Создание проекта и управление проектом

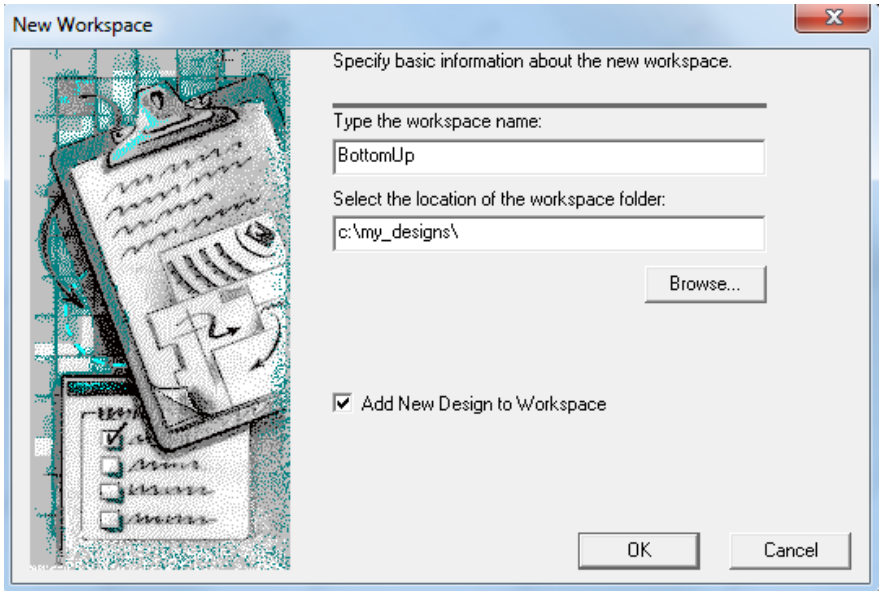
Создание тестовых модулей HDL

Концепция восходящего проектирования

- Создание новой рабочей области и проекта
  - Использование мастера **New Source File**
  - Добавление существующих файлов
  - Создание пустого проекта
- Разработка исходного кода
- Синтаксический контроль на наличие ошибок
- Верификация функционирования проекта
- Создание блок-схемы или объекта верхнего уровня

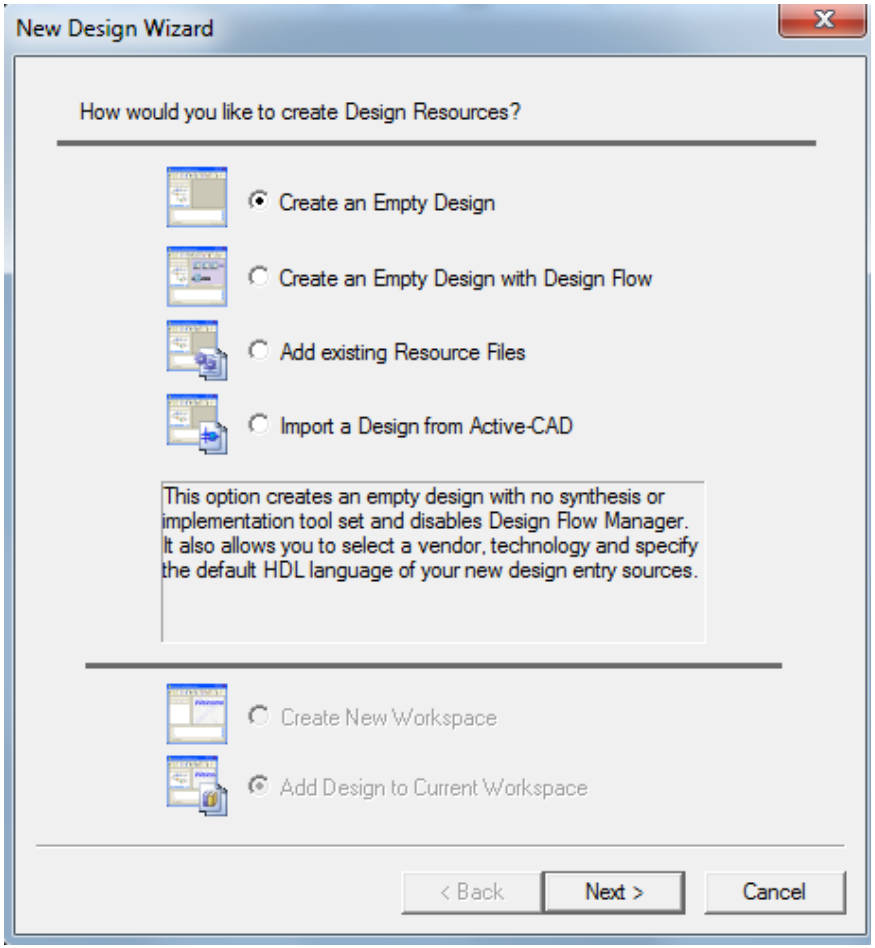
Создание проекта снизу-вверх

- *Рабочая область* это группа отдельных проектов вместе с их ресурсами (исходные, выходные файлы с результатами моделирования и др.). Рабочая область позволяет работать с несколькими проектами одновременно.
- Для создания новой рабочей области зайдите в меню **File | New** и выберете **Workspace**. В появившемся окне **New Workspace** задайте имя рабочей области (напр., BottomUp).
- Если установлен флажок **Add New Design to Workspace**, появится окно мастера **New Design Wizard**, как только кликните OK.



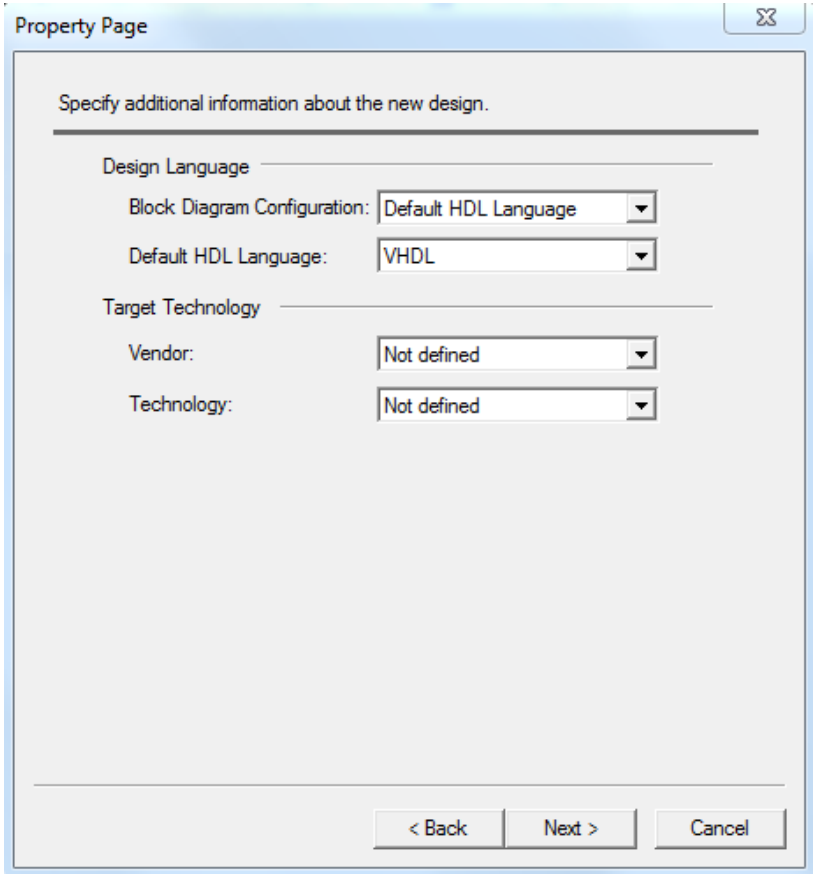
Мастер нового проекта

- Первое окно мастера **New Design Wizard** предлагает четыре варианта ввода описания проекта:
- Создать пустой проект
- Создать пустой проект вместе с маршрутом проектирования
- Добавить существующие файлы ресурсов
- Импортировать проект из Active-CAD



