

Иерархическое проектирование простых логических схем в LabVIEW для использования в оценочном модуле DE FPGA Board

Цель работы:

Целью этой лабораторной работы является приобретение базовых знаний и навыков иерархического (модульного) проектирования схем, обучение созданию sub-VI и компоновке низкоуровневых функциональных блоков в LabVIEW. Результаты проектирования вы проконтролируете на оценочном модуле DE FPGA и системе NI ELVIS.

Задание на проектирование:

Разработать простейшую логическую схему с тремя входами и одним выходом. Выходной сигнал схемы должен быть равен TRUE, если количество входов с состоянием TRUE меньше или равно единице. Схема должна быть иерархической; используя вначале функции самого низкого уровня, вы должны создать sub-VI, которые затем послужат функциональными блоками более высокого уровня.

Выполнение задания:

В проектируемой схеме 3 входа и один выход. Обычно такие устройства могут быть реализованы, как комбинационные схемы. По заданной таблице истинности можно, воспользовавшись некоторыми методами минимизации (например, К-Мар – картами Карно), уменьшить количество необходимых логических элементов. Таблица истинности рассматриваемого устройства приведена ниже вместе с картой Карно и минимизированным булевым выражением.

Inputs			Output
A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

		AB			
		00	01	11	10
C	0	1	1	0	1
	1	1	0	0	0

$$F = \bar{A}\bar{C} + \bar{B}\bar{C} + \bar{A}\bar{B}$$
$$F = (\bar{A} + \bar{C}) * (\bar{B} + \bar{C}) * (\bar{A} + \bar{B})$$

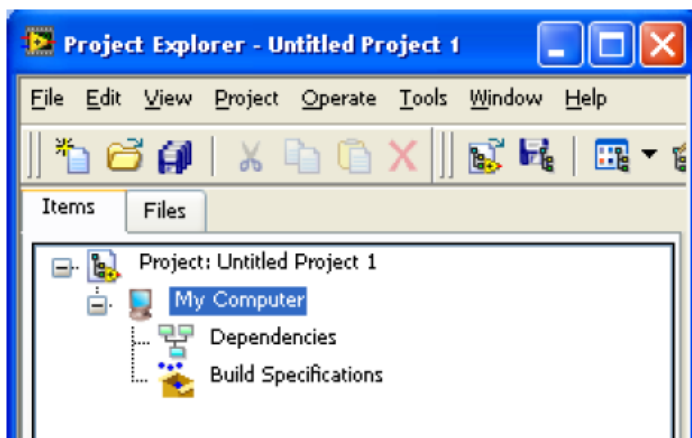
Оба выражения эквивалентны. При проектировании мы будем использовать 2-е выражение.

Для быстрого освоения LabVIEW выполните курс “Learn LabVIEW in 3 hours” (http://www.ni.com/academic/learn_LabVIEW/)

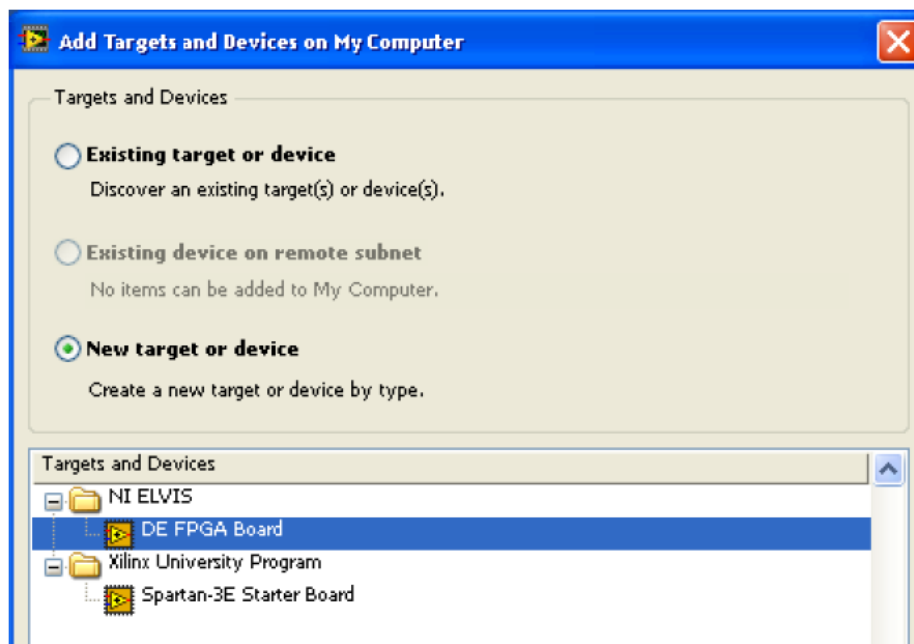
Раздел 1: разработка устройства

Порядок выполнения:

- Разархивируйте файл **minoritygate_Lab.zip** в папку **c:\NI\LabVIEW_Labs**
- Запустите на исполнение LabVIEW (**Start>>All Programs>>National Instruments>>LabVIEW**)
- Выберите вариант **File>>New Project**
- В окне *Project Explorer* щелкните правой кнопкой мыши по пункту **My Computer** и выберите **New>>Targets and Devices**

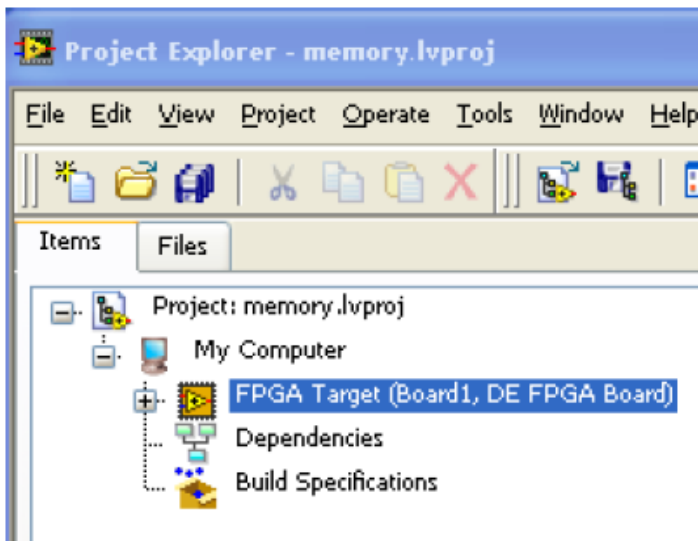


- Установите флажок в позиции *New target or device* и выберите **DE FPGA Board** в секции **NI ELVIS**



- Щелкните по кнопке **OK**
- Выберите в меню **File>>Save** и сохраните проект под именем **minoritygate** в папке **c:\NI\LabVIEW_Labs\ minoritygate_Lab**

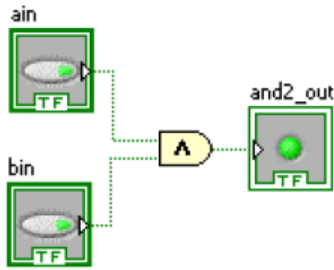
- В окне *Project Explorer* щелкните правой кнопкой мыши по пункту **FPGA Target (Board1, DE FPGA Board)** и выберите **New VI**



