

## Специализированный стенд для тестирования структурных автоматов

### Структура стенда

Стенд представляет собой проект с именем TSTAND, выполненный в САПР QUARTUS II. Все файлы проекта находятся в одноименной папке. Модуль верхнего уровня содержится в файле TSTAND.bdf. Он представляет собой схемный проект, приведенный на рисунке 1.

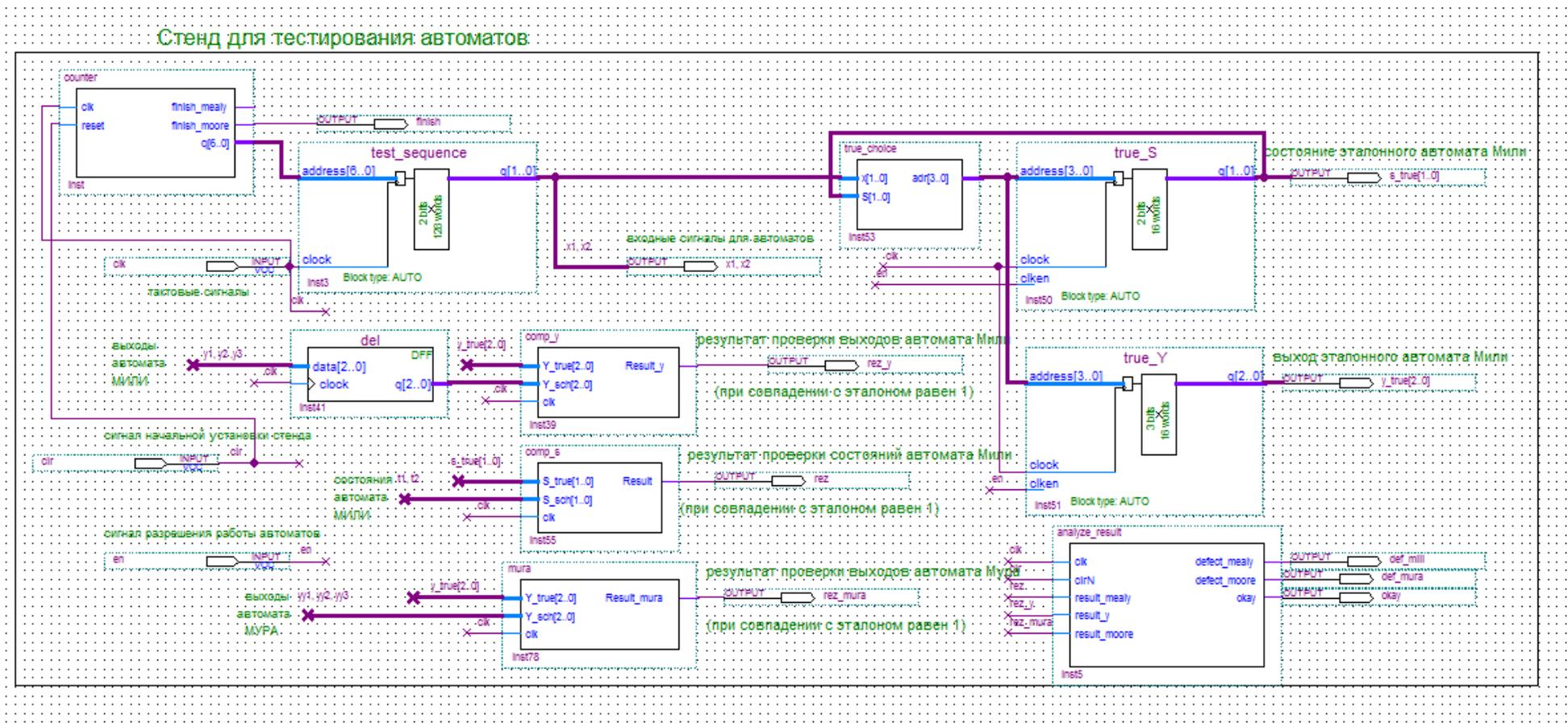
В состав стенда в качестве тестируемых автоматов входят структурные автоматы МИЛИ и МУРА, созданные студентами. Это блоки **avtomat\_mili** и **avtomat\_myra**, соответственно. Их условные графические обозначения (символы) изображены на рисунке 2.

