр mode. Этот параметр зависит от кода и формата, в котором представлены операнды.

4. Тестирование операционного устройства с помощью специализированного стенда TEST_OY

Чтобы выполнить тестирование операционного устройства с помощью стенда TEST_OY, необходимо промоделировать поведение стенда с включенным в его состав испытуемым устройством. Для этого необходимо подать на стенд внешние сигналы reset и clk. Первый сигнал устанавливает стенд и тестируемое устройство в исходное состояние. Второй сигнал используется для тактирования работы стенда и операционного устройства. В папке MY_AVT имеется файл с подготовленными временными диаграммами test_for_stand. После открытия этого файла, запускаем функциональное моделирование, используя соответствующую пиктограмму на панели инструментов.

5. Анализ полученных результатов

После завершения моделирования необходимо проанализировать полученные результаты, представленные на временной диаграмме.

Во-первых, необходимо убедиться, что происходит изменение подаваемых на ОУ тестовых наборов до тех пор, пока не будет сформирован единичный сигнал finish, свидетельствующий о подаче всех тестовых наборов на испытуемое устройство.

Во-вторых, сигнал defect, свидетельствующий об обнаружении несоответствия полученного с испытуемого ОУ результата с эталонным, должен быть равен нулю на всем протяжении временной диаграммы. В противном случае его единичное значение будет указывать о неверном

результате выполнения операции. Проанализировав момент времени формирования этого сигнала, студент сможет выявить причину и устранить её.

В- третьих, сигнал окау, подтверждающий правильную работу ОУ, также на всем протяжении временной диаграммы должен быть равен единице. В случае несовпадения формируемого устройством результата с эталонным значением сигнал окау будет низким, а сигнал defect наоборот установится в единичное значение и останется неизменным до завершения текущего сеанса моделирования.