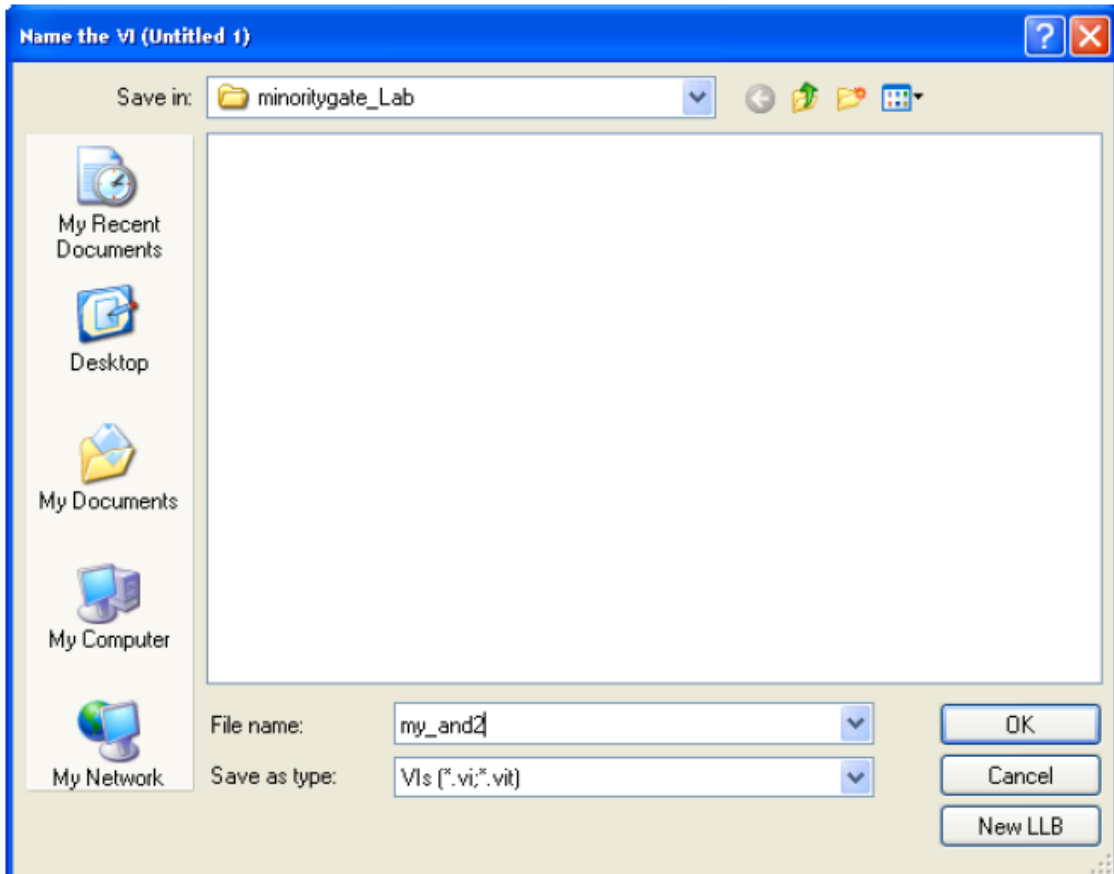
- Откройте окно **Block Diagramm** (Ctrl-E) и щелкните правой кнопкой мыши где-нибудь на белом поле (в рабочей области). В палитре функций откройте субпалитру **Boolean**. Вы можете также найти ее щелчком по кнопке Search



- Выберите двухвходовый элемент **and** и поместите его на рабочую область блок-диаграммы
- Переместите курсор на левый угол элемента **and** и обратите внимание, что курсор принял вид инструмента соединения. Щелкните правой кнопкой мыши и выберите **Create>> control**. Создастся элемент управления с именем, по умолчанию, **x**. Измените имя на **ain**. Аналогично создайте второй элемент управления (для второго входа элемента **and**) и присвойте ему имя **bin**. Создайте индикатор для выхода элемента **and** и присвойте ему имя **and2_out**. Перегруппируйте элементы управления и индикатор так, чтобы блок-диаграмма стала похожей на показанную ниже



- Выберите в меню **File>>Save** и сохраните VI под именем **my_and2** в папке **c:\NI\LabVIEW_Labs\memory_Lab**




- Переключитесь на лицевую панель (Ctrl-E) **my_and2**, щелкните правой кнопкой мыши по иконке, расположенной в правом верхнем углу, и выберите *Show Connector*
- Еще раз щелкните правой кнопкой мыши по иконке, выберите пункт *patterns* (шаблоны), а затем – подходящий шаблон с достаточным количеством блоков




для входов и выходов вашего VI, (например, )

- Щелкните по входу в левом верхнем углу окна коннектора, а затем щелкните по кнопке **ain**; щелкните по входу в левом нижнем углу окна коннектора, а затем щелкните по кнопке **bin**; щелкните по крайнему правому блоку коннектора, а



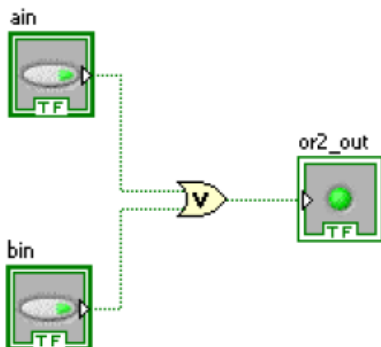
затем по индикатору выхода **and2_out** (). Запомните, что эти операции надо выполнять именно в таком порядке (вначале – коннектор, затем элемент

управления или индикатор. Если вы проделаете это в другом порядке, подключения могут получиться другими.

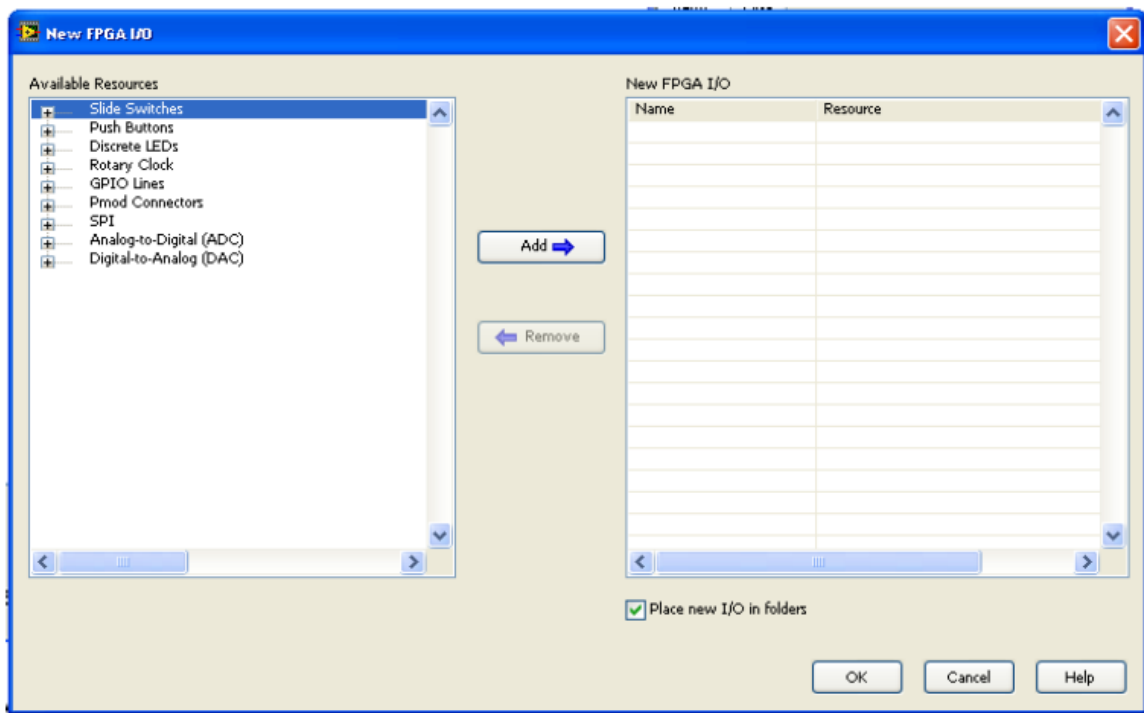
- Щелкните правой кнопкой мыши по окну коннектора и выберите **Edit Icon**
- В окне редактора иконок выберите **Edit>>Clear** для очистки рисунка
- В палитре инструментов выберите А – инструмент ввода текста
- В очищенном окошке введите **MY** (затем нажмите на клавишу enter), а затем **AND2**
- В палитре инструментов выберите инструмент рисования прямоугольника и нарисуйте в окошке прямоугольник
- Если необходимо, воспользуйтесь инструментом () для выравнивания содержимого окошка, чтобы иконка выглядела так, как показано ниже