Основными блоками стенда являются модуль **test\_sequence**, хранящий подаваемую на проверяемые автоматы тестовую последовательность входных наборов, блоки **true\_s** и **true\_y**, хранящие таблицы переходов и выходов эталонного автомата МИЛИ.

Модуль **counter** представляет собой двоичный счетчик, который формирует значения последовательных адресов, необходимых для извлечения из модуля постоянной памяти (блока **test\_sequence**) заданного количества тестовых наборов. Этот модуль также формирует сигнал **finish**, высокий уровень которого свидетельствует о подаче всех наборов из тестовой последовательности на испытываемые автоматы.

Модуль **true\_choice** выполняет формирование адреса для обращения к модулям памяти, хранящим таблицы переходов (блок **true\_s**) и выходов (блок **true\_y**) эталонного автомата МИЛИ. Эти адреса зависят не только от текущего

состояния автомата, но и от входного набора, подаваемого на автомат в текущий момент времени. Модуль создан с



использованием языка VHDL, и его описание находится в файле **true\_choice.vhd**.

Рис.1 Структура специализированного стенда TSTAND

## Синтезированные автоматы

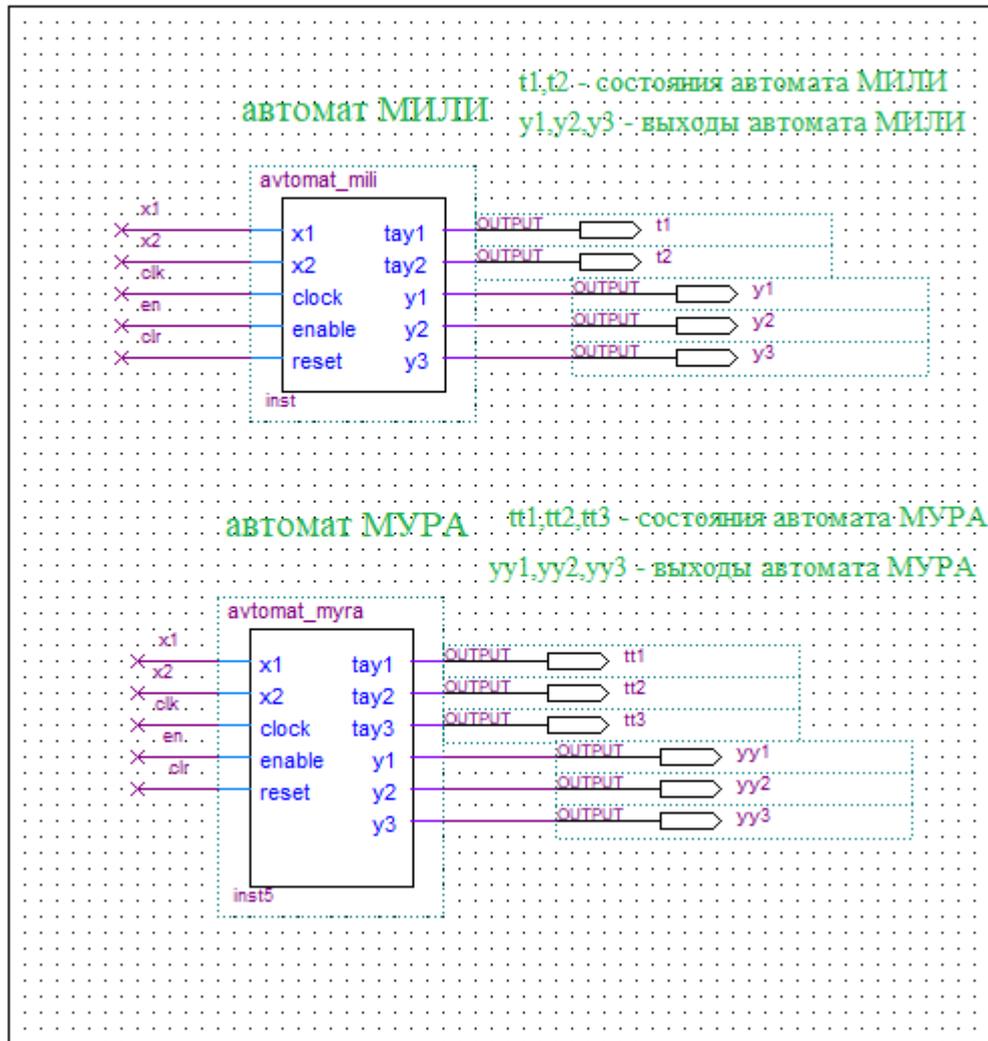


Рис.2 Тестируемые автоматы МИЛИ и МУРА

Модуль **analyze\_result** формирует сигналы `def_mili`, `def_myra` и `окау`, позволяющие оценить правильность работы тестируемых автоматов. Сигнал `окау` формируется в каждом такте, и его высокий уровень свидетельствует о корректной работе обоих автоматов в текущем такте. Высокий уровень сигналов `def_mili` и `def_myra` говорит о неправильной работе автомата МИЛИ и автомата МУРА соответственно, причём эти сигналы, единожды установившись в высокое состояние, сохраняют его до окончания сеанса моделирования.