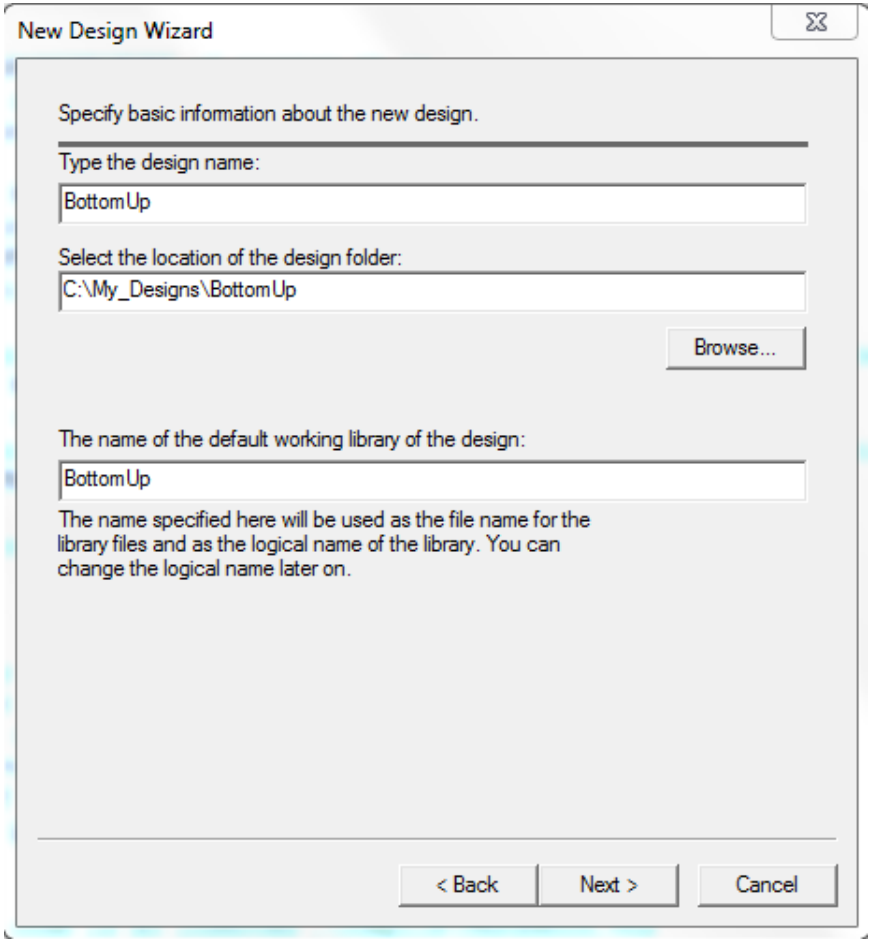
Мастер нового проекта (продолжение)

- Вторым окном является страница свойств **Property Page**. Здесь можно указать дополнительную информацию по проекту:
- Язык проектирования
  - Конфигурация блок-схемы
  - Язык HDL по умолчанию
    - VHDL, Verilog
- Целевая технология
  - Производитель
    - Actel, Altera, Lattice, др.
  - Технология



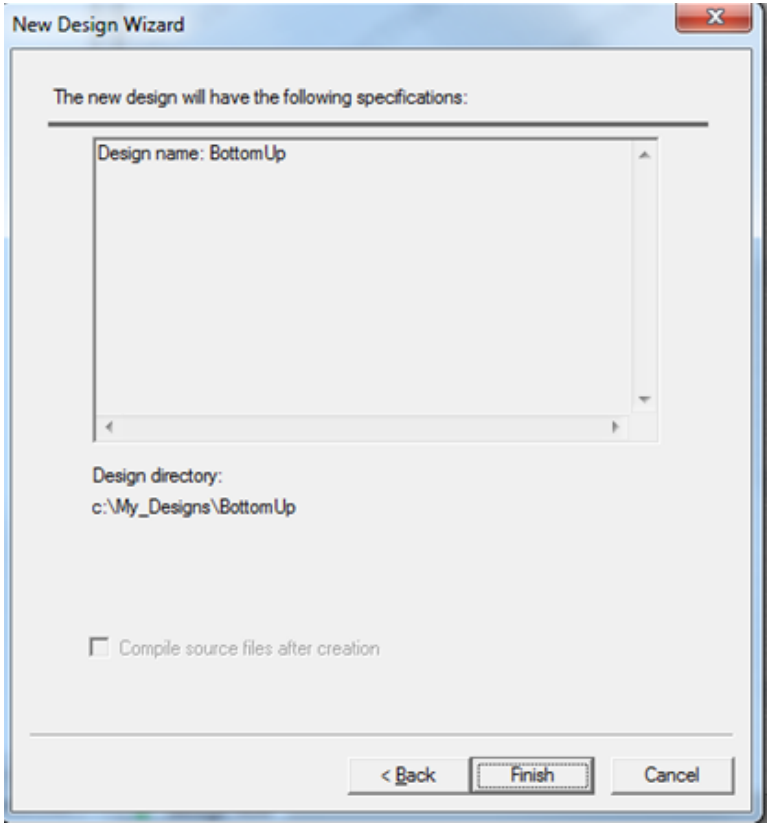
Мастер нового проекта (продолжение)

- В третьем окне вас просят указать:
- Имя вашего проекта
- Расположение папки проекта
- Название рабочей библиотеки
  - По умолчанию имя рабочей библиотеки совпадает с именем вашего проекта



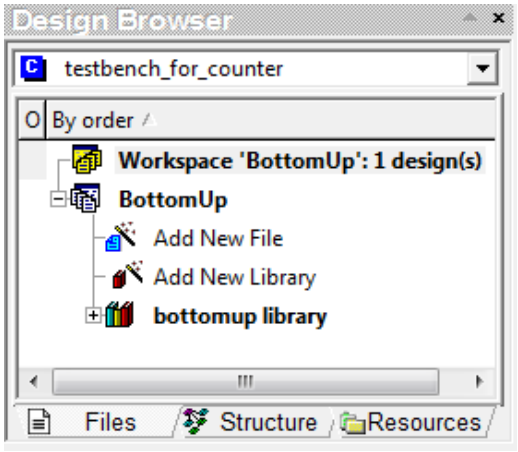
Мастер нового проекта (продолжение)

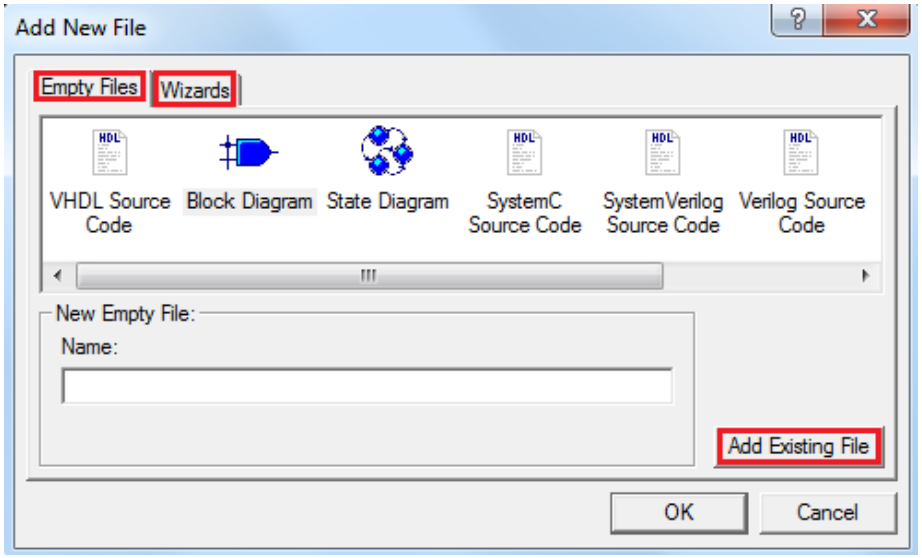
- Последнее окно показывает сводку всех характеристик, которые вы определили в предыдущих окнах.