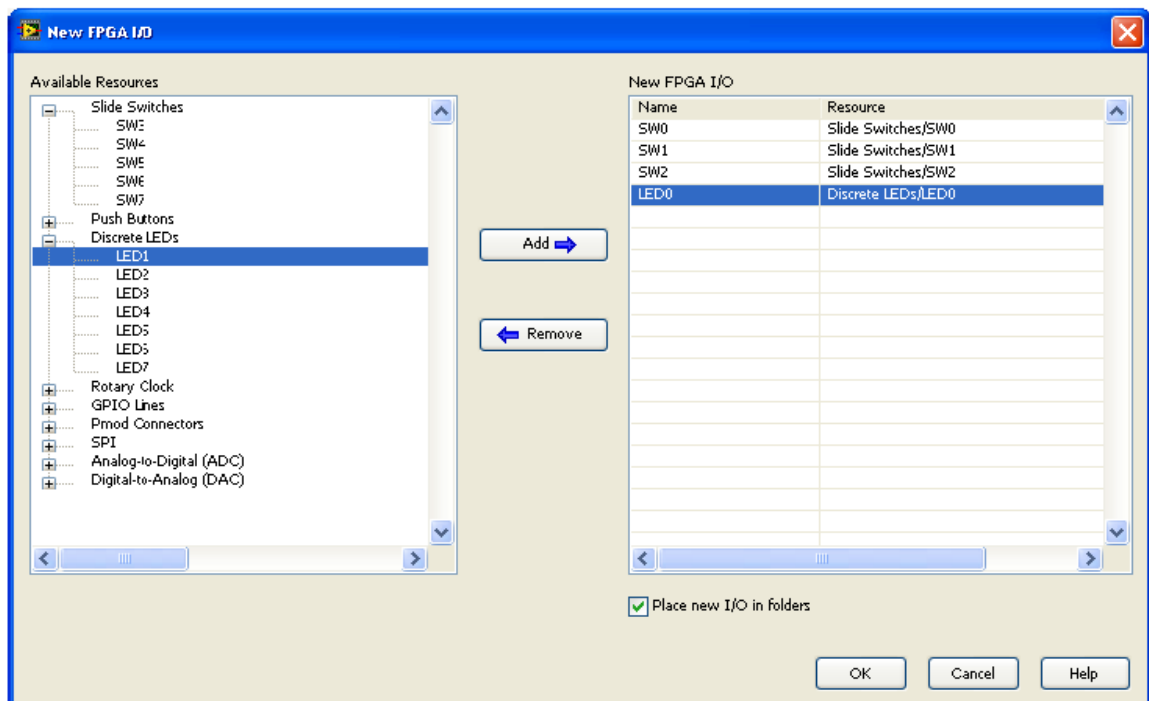
- Щелкните по кнопке **OK** и сохраните vi
- Таким же образом создайте новый VI, разместите в нем 2-входный элемент or и создайте элементы управления и индикации, как показано ниже



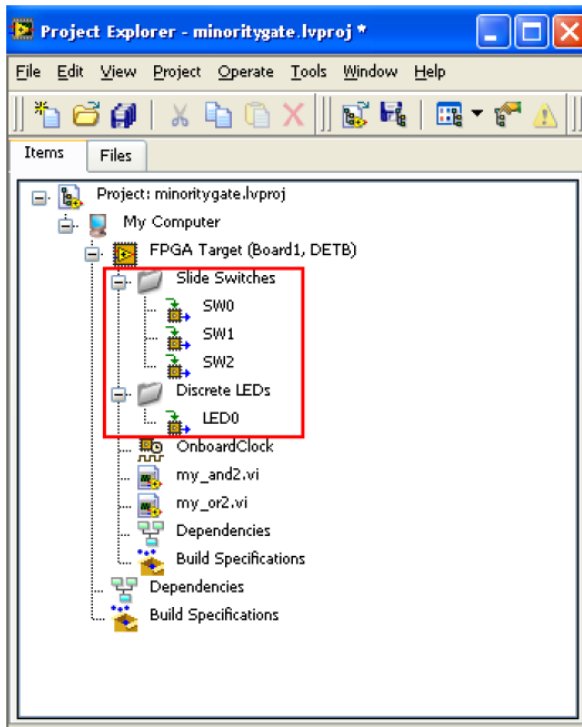
- Преобразуйте его в sub-VI, измените иконку на **MY OR2** и сохраните, как **my_or2**
- Закройте все VI
- Мы завершили создание простых sub-VI низкого уровня и теперь нужно спроектировать на их основе схему более высокого уровня. Но вначале нам необходимо добавить каналы ввода-вывода (FPGA IO), после чего можно будет реализовать требуемую функциональность
- В окне *Project Explorer* щелкните правой кнопкой мыши по пункту **FPGA Target (Board1, DE FPGA Board)** и выберите **New>>FPGA IO**
- Доступные в оценочном модуле (DE FPGA Board) каналы ввода-вывода отображаются в левой части окна выбора



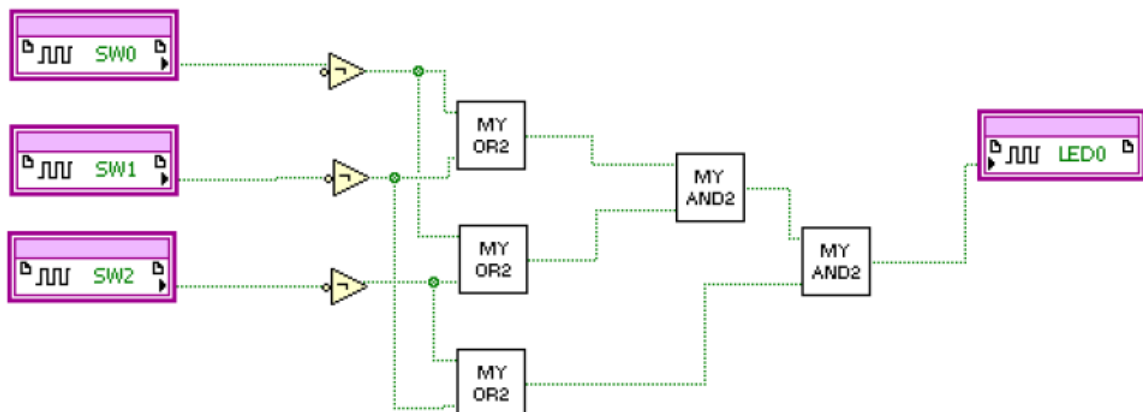
- В секции **Available Resources** раскройте папку **Slide Switches** и выберите **SW0**, **SW1** и **SW2**. Щелкните по кнопке **Add**, чтобы добавить выбранные элементы в проект. Аналогично, раскройте папку **Discrete LEDs**, выберите **LED0** и щелкните по кнопке **Add**. С добавленными каналами ввода-вывода окно выбора будут выглядеть так