Модуль **del** представляет собой трехразрядный регистр, в котором запоминается считанное с тестируемого автомата МИЛИ значение выходного сигнала  $Y$ , которое затем сравнивается с эталонным значением модулем **comp\_y**. Последний, в случае несовпадения формирует сигнал низкого уровня. Аналогичным образом работают модули **comp\_s** и **mura**, которые предназначены для анализа состояния тестируемого автомата МИЛИ и выходного сигнала автомата МУРА, соответственно. Эти модули созданы с использованием языка VHDL и их описания можно найти в одноименных файлах с расширением vhd.

В папку с проектом входят модули **KodirovkaX**, **KodirovkaS**, **KodirovkaY**, **KodirovkaMura**, которые в случае необходимости позволяют оперативно изменить кодировку входных сигналов, состояний автомата МИЛИ, выходных сигналов автомата МИЛИ, и выходных сигналов автомата МУРА, соответственно. Они не изображены на рисунке 1, однако их описания на языке VHDL можно найти в одноименных файлах в папке с проектом.

### **Подготовка тестовых наборов для проверки автоматов**

Перед началом тестирования структурных автоматов необходимо подготовить корректную тестовую последовательность, которая потом будет подаваться на спроектированные студентами автоматы МИЛИ и МУРА. Особенностью синтезированных автоматов является их неполная определенность, то есть существуют некоторые состояния и входные наборы, после подачи которых на вход автомата его поведение будет не определено. В соответствующих позициях таблиц переходов и выходов исходного автомата МИЛИ стоят прочерки. Поэтому в корректной тестовой последовательности входных сигналов для испытуемых автоматов таких наборов не должно быть.

Тестовую последовательность готовит приложение Gen, исполняемый файл которого входит в состав папки TSTAND. Также в эту папку входят текстовые файлы State и Pdoc. Файл State хранит таблицы переходов и выходов

для всех вариантов заданий курсовой работы, а файл Pdoc – зашифрованные тестовые последовательности для проверки синтезированных автоматов.