Добавление нового файла: исходный код VHDL

- В окне браузера **Design Browser** вы можете отслеживать создание вашей рабочей области и проекта. Чтобы добавить новый или существующий файл в ваш проект, дважды кликните **Add New File**

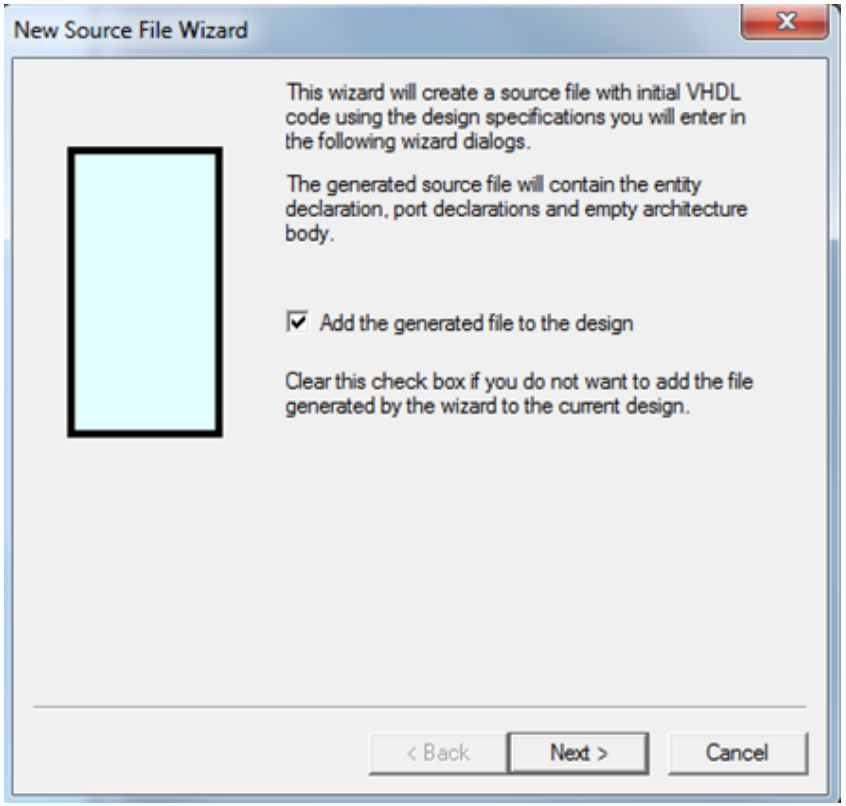




- Вы можете создать новый исходный код VHDL с нуля с помощью вкладки **Empty Files** или с помощью пошагового мастера на вкладке **Wizards**.
- Чтобы добавить существующий файл, нажмите кнопку **Add Existing File**

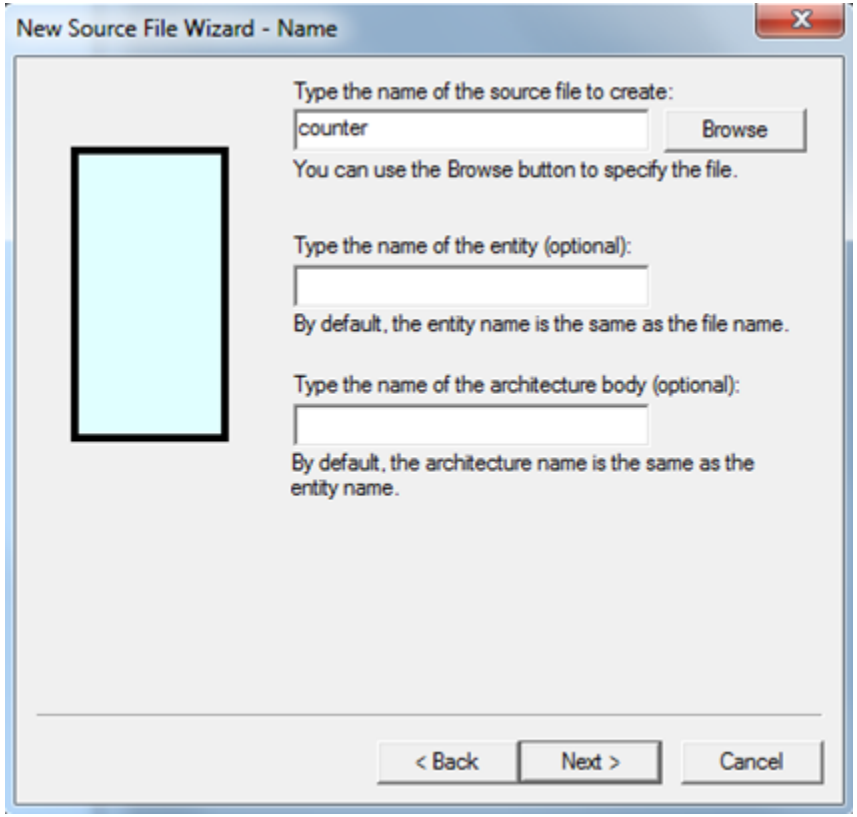
Мастер создания исходного файла

- При выборе **VHDL Source Code** на вкладке **Wizards**, откроется мастер создания нового исходного файла **New Source File Wizard**.
- Обязательно установите флажок **Add the generated file to the design**, если хотите добавить сгенерированный файл в свой текущий проект.



Мастер создания исходного файла (продолжение)

- Второе окно мастера просит вас указать:
- Имя исходного файла
  - Например, counter
- Имя интерфейса объекта (опционально)
- Имя архитектурного тела (опционально)
  - По умолчанию имя архитектурного тела совпадает с именем интерфейса объекта