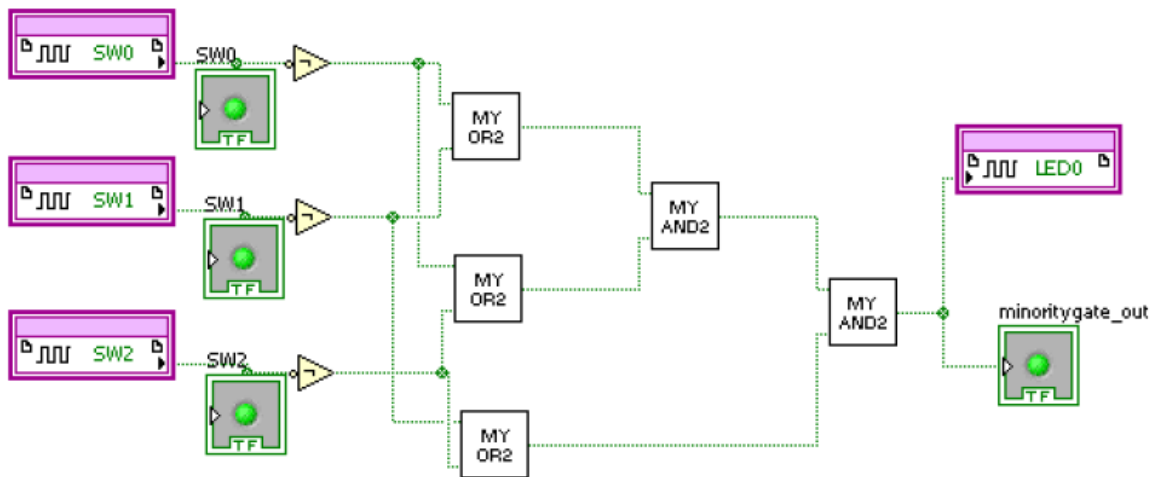
- Щелкните по кнопке **ОК** для подтверждения изменений и закройте окно выбора
- Ниже на рисунке показано, как должно выглядеть окно **Project Explorer** с внесенными изменениями



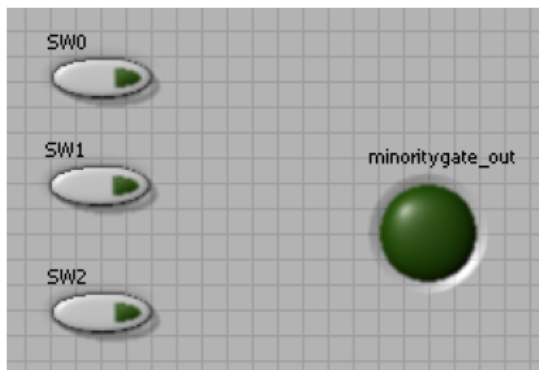
- Создайте новый VI
- Войдите в окно блок-диаграммы
- В окне **Project Explorer** выберите по одному и перетащите в рабочую область блок-диаграммы переключатели (*Switches*). Аналогично выберите светодиод (LED) и перетащите его в рабочую область блок-диаграммы
- Добавьте инверторы (логический элемент NOT) из субпалитры **Boolean**, sub-VI **my_and2** и **my_or2** и соедините их, как показано ниже (обратите внимание, что полученная блок-диаграмма теперь представляет собой соответствующее уравнение, которое мы получили вначале)



- Добавьте индикаторы (щелкнув правой кнопкой мыши по проводнику и выбрав **Create>>Indicator**) для переключателей и выхода, как показано ниже. Переименуйте выходной индикатор на **minoritygate_out**




- Переключитесь на лицевую панель (**Front Panel**) (Ctrl-E) – на ней видны три переключателя и индикатор выхода. Если хотите, можно изменить их форму и переконфигуровать лицевую панель. Ниже показана одна из возможных компоновок

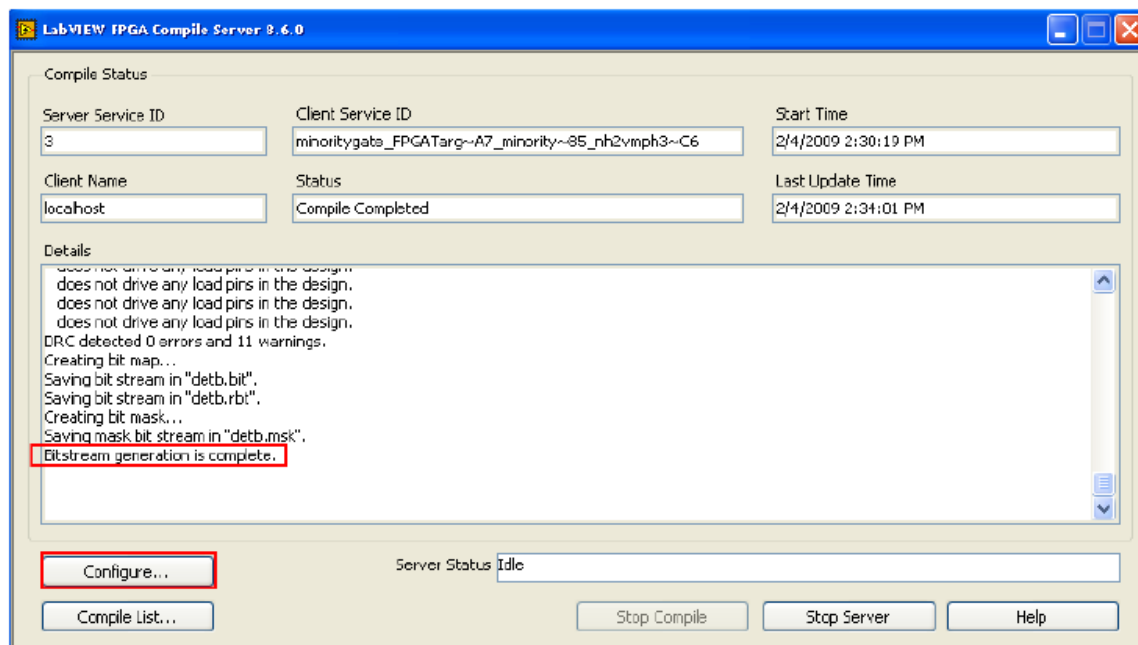


- Сохраните VI под именем **minoritygate.vi** в той же папке

Раздел 2: Проверка результатов проектирования с помощью оценочного модуля


Порядок выполнения:

- Убедитесь, что **minoritygate.vi** открыт
- Щелкните по кнопке **Run** () или выполните команду меню **Operate>>Run**



- Как только завершится генерация двоичного кода **Bitstream**, о чем появится сообщение в окне хода компиляции **Compile Server**, закройте его щелчком по кнопке **X** в правом верхнем углу окна. Щелкните по кнопке **OK**, чтобы закрыть сводное окно состояния компиляции
- Щелкните также по кнопке **X** в правом верхнем углу окна сервера компиляции, чтобы закрыть это окно