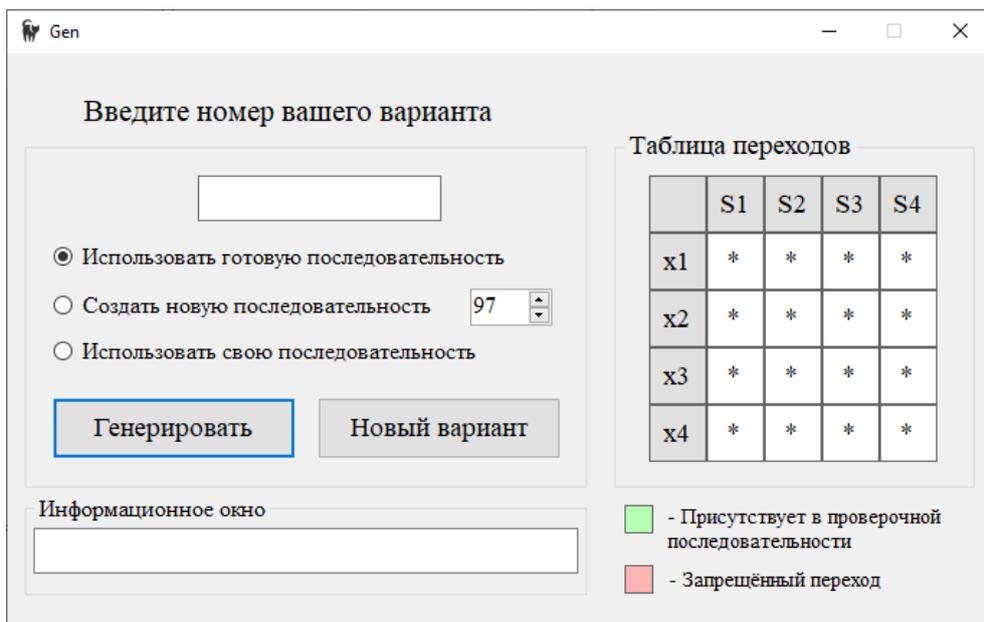


Рис.3 Форма экрана после запуска приложения Gen

Окно, выводимое на экран инструментального компьютера после запуска приложения, приведено на рисунке 3. Студенту помимо указания номера своего варианта задания также предлагается выбрать один из трех возможных способов формирования тестовой последовательности для испытания его автоматов.

Первый способ предполагает использование готовой последовательности. Она извлекается из файла Pdoc, хранящего тестовые последовательности минимальной длины для каждого варианта. Этот файл, вместе с файлом State формируется приложением **VGen** в процессе генерации новых вариантов заданий. Приложение выполняется каждый учебный год перед выдачей задания студентам и подготовленные им варианты заданий размещаются на странице сайта кафедры. При выборе готовой тестовой последовательности, в стенд будет загружена корректная последовательность тестовых наборов минимальной длины, проверяющая все переходы автомата МИЛИ, заданного в задании, и правильность формирования им выходных сигналов.

Второй способ предполагает создание новой тестовой последовательности с использованием генератора псевдослучайных наборов. Пользователь задает длину последовательности, и программа готовит корректную последовательность тестовых наборов заданной длины для введённого номера варианта задания следующим образом.

За начальное состояние автомата МИЛИ принимается состояние «00». Затем с помощью генератора псевдослучайных чисел формируется первый тестовый набор, предназначенный для проверки автомата. Приложение по таблице переходов эталонного автомата проверяет, не переключает ли этот входной набор автомат в неопределенное состояние. Если нет, то определяется следующее состояние автомата и входной набор помещается в тестовую последовательность, после чего аналогичным образом формируются все последующие наборы тестовой последовательности.

В противном случае, если входной набор переключает автомат в неопределенное состояние, то такой набор считается некорректным и в формируемую тестовую последовательность он не включается. Таким образом, приложением Gen готовится корректная последовательность тестовых наборов заданной длины для испытываемых автоматов.

Третий способ позволяет студенту для тестирования синтезированных им автоматов использовать заранее подготовленную им самим тестовую последовательность входных наборов. После нажатия на кнопку «Генерировать» вызывается окно, приведенное на рисунке 4, в котором пользователь должен сам ввести подготовленную последовательность. При выборе третьего способа создания тестовой последовательности пользователь берёт на себя ответственность за её качество. В случае, если последовательность окажется некорректной, то программа оповестит об этом пользователя в информационном окне.

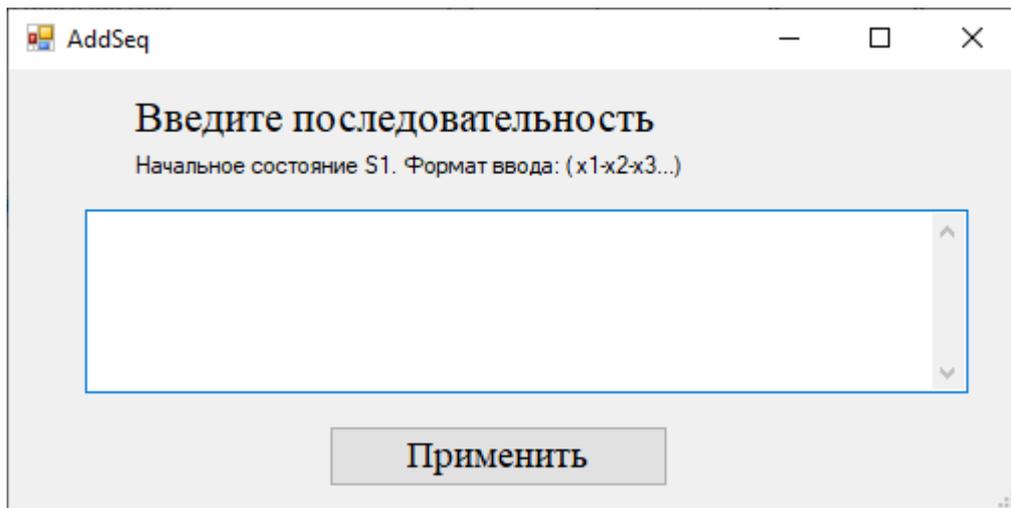


Рис.4 Окно ввода тестовой последовательности

После создания тестовой последовательности любым из предложенных способом, окно приложения Gen.exe будет выглядеть примерно, как на рис.5.

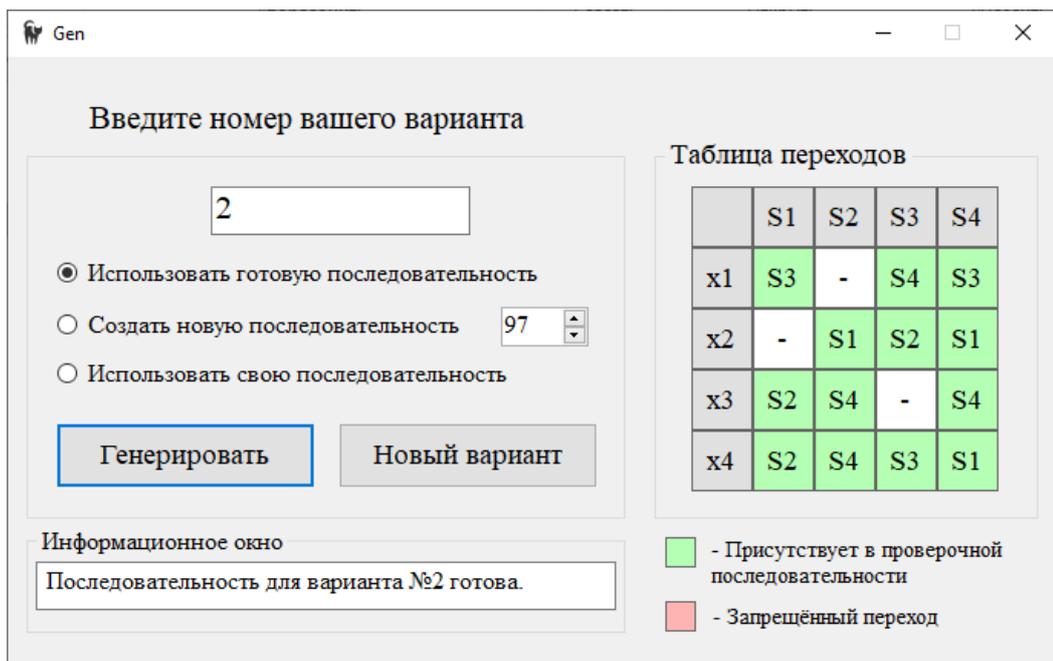


Рис.5 Окно приложения Gen.exe после генерации тестовой последовательности первым способом

В правой части окна располагается таблица переходов исходного автомата МИЛИ, соответствующая индивидуальному варианту задания, указанному студентом. Состояния в клетках таблицы закрашиваются зелёным цветом, если соответствующие переходы проверяются подготовленной

тестовой последовательностью и красным цветом отмечаются запрещённые переходы, если они обнаружатся.

Затем приложение формирует соответствующие заданному варианту файлы инициализации памяти модулей **true\_s** и **true\_y**. Эти модули представляют собой модули постоянной памяти, в которых размещаются таблицы переходов и выходов заданного номером варианта эталонного автомата МИЛИ. Созданная тестовая последовательность сохраняется в файле в папке стенда и подгружается в проект.

Затем готовится файл инициализации памяти модуля **test\_sequence** стенда. Этот модуль в процессе тестирования автоматов будет последовательно извлекать хранящиеся в нем тестовые наборы и подавать их на проверяемые автоматы.

Приложение Gen позволяет добавить новый произвольный вариант задания для синтеза автоматов. В этом случае при нажатии на кнопку «Новый вариант» откроется окно ввода данных в таблицы переходов и выходов.

Ввод данных

Таблица переходов

	S1	S2	S3	S4
x1	-	-	-	s*
x2	s*	s*	s*	s*
x3	s*	s*	s*	s*
x4	s*	s*	s*	s*

Таблица выходов

	S1	S2	S3	S4
x1	-	-	-	y*
x2	y*	y*	y*	y*
x3	y*	y*	y*	y*
x4	y*	y*	y*	y*

Ввод

Информационное окно

\* Формат вводимых данных должен быть сохранён  
\* Введите цифру (1, 2, 3, 4) вместо '\*', при неопределённом значении введите '!'

Рис.6 Окно задания таблиц переходов и выходов для нового варианта

В выводимом окне необходимо заполнить две таблицы: таблицу переходов и выходов. После нажатия кнопки «Ввод» и при корректном заполнении таблиц, новый вариант будет записан в файл вариантов в папке TSTAND. Главное окно приложения оповестит вас о добавлении нового варианта и выведет его номер. С добавленным вариантом можно работать также, как и с остальными, за исключением возможности использования готовой тестовой последовательности минимальной длины.

Описанный функционал будет полезен для студентов, которые вернулись из академического отпуска. Воспользовавшись им такие студенты смогут продолжить работу над своим проектом, задание на которое было выдано в прошлые годы и выполнить тестирование синтезированных ими автоматов.

Таким образом, результатом работы приложения Gen является подготовка файлов инициализации модулей памяти стенда с последовательностью тестовых наборов, подаваемых на входы тестируемых автоматов, и таблицами переходов и выходов эталонного автомата МИЛИ.

### **Подключение тестируемых автоматов к стенду**

Ниже приведена последовательность шагов, которые необходимо выполнить для подключения испытуемых автоматов к стенду TSTAND.

1. Откройте проект со стендом (файл **tstand.qpf**). Он находится в папке TSTAND. Одноименный файл с расширением bdf содержит модуль верхнего уровня. Во вложенной папке MY AVT найдите файлы с именами **avtomat\_mili.bdf** и **avtomat\_myra.bdf**. Эти файлы следует заменить на файлы с разработанными Вами структурными автоматами МИЛИ и МУРА, соответственно.
2. Расположение входных и выходных выводов в разработанных автоматах должно соответствовать приведенному на рисунке 2

примеру. В противном случае после создания символьных файлов разработанных автоматов и дальнейшем подключении их к стенду имеется вероятность неверной коммутации цепей схемы.

3. Последовательно выполните компиляцию добавленных в стенд автоматов, предварительно объявив их модулями верхнего уровня. Для этого необходимо выполнить команду **Set as Top-Level Entity** для каждого файла, соответственно. Создайте графические обозначения ваших автоматов используя команду **File→Create/Update→Create Symbol Files for Current file**. Проследите, чтобы созданные файлы с расширением bsf были сохранены в папке MY AVT.
4. Вернитесь в основной файл со стендом (tstand.bdf). Объявите его модулем верхнего уровня, как было описано выше. Найдите в нем графические обозначения тестируемых автоматов и удалите их. Вместо удаленных обозначений вставьте созданные в предыдущем пункте символы синтезированных автоматов. Рекомендуется проверить правильность подачи входных сигналов, сигналов синхронизации, установки и разрешения на соответствующие входы тестируемых автоматов, а также сигналов их состояния и выходов к соответствующим модулям стенда.

## **Тестирование автоматов в составе специализированного стенда TSTAND**

Замечательной особенностью стенда TSTAND является возможность одновременного использования в его составе автоматов обоих типов МИЛИ и МУРА, причем допустимы проекты, созданные с применением разных стилей проектирования. Заключение о соответствии спроектированных студентами автоматов заданному варианту выполняется на основании моделирования стенда с включенными в него автоматами МИЛИ и МУРА. Для этого подготовлен файл **test.wvf**, содержащий временные диаграммы подаваемых на стенд сигналов. Они приведены на рисунке 7.

Сигнал «clk» является тактовым сигналом. Он поступает на все элементы памяти стенда, включая элементы памяти тестируемых автоматов. По положительному фронту сигнала «clk» из модуля **test\_sequence** извлекаются подаваемые на тестируемые автоматы наборы, фиксируются выходные сигналы автомата МИЛИ регистром **del**, после чего они сравниваются с эталонными значениями.

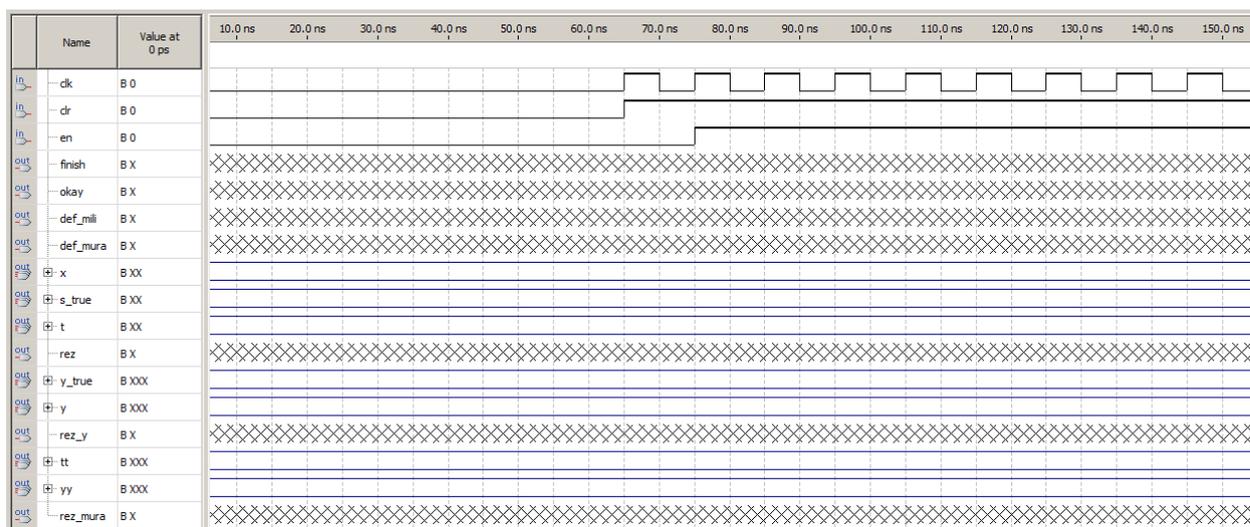


Рис. 7 Временные диаграммы подаваемых на стенд сигналов

Тестовая последовательность может содержать до девяноста семи наборов. В первом такте по сигналу «clr» обнуляется счётчик counter и элементы памяти тестируемых автоматов устанавливаются в начальное состояние, что позволит снять неопределенность в дальнейшем поведении автоматов. Сигнал «en» также подается на все элементы памяти стенда, включая память тестируемых автоматов. В первом такте этот сигнал не активен, так как в этом такте выполняется установка в начальное состояние тестируемых автоматов. Все последующие такты сигнал «en» активен, что разрешает реагировать на тактовый сигнал элементам памяти стенда. Остальные сигналы на временной диаграмме являются выходными для стенда, поэтому на временной диаграмме они не определены и им соответствуют неопределенные значения. Выходные сигналы будут сформированы в процессе моделирования.



Следующая группа сигналов «s\_true», представляет собой ожидаемое состояние эталонного автомата МИЛИ, в которое он должен переключаться по положительному фронту тактового сигнала. Группа сигналов «y\_true» представляет собой выходные сигналы эталонного автомата МИЛИ. Сигналы «s\_true» и «y\_true» извлекаются из соответствующих модулей памяти **true\_s** и **true\_y**, которые хранят таблицы переходов и выходов эталонного автомата МИЛИ.

Группы сигналов «t» и «y» отображают состояние и выходные сигналы тестируемого автомата МИЛИ. А группы «tt» и «yy» отображают состояние и выходные сигналы тестируемого автомата МУРА.

Следует заметить, что выходные сигналы тестируемого автомата МИЛИ «y» на один такт опережают сигналы эталонного автомата «y\_true». Это связано с необходимостью извлечения последних из модуля памяти **true\_y**. Поэтому для корректного сравнения этих сигналов модулем **comp\_y** используется дополнительный регистр **del**, задерживающий выходные сигналы тестируемого автомата МИЛИ на один такт.

На временной диаграмме отображаются сигналы «res» и «res\_y», которые формируются стендом по результатам сравнения состояний и выходных сигналов испытуемого автомата МИЛИ с ожидаемыми соответствующими значениями эталонного автомата. Если эти сигналы равны единице на всем интервале временной диаграммы, то это означает, что работа тестируемого автомата МИЛИ совпадает с работой эталонного автомата. В противном случае низкий уровень сигналов «res» и «res\_y» будет говорить о несоответствии работы тестируемого автомата МИЛИ с эталонным автоматом. Причем временные интервалы, в которых это несоответствие наблюдается, а также анализ других сигналов на временной диаграмме позволит студентам понять причину и устранить её.

Сигналом, подтверждающим корректную работу эквивалентного автомата МУРА, является сигнал «rez\_mura», который на всем протяжении

временной диаграммы должен принимать единичное значение. Этот сигнал формируется на основе сравнения выходных сигналов тестируемого автомата МУРА и выходных сигналов эталонного автомата. Как можно видеть на Рис. 8, на выходе тестируемого автомата МУРА наблюдается такая же последовательность, как и на выходе автомата МИЛИ, смещенная на один такт. Это связано с особенностью функционирования автомата МУРА.