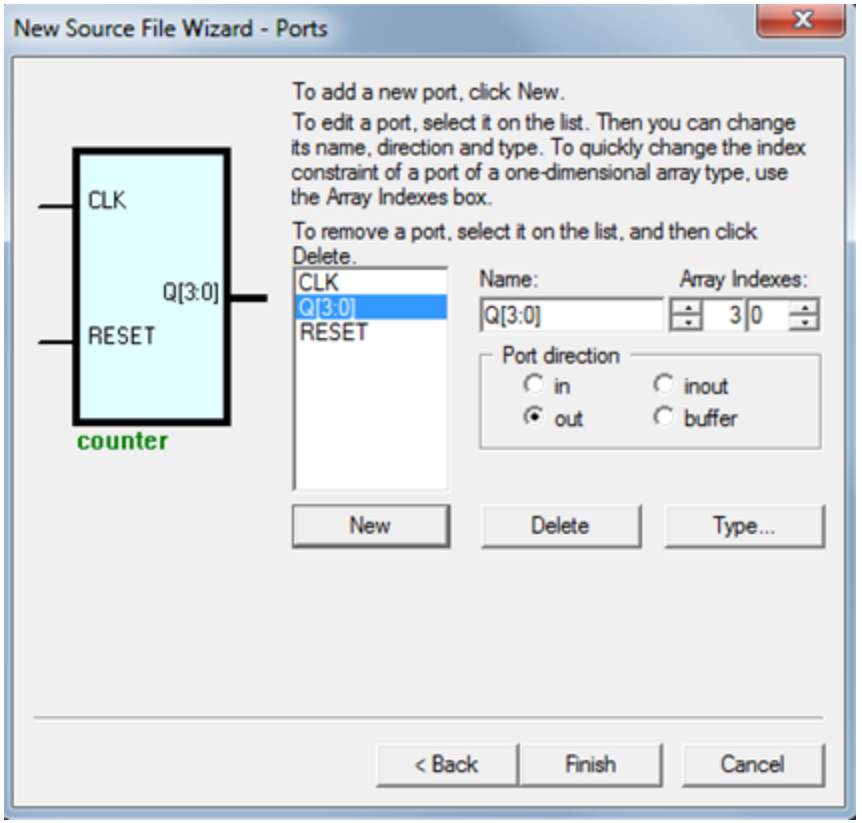


Мастер создания исходного файла (продолжение)

- В следующем окне **Ports** мастера вас попросят определить порты. Для каждого порта потребуется указать:
- Имя порта
- Направление порта
  - in, inout, out, buffer
- Array Index: ширина шины (при необходимости)

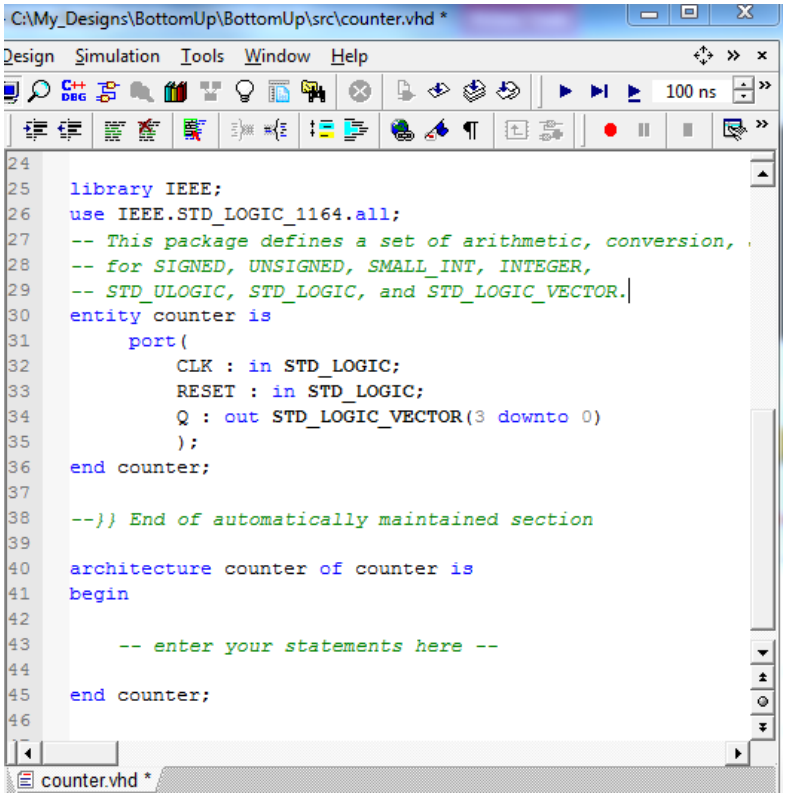
Например:

- CLK, RESET
  - Port direction: in
- Q
  - Port direction: out
  - Array Index: [3:0]



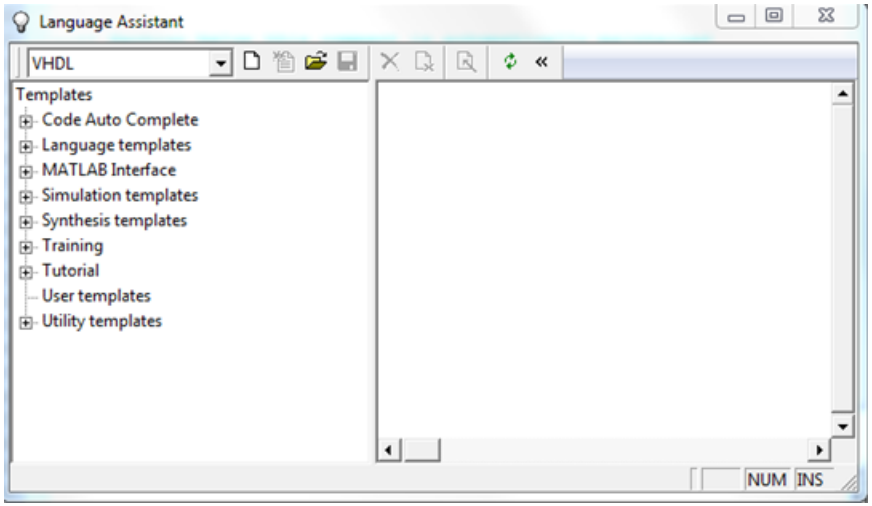
Сгенерированный исходный код VHDL

- После завершения работы мастера вы увидите скелет исходного кода VHDL в окне редактора HDL.
- Обратите внимание, что архитектурная секция вашего кода пуста. Active-HDL имеет встроенные шаблоны, которые вы можете перетаскивать в код с помощью **Language Assistant**.



Language Assistant

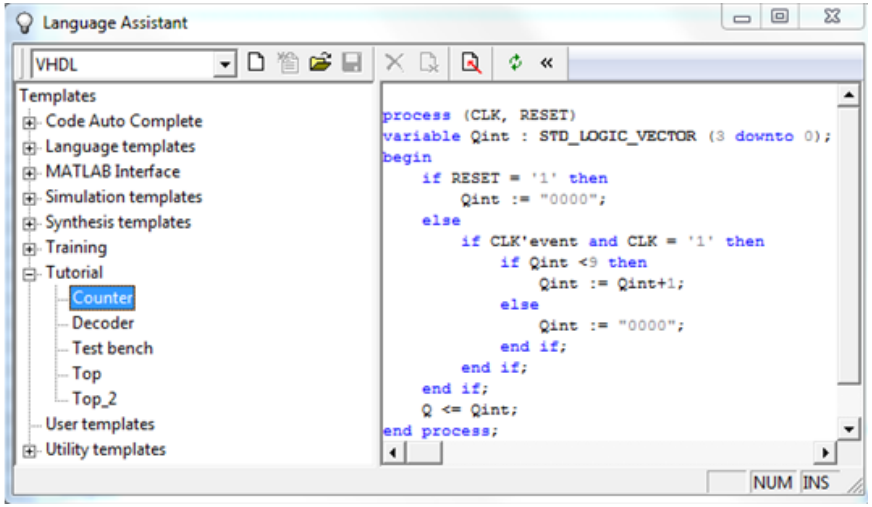
- Окно **Language Assistant** содержит шаблоны часто используемых моделей, пользовательских моделей, и конструкций VHDL/ Verilog.

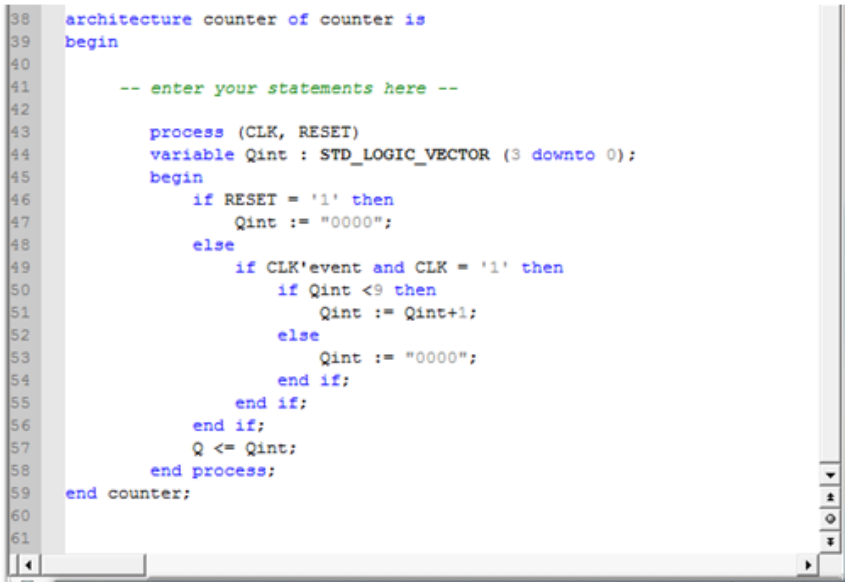


- Чтобы открыть **Language Assistant** перейдите в **Tools | Language Assistant** или выберете значок **Language Assistant**  на панели инструментов HDL Editor.

Language Assistant (продолжение)

- Чтобы завершить секцию архитектуры вашего кода:
- Разверните раздел **Tutorial**
- Выберете **Counter**
- Перетащите **Counter** в ту часть кода, которая гласит:
  - *Enter your statements here*





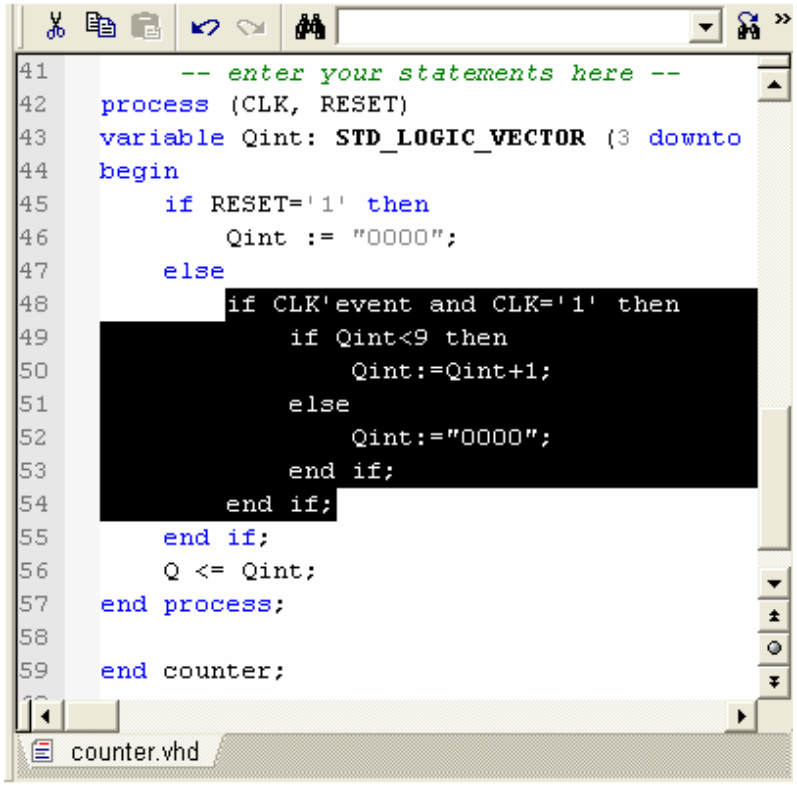
Функции редактора HDL

- Редактор HDL обладает многими функциями, которые позволяют пользователю эффективно управлять своим кодом:
- Раскраска ключевых слов для языков VHDL, Verilog/ SystemVerilog и C/C++/Handel-C
- Комментирование выбранных частей кода
- Создание групп из выделенных блоков
- Автоматическое создание структуры для исходного кода
- Автоформатирование для исходного кода
- Установка закладок по всему коду для удобства навигации
- Выделение некорректных конструкций после компиляции
- Нахождение и замена заданных строк
- Запись и воспроизведение сочетаний клавишей и макросов

Разметка блоков

- Вы можете выбирать блоки кода для выполнения различных функций (напр. отступ, комментарий, создание структуры и т.д.)

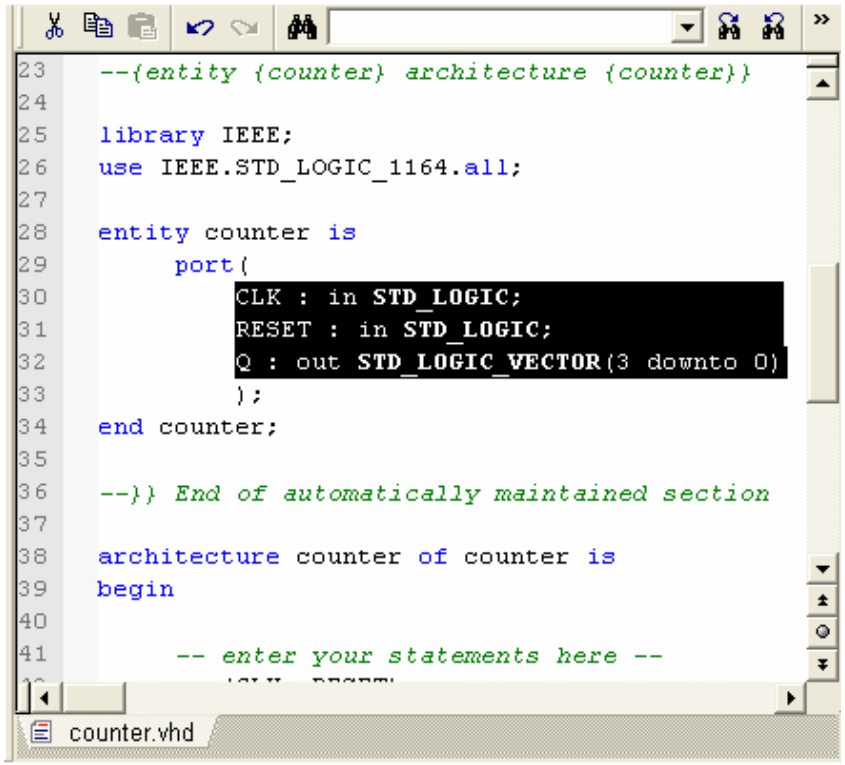
- Есть два способа выбрать блок:
  - Удерживая левую кнопку мыши перетащите курсор через текст для выделения желаемого фрагмента.
  - Удерживайте клавишу **Shift** и используйте клавиши со стрелками для выделения нужного фрагмента.



Примечание: Целые слова могут быть выбраны одновременным нажатием **Ctrl** и **Shift**

Разметка столбцов

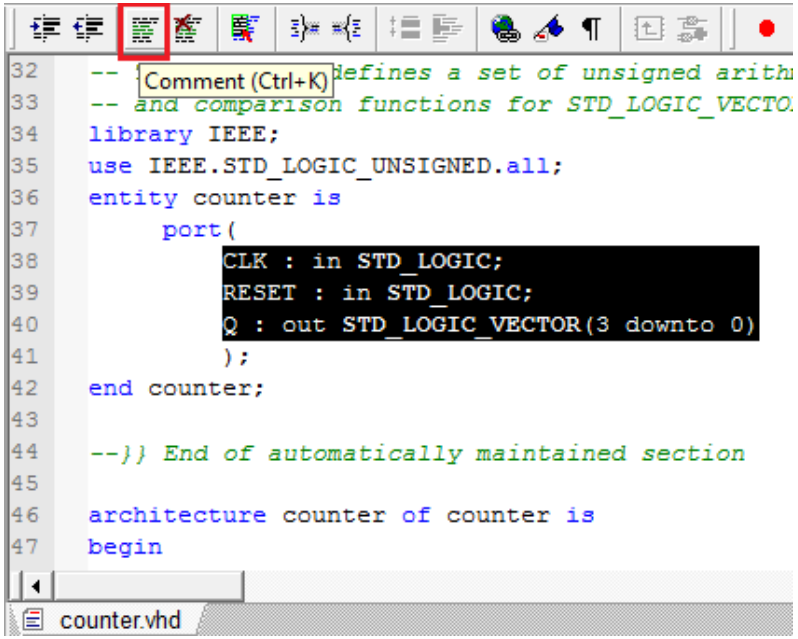
- Вы также можете выбрать столбцы в коде, если хотите избежать лишних пробелов.
- Есть два способа пометить столбцы:
  - Удерживая нажатой клавишу **Alt**, нажмите левую кнопку мыши и переместите курсор, чтобы выделить нужный столбец.
  - Нажмите **Alt+C** и используйте клавиши **Shift** и **стрелки** для выделения нужного столбца.



Комментирование блоков/столбцов

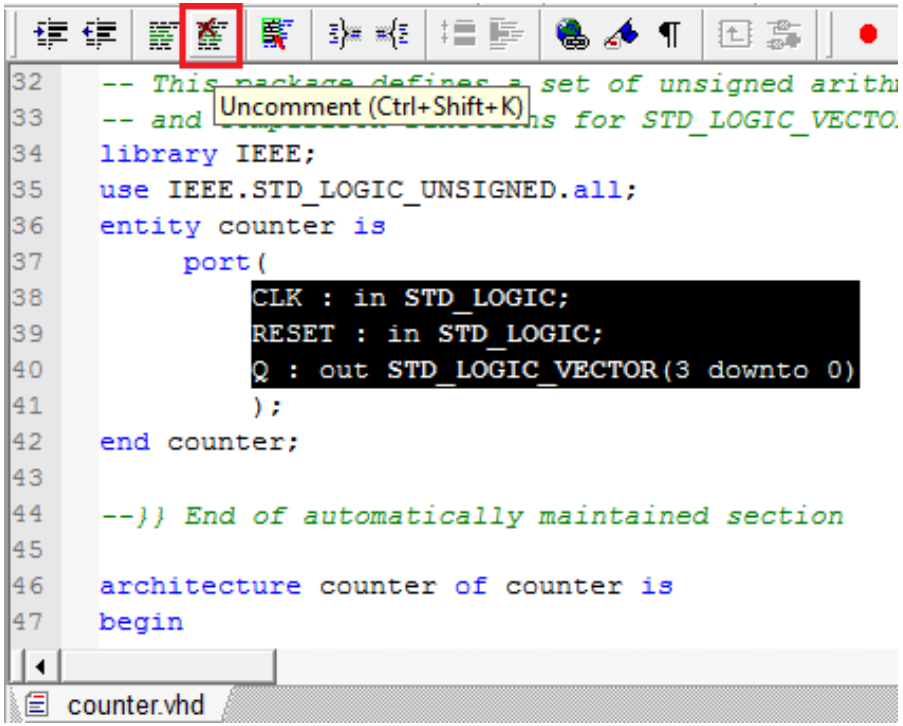
- Иногда требуется комментировать целые блоки/ столбцы кода или для отладки, или когда вы не уверены в эффективности такого блока.

- Существует два способа комментирования блоков/ столбцов:
  - Выберете нужный блок/столбец и щёлкните значок комментария **Comment**  на панели инструментов HDE.
  - Выберете нужный блок/столбец и нажмите **Ctrl+K**



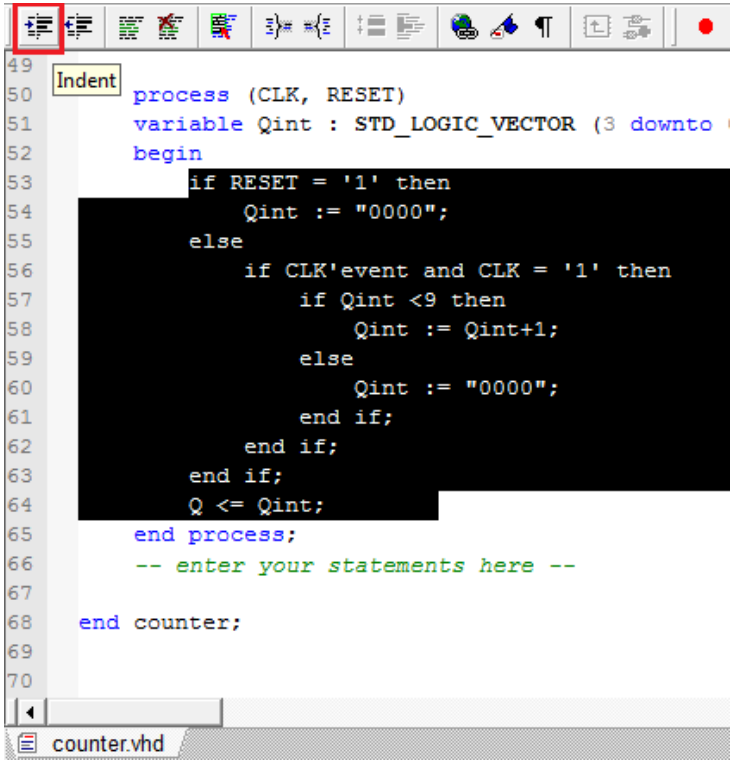
Раскомментирование блоков/столбцов

- Есть два способа раскомментировать блоки/столбцы:
  - Выберете нужный блок/столбец и используйте значок **Uncomment** на панели инструментов HDE.
  - Выберете нужный блок/столбец и нажмите **Ctrl+Shift+K**



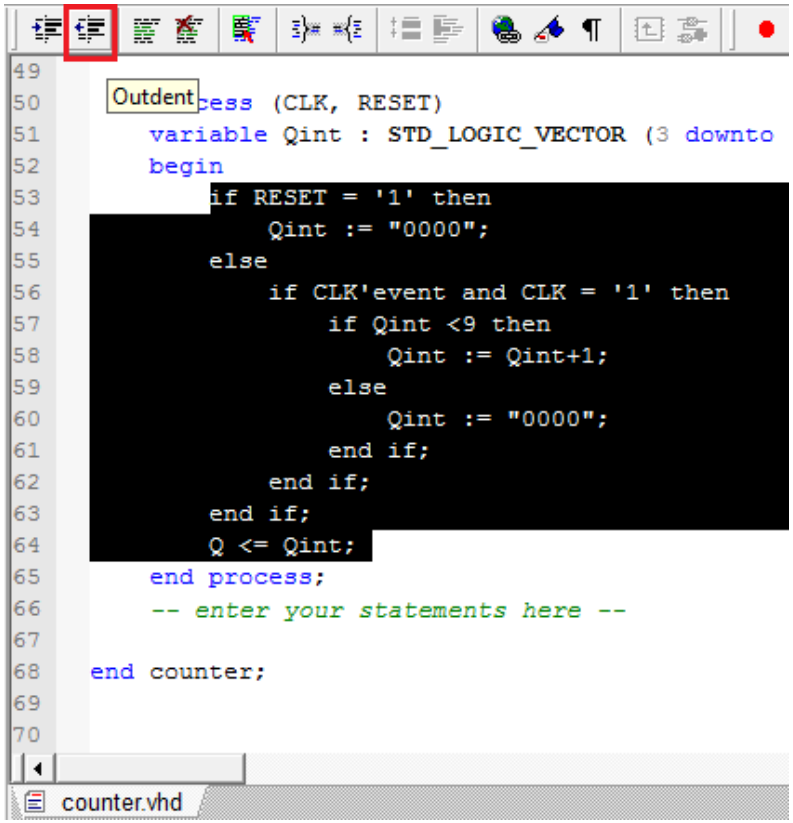
Отступ

- Отступ некоторых частей в коде является предпочтением пользователя в эстетических целях.
- Есть два способа сделать отступ:
  - Выберите нужный блок кода и используйте значок **Indent** на панели инструментов HDE.
  - Выберите нужный блок кода и нажмите кнопку **Tab**.



Втяжка

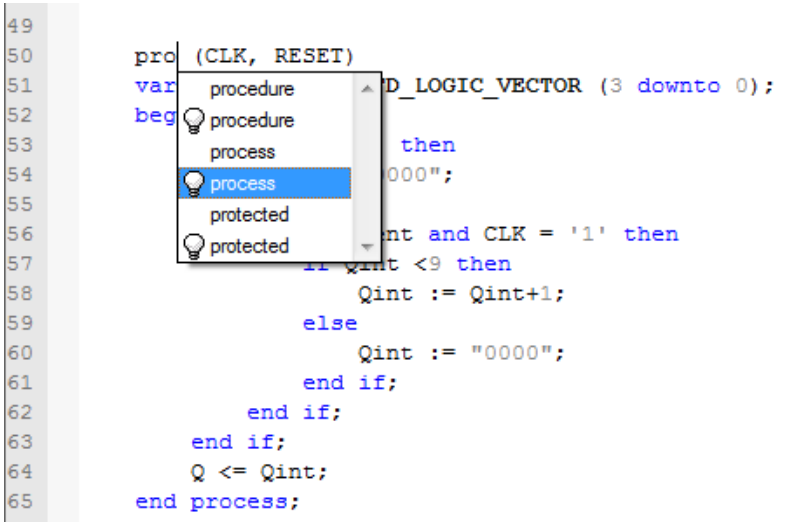
- Как и отступ, втяжка **Outdent** является предпочтением пользователя в эстетических целях.
- Есть два способа сделать втяжку:
  - Выберите нужный блок кода и нажмите значок **Outdent**  на панели инструментов HDE.
  - Выберите нужный блок кода и нажмите кнопку **Shift+Tab**.



Примечание: Вы можете изменить размер табуляции в категории **HDL Editor** в разделе **Tools | Preferences**

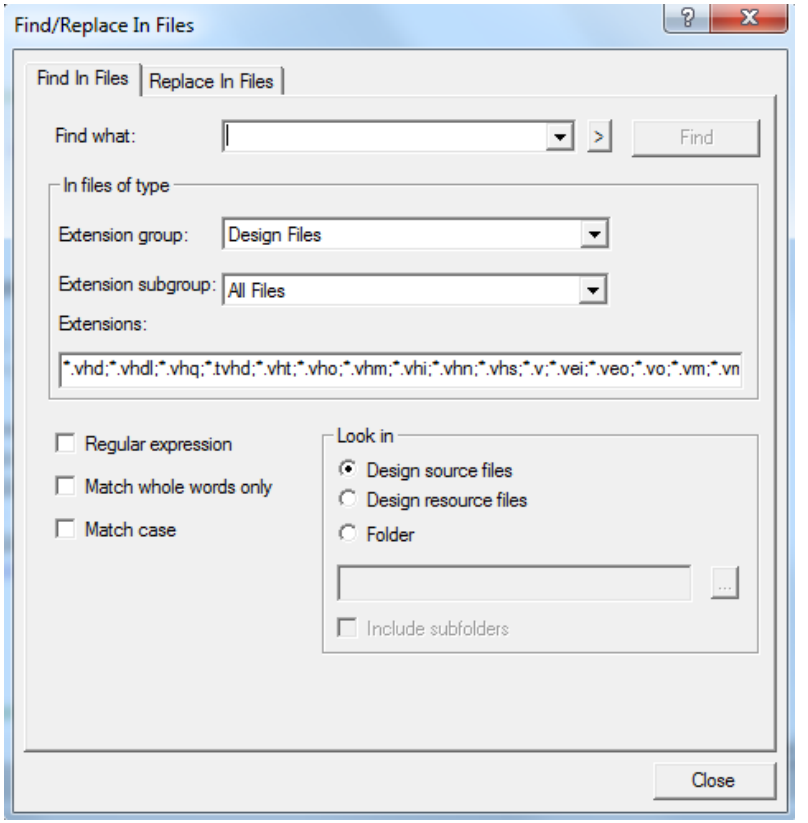
Автозаполнение

- Редактор HDL автоматически подставляет ключевые слова VHDL и Verilog на основе введённых начальных букв.
- Когда вы начинаете печатать, появляется выпадающий список. Вы можете прокручивать список ключевых слов с помощью клавиш со стрелками.
- Чтобы выбрать ключевое слово нажмите **enter**.



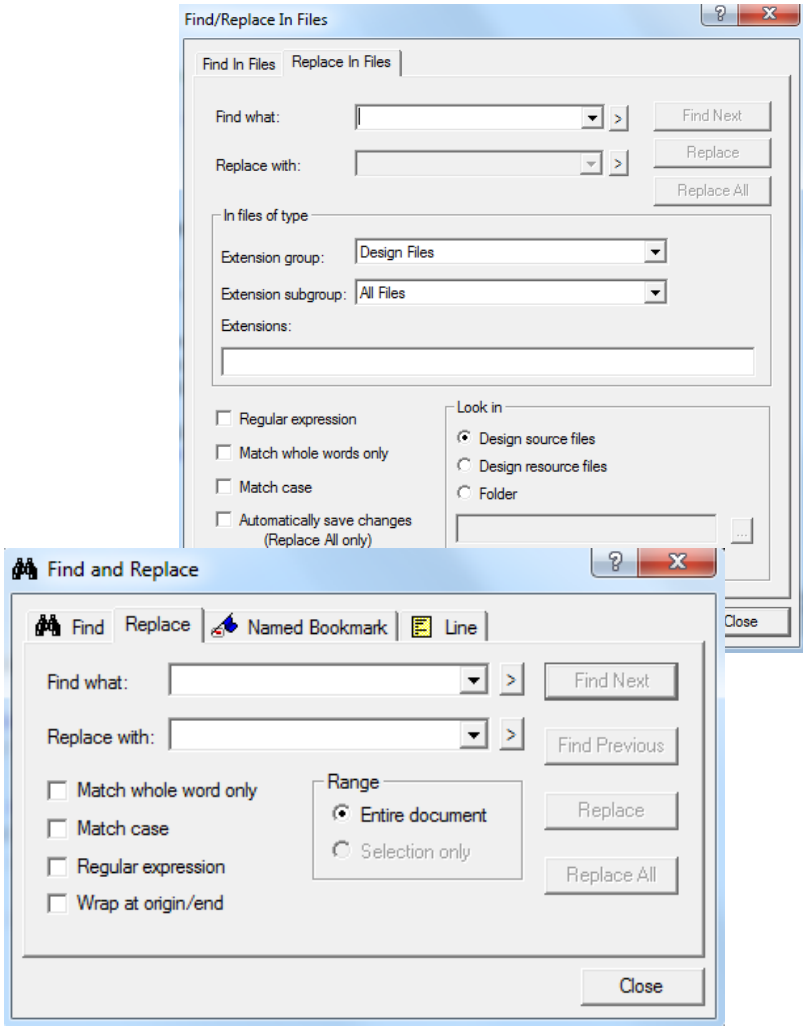
Поиск

- Функция поиска **Find** позволяет искать файлы по определенной строке. Это полезно для целей отладки. Есть два вида поиска Find:
- **Find**
  - Поиск строки в одном файле
- **Find in Files…**
  - Поиск строки в нескольких файлах одновременно
- Обе функции можно найти в меню поиска **Search**



Замена

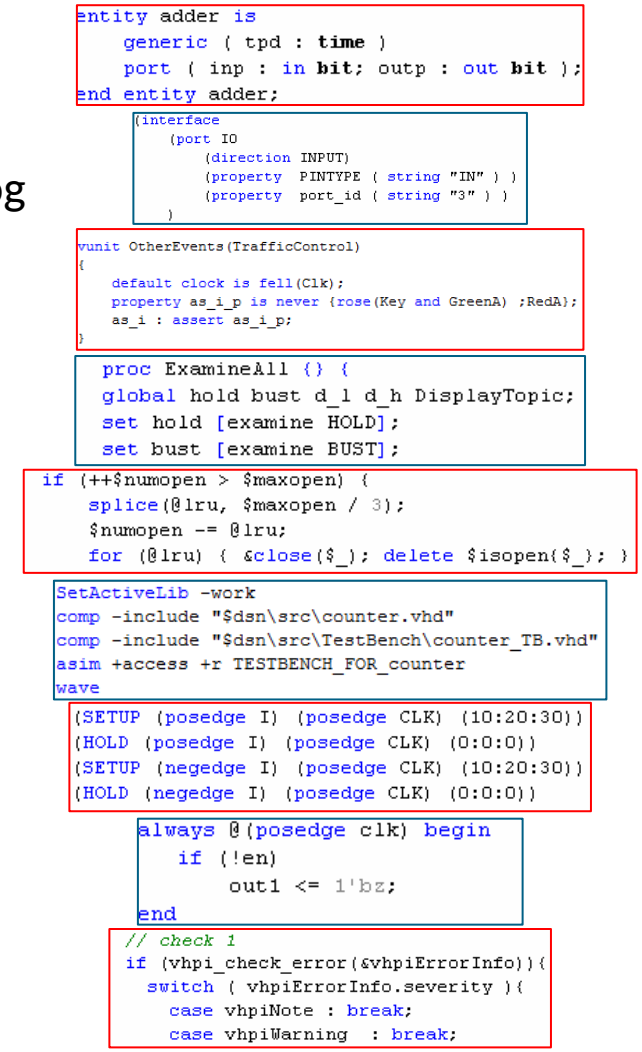
- Функция замены **Replace** позволяет вам искать определённую строку и заменять один экземпляр указанной строки или ВСЕ её экземпляры чем-то другим.
- Аналогично **Find**, **Replace** также может использоваться для поиска как в одном файле, так и в нескольких файлах.
- Функцию замены **Replace** можно найти в меню поиска **Search**.



Выделение синтаксиса

- Редактор HDL поддерживает выделение синтаксиса для следующих типов файлов:

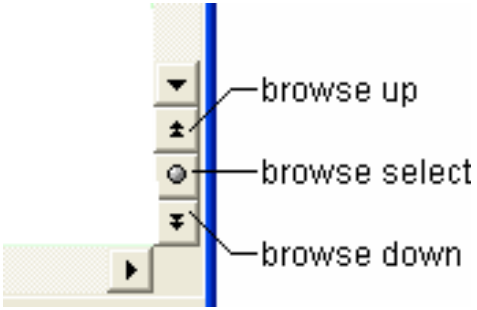
- VHDL
- Verilog/SystemVerilog
- PSL/OVA
- Tcl
- Perl
- Macro
- SDF
- EDIF
- C++



Навигация

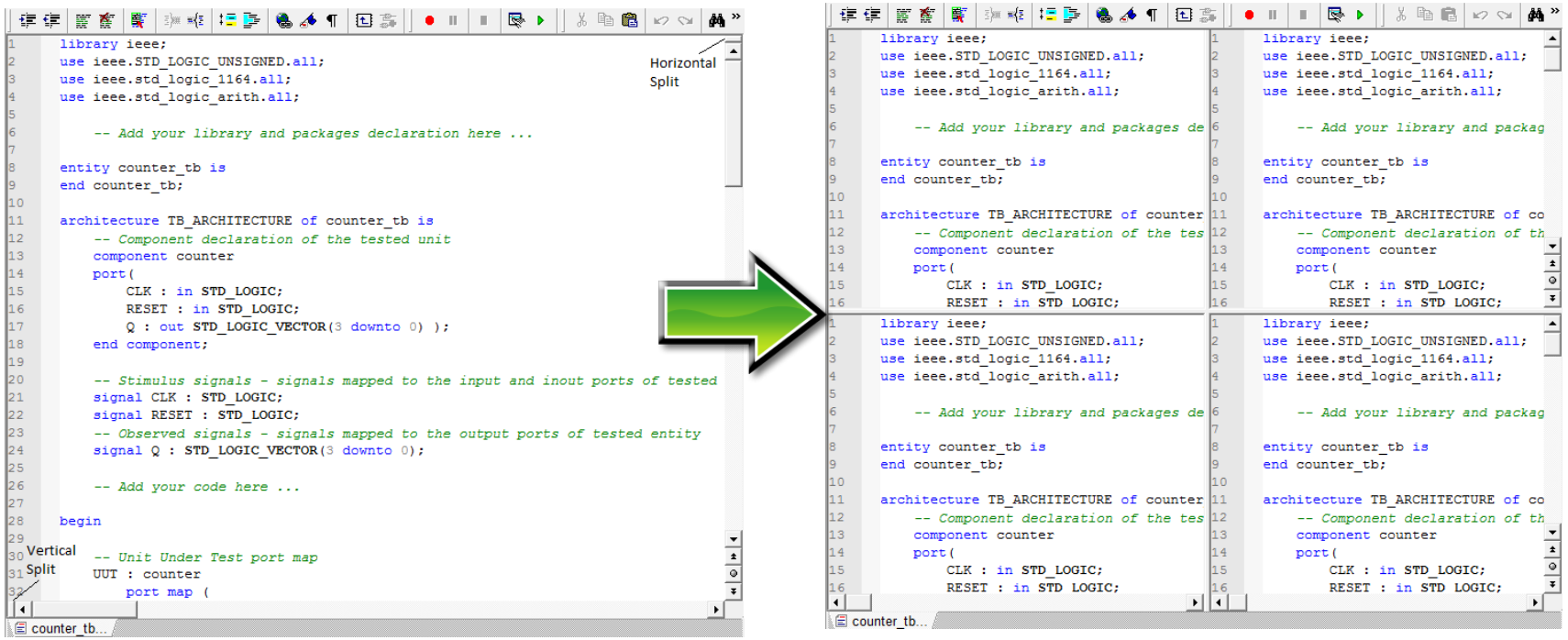
- Редактор HDL предоставляет ряд функций, предназначенных для навигации по исходным документам в среде Active-HDL.
- Вы можете