FPGA сконфигурируется и, в зависимости от состояния переключателей, светодиод LED0 может быть включен или выключен. Если вы попытаетесь изменять состояния переключателей, то не увидите изменений на LED0, т.к. программа однократно выполнилась и остановилась

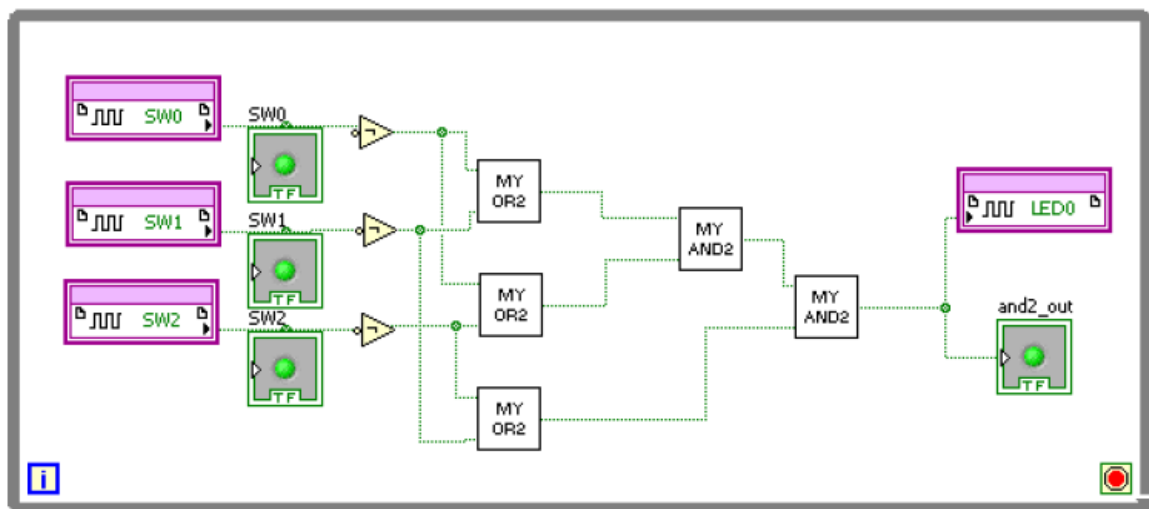
- Если вы хотите видеть состояние выхода схемы при различных положениях переключателей, то вначале измените состояния переключателей, а затем щелкните по кнопке **Run** () для конфигурирования FPGA и однократного запуска устройства. При этом двоичный код конфигурирования загружается в FPGA через USB-JTAG кабель. В зависимости от состояния переключателей **SW0-SW2** будет видно изменение выхода на светодиоде **LED0**. И положение переключателей, и состояние **LED0** отражаются в окне лицевой панели
- Обратите внимание, что изменения состояния переключателей, не изменяющие выход **LED0**, не отражаются на лицевой панели

- Щелкните по кнопке **Run Continuously** () и теперь вы можете, изменяя состояния переключателей, наблюдать изменения выхода **LED0** на лицевой панели. Щелкните по кнопке **Stop** () для прерывания моделирования

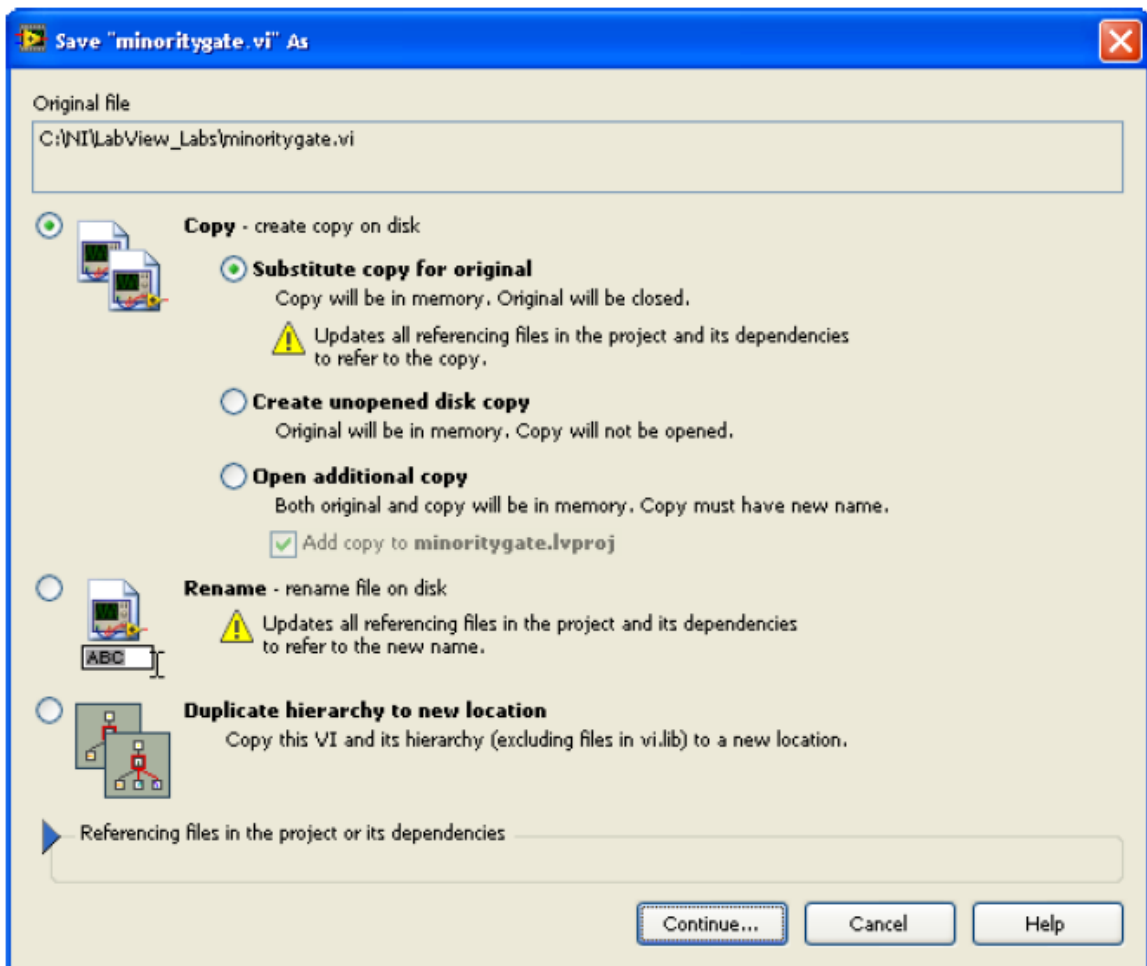
Раздел 3: Изменение блок-диаграммы для работы в непрерывном режиме

Порядок выполнения:

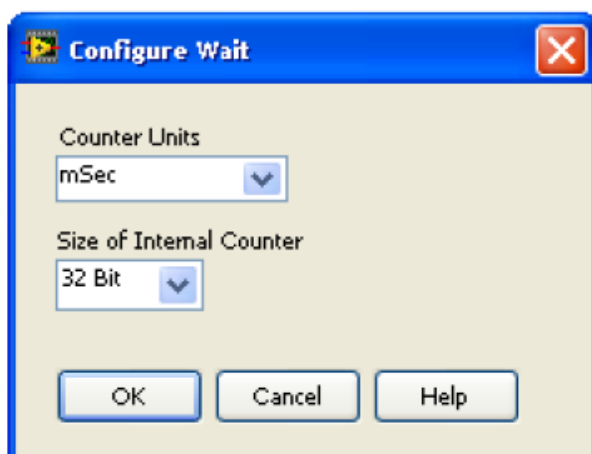
- Чтобы программа запускалась в непрерывном режиме кнопкой **Run**, вы должны заключить схему в цикл **while loop**, управляемый ключом
- Щелкните правой кнопкой мыши где-нибудь в рабочей области и выберите из палитры **Structures** структуру **while loop**. Нарисуйте прямоугольник, заключающий в себе созданную блок-диаграмму




- Сохраните VI командой меню **File>>Save As**

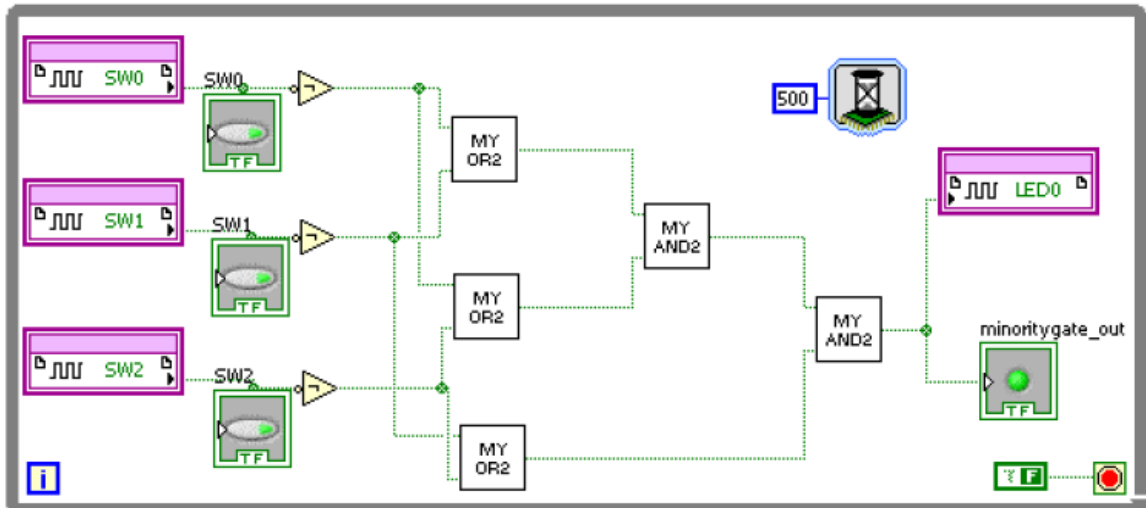


- Щелкните по кнопке **Continue...**
- Сохраните VI под именем **minoritygate_whileloop**
- Щелкните правой кнопкой мыши в окне **Block-Diagram** и добавьте функцию wait из палитры **Programming>>Timing**. Выберите единицу счета **msec** и щелкните по кнопке **OK**



- Щелкните правой кнопкой мыши по пиктограмме wait и выберите **Create>>Constant**. Измените значение константы с **0** на **500**
- Щелкните правой кнопкой мыши по терминалу **Loop Condition** и выберите из палитры **Boolean** константу ()

- Созданная блок-диаграмма должна выглядеть так



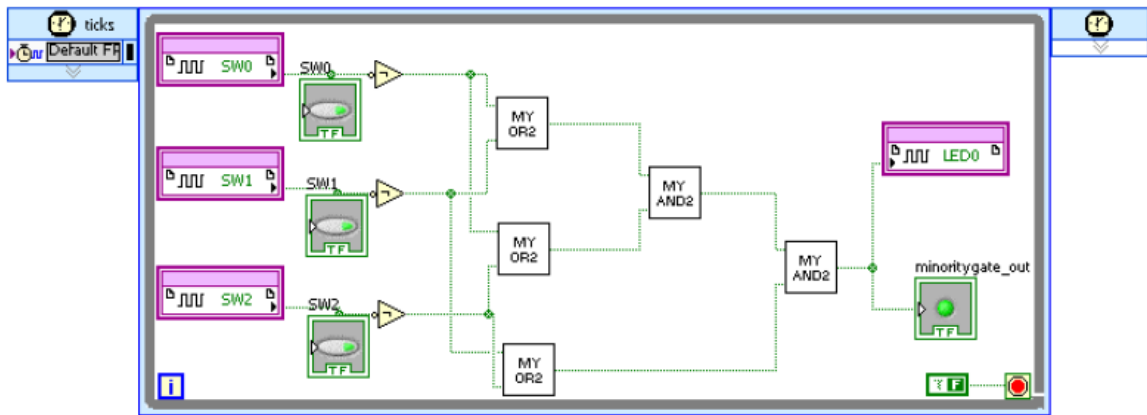
- Сохраните VI
- Щелкните по кнопке Run. Схема компилируется. Как только завершится генерация двоичного кода **Bitstream**, закройте диалоговое окно щелчком по кнопке **ОК**. FPGA сконфигурируется и вы можете тестировать схему, изменяя положение переключателей, не перезапуская при этом VI каждый раз и не пользуясь кнопкой **Run Continuously**

Раздел 4: Изменение блок-диаграммы для моделирования комбинационной схемы

Порядок выполнения:

С ранее спроектированной блок диаграммой моделируемая схема работала, как тактируемая, а не как комбинационная, поскольку код логики был включен в структуру while loop. Это увеличивает задержку. В нашей схеме 4 уровня логики и задержка равна 6 (4 – на уровни логической схемы и 2 – задержка while loop). Если вы хотите чтобы созданная схема или ее часть вела себя, как настоящая комбинационная схема, вам необходимо заключить код в структуру Single-Cycle Timed Loop. Последующие шаги покажут, как проектировать схему, выполняющую все операции за один такт.

- Сохраните VI командой меню **File>>Save As** под именем **minoritygate_singlecycle**
- Удалите функцию wait и подключенную к ней константу
- Щелкните правой кнопкой мыши по структуре while loop и выберите **Replace with Timed Loop**. Другой способ – удалить while loop из контекстного меню, щелкнуть правой кнопкой мыши где-нибудь на блок-диаграмме и, выбрав **Structures>>Timed Structure>> Timed Loop**, нарисовать цикл вокруг схемы. Добавьте константу для управления циклом. Блок-диаграмма должна выглядеть, как показано ниже

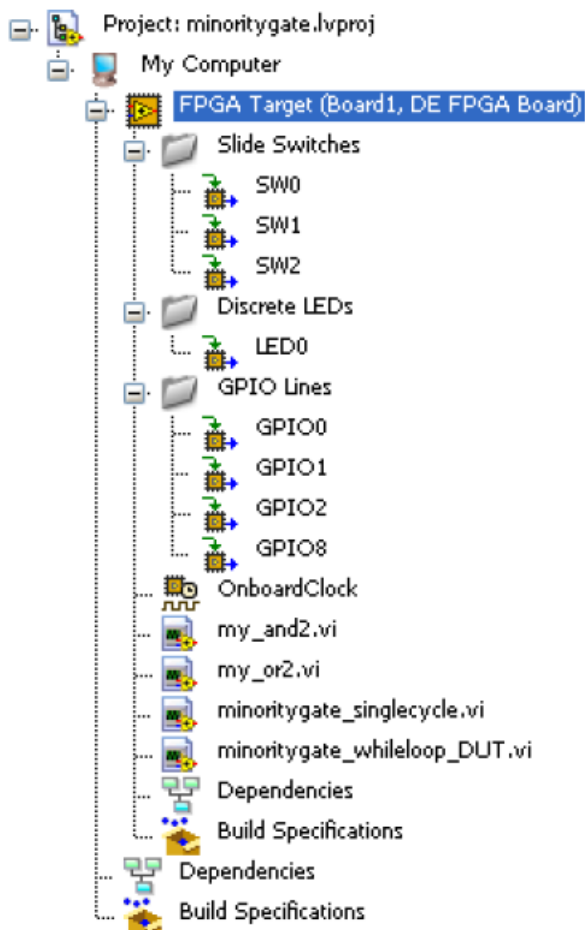


- Сохраните VI
- Щелкните по кнопке Run. Схема компилируется. Как только завершится генерация двоичного кода **Bitstream**, закройте диалоговое окно щелчком по кнопке **ОК**. FPGA сконфигурируется и вы можете тестировать схему, изменяя положение переключателей

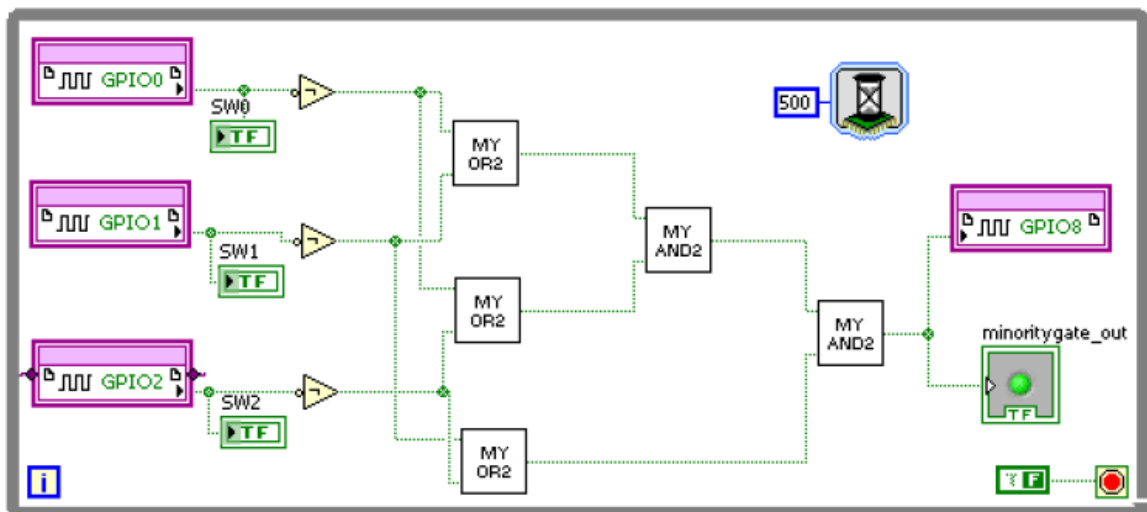
Раздел 5: Проверка результатов проектирования с помощью VI тестера

Порядок выполнения:

- В окне *Project Explorer* щелкните правой кнопкой мыши по пункту **FPGA Target (Board1, DE FPGA Board)** и выберите **Add File**
- Найдите в папке проекта файл **minoritygate_whileloop.vi** и добавьте его в проект
- Откройте **minoritygate_whileloop.vi**
- Сохраните VI под именем **minoritygate_whileloop_DUT.vi**
- Сохраните VI под именем **ram_memory_DUT.vi**
- На блок-диаграмме удалите элементы SW0, SW1, SW2 и LED0, оставив соответствующие индикаторы
- В окне *Project Explorer* щелкните правой кнопкой мыши по пункту **FPGA Target** и выберите **New>>FPGA I/O**
- В открывшемся окне откройте папку линий ввода-вывода **GPIO**, выделите линии **GPIO0**, **GPIO1**, **GPIO2** и **GPIO8**, добавьте их в проект, а затем щелкните по кнопке **ОК**
- Окно *Project Explorer* должно выглядеть так



- Выделите добавленные линии GPIO и перетащите их в цикл while loop
- Подключите линии GPIO0, GPIO1 и GPIO2 вместо переключателей SW0, SW1 и SW2
- Щелкните правой кнопкой мыши по узлу ввода-вывода GPIO8 и измените его назначение для записи
- Подключите выход схемы ко входу узла GPIO8. Блок-диаграмма должна выглядеть, как показано ниже



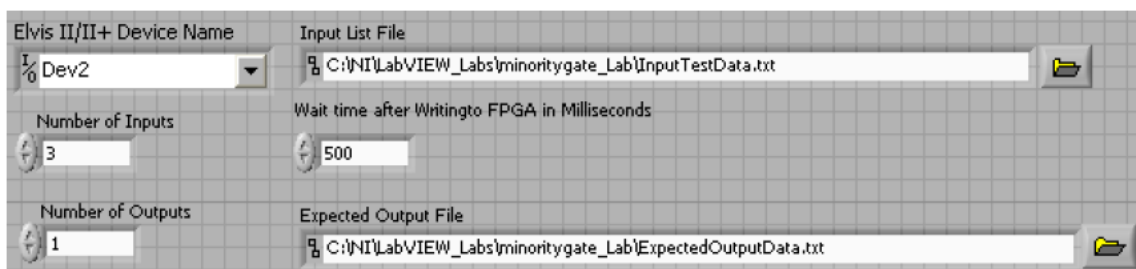
- Сохраните VI
- Щелкните по кнопке **Run**. Схема откомпилируется. Как только завершится генерация двоичного кода, закройте диалоговое окно щелчком по кнопке **OK**. FPGA сконфигурирован
- В окне *Project Explorer* щелкните правой кнопкой мыши по пункту **My Computer**, выберите **Add File**, найдите в папке проекта файл **Simple FPGA Tester.vi** и добавьте его в проект
- Двойным щелчком откройте **Simple FPGA Tester.vi**
- Ознакомьтесь с блок-диаграммой и разберитесь с ее содержанием
- Обратите внимание, что Digital Writer использует каналы DIO с 8 по 15 [канал 15 - младший] для записи в FPGA, а Digital Reader использует каналы DIO с 0 по 7 [канал 7 - младший] для чтения откликов из FPGA
- Проводниками для макетной платы физически соедините контакты разъемов BB5 и BB2 согласно следующей таблице

BB5 Connector	BB2 Connector
DIO15	GPIO0
DIO14	GPIO1
DIO13	GPIO2
DIO7	GPIO8

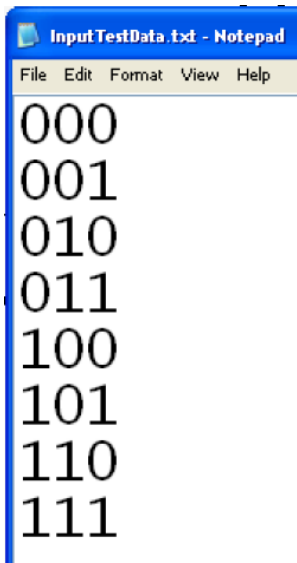
- Щелкните по стрелке в поле **Elvis II/II+ Device Name** и выберите устройство, соответствующее Elvis II



- Измените **Number of Inputs** на **3** и **Number of Outputs** на **1**
- Измените время Wait на **500 ms**
- Щелкните по кнопке Browse для файла Input и добавьте **rom_InputTestData.txt**
- Таким же образом добавьте **ExpectedTestData.txt** с ожидаемыми состояниями выходов. На этом этапе проектирования лицевая панель должна выглядеть похожей на изображенную ниже

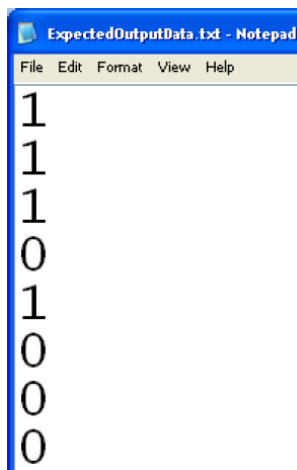


Файл входных данных – текстовый, его можно открыть/создать любым текстовым редактором. На следующем рисунке показано содержимое этого файла



```
InputTestData.txt - Notepad
File Edit Format View Help
000
001
010
011
100
101
110
111
```

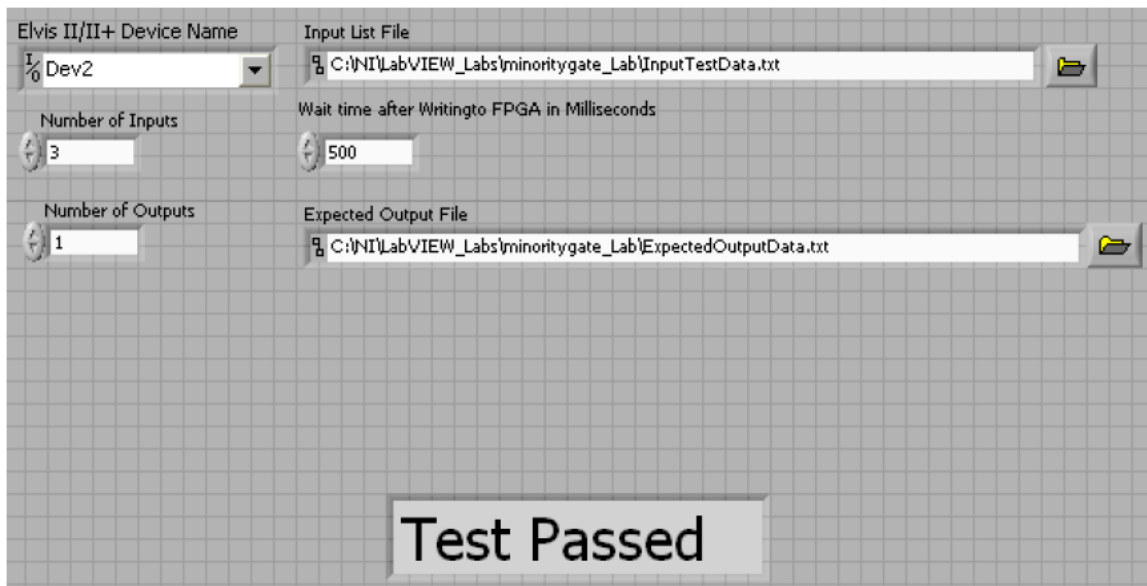
Обратите внимание, что количество входов меньше 8, для старших каналов дополнение нулями выполняется в тестируемом VI. В соответствии с этим файлом входных данных каналам (DIO) с 8-го по 12-й автоматически присваиваются нули, каналу 13 присвоен крайний левый бит (SW2), каналу 14 присвоен средний бит (SW1), каналу 15 присвоен крайний правый бит (SW0). Файл выходных данных – тоже текстовый, и его тоже можно открыть/создать любым текстовым редактором. На следующем рисунке показано содержимое этого файла



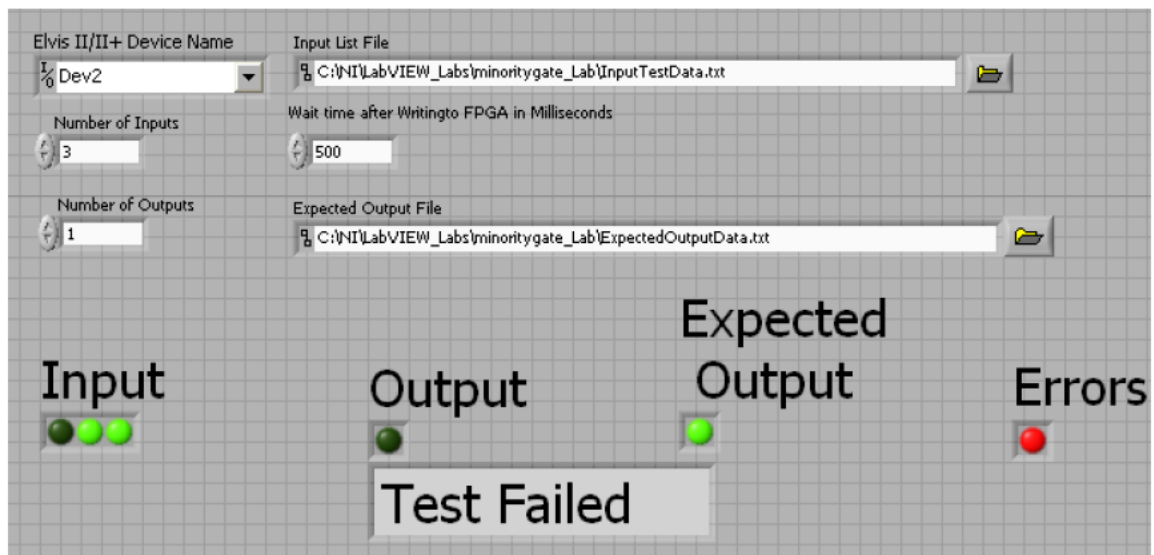
```
ExpectedOutputData.txt - Notepad
File Edit Format View Help
1
1
1
0
1
0
0
0
```

В соответствии с файлом выходных данных каналам (DIO) с 0 по 6 автоматически присваиваются нули, каналу 7 присвоен только столбец, показанный на рисунке выше

- Удостоверьтесь, что minoritygate_whileloop_DUT.vi продолжает исполняться
- Щелкните по кнопке Run на лицевой панели тестера. Тест запустится на исполнение, а его результаты отображаются на лицевой панели



- Теперь измените содержимое файла ожидаемых выходных реакций. Измените четвертый набор выходных сигналов с 0 на 1, при этом тест должен показать ошибку. Сохраните изменения
- Выключите и включите питание ELVIS и оценочного модуля FPGA
- Запустите тест еще раз и обратите внимание, что тестер сообщает об ошибке (Failed). Он также сообщает, где ошибка, какие были входные сигналы, ожидаемые выходные сигналы, какие были реальные выходные сигналы, включает индикатор ошибки



- Остановите исполнение minoritygate_whileloop_DUT.vi, закройте проект, закройте LabVIEW, выключите питание NI ELVIS II и оценочного модуля

Выводы:

В этой работе вы научились создавать sub-VI, проектировать иерархические схемы с использованием созданных sub-VI. Вы узнали, что входы должны быть заключены в цикл while loop, если нужно наблюдать эффект от изменения входных сигналов. Вы

также узнали, как функциональные блоки реализуются для работы в тактируемом режиме, если только функциональные блоки не заключены в структуру Single-Cycle Timed Loop. Вы научились использовать ELVIS вместе с оценочным модулем DE FPGA Board для проверки результатов проектирования. Вы протестировали проект с помощью оценочного модуля DE FPGA Board и системы NI ELVIS.