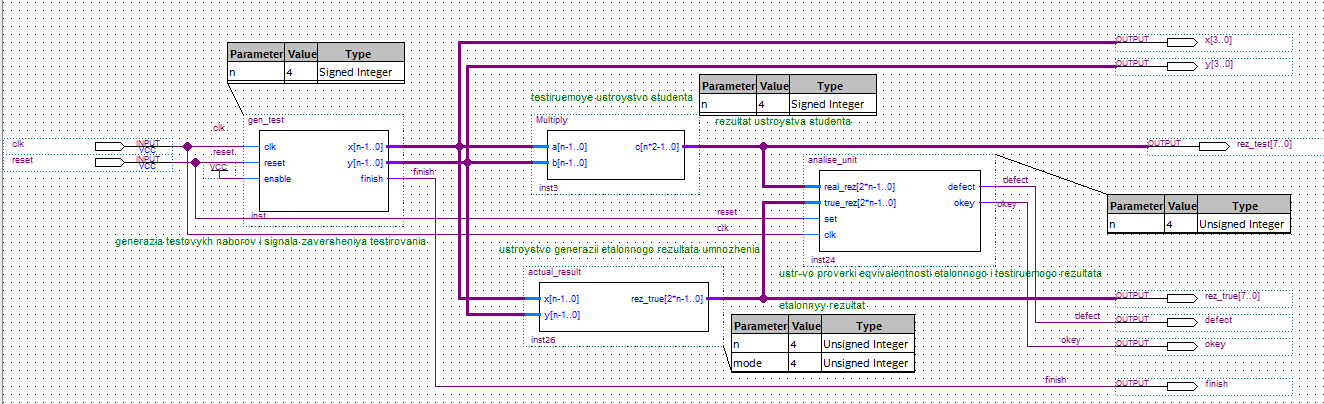
# Специализированный стенд для тестирования операционных устройств test\_algoritm

# Структура стенда test\_algoritm

Стенд представляет собой проект с именем test\_algoritm, выполненный в САПР QUARTUS II версии 18.1 на device Cyclone IV E. Все файлы проекта находятся в одноименной папке. Модуль верхнего уровня содержится в файле test\_alg.bdf. Он представляет собой схемный проект, приведенный на рис.

В состав стенда в качестве испытуемого входит операционное устройство, созданное студентами. На схеме представлен готовый тестовый блок **Multiply**. Помимо испытуемого устройства в состав стенда входят следующие блоки: генератор тестовых наборов **gen\_test**, блок формирования эталонного результата **actual\_result**, и блок анализа результата **analise\_unit**. Все модули стенда созданы с использованием языка VHDL, и их описания с подробными комментариями можно найти в одноименных файлах с расширением vhd. Для всех модулей доступно изменение разрядности чисел при помощи параметра n. Для изменения разрядности выходных сигналов потребуется редактирование пинов x, y, rez\_true, rez\_test;

Генератор тестовых наборов **gen\_test** представляет собой двоичный счетчик, который по положительному фронту синхро-сигнала clk формирует последовательные двоичные наборы, начиная с начального и до конечного. Начальное и конечные значения задаются в виде параметров, start\_counter и stop\_counter, 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. После того как генератор сформирует последний тестовый набор, он выработает сигнал finish, который покажет момент окончания тестирования.

Блок **actual\_result** формирует эталонный результат rez\_true, 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 этого параметра дополнительно приведены на схеме и в файле с описанием этого блока. Блок вычисляет значение произведения в прямом коде и преобразовывает в код, заданный параметром mode.

* 0 Прямой код, целые числа.
* 1 Прямой код, дробные числа.
* 2 Обратный код, целые числа.
* 3 Обратный код, дробные числа.
* 4 Дополнительный код, целые числа.
* 5 Дополнительный код, дробные числа.

Блок **analise\_unit** сравнивает результат, формируемый испытуемым устройством **Multiply**, с эталонным результатом rez\_true, вырабатываемым блоком **actual\_result**. 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 okey, формируется каждый раз после формирования сигнала конца операции. Его единичный уровень свидетельствует о совпадении результата с эталонным.

Стенд **stand\_test\_algoritm** 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\_counter в состояние, соответствующее первому тестовому набору. После снятия сигнала reset генератор тестовых наборов **gen\_test** начнет по положительному фронту тактового сигнала вырабатывать последовательные тестовые наборы. Тестовые наборы одновременно поступают в тестируемый блок **Multiply** блок формирования эталонного результата **actual\_result** и на выходные пины. По поступившим тестовым наборам блоки **Multiply** и **actual\_result** вырабатывают соответственно сигналы испытываемого произведения операционного устройства студента **rez** и эталоного произведия **rez\_true.** Выработанные сигналы поступают в блок анализа **analise\_unit**, который по положительному фронту тактового сигнала clock и при условии снятия сигнала reset сравнивает результат с эталонным. В случае совпадения сигналов, формируется высокий сигнал okey и низкий сигнал defect. Если хотя бы раз результат не совпадает с эталонным, defect до конца моделирования остается высоким сигналом, а okey в данном такте будет выработан низким сигналом. 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.

# 2 Подключение испытуемого устройства к стенду

Чтобы подключить операционное устройство к стенду необходимо в проекте удалить условное графическое обозначение операционного устройства **Multiply** и вместо него вставить символьное обозначение устройства, разработанного студентом, подобно тому как вставлялись структурные автоматы МИЛИ и МУРА в стенд TSTAND.

# 3 Тестирование операционного устройства с помощью специализированного стенда stand\_test\_algoritm

Чтобы выполнить тестирование операционного устройства с помощью стенда stand\_test\_algoritm, необходимо промоделировать поведение стенда с включенным в его состав испытуемым устройством. Для этого необходимо подать на стенд внешние сигналы reset, clk и mode. Первый сигнал устанавливает стенд и тестируемое устройство в исходное состояние. Второй сигнал используется для тактирования работы стенда и операционного устройства. Третий сигнал mode настраивает модуль **actual\_result** для корректного формирования эталонного результата, который зависит от кода и формата, в котором представлены операнды. В папке c проектом имеется файл с временными диаграммами Waveform. В нем необходимо изменить значение mode, в соответствии с вариантом задания. Кодирование приведено непосредственно на схеме стенда, в комментариях в файле actual\_result, в текстовом файле Text1.txt и инструкции. В случае ошибки при запуске моделирования Waveform возможно создать новую временную диаграмму University Program VMF. В первом такте сигнал reset необходимо поставить в высокое положение, на всем остальном участке моделирования в низком. Сигнал clock рекомендуется выставить с периодом до 3 ns.

Итак, меняем сигнал mode, выставляем разрядность вычислений n, при необходимости изменяем разрядность выходных пинов, компилируем и запускаем моделирование.

# 4 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. 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 okey, подтверждающий правильную работу ОУ, также на всем протяжении временной диаграммы должен быть равен единице. В зависимости от алгоритма возможно получение неправильного результата в случае умножения нуля или на нуль. В таком случае необходимо убедиться, что ошибка присутствует только при операциях с нулем или дополнить алгоритм для получения верного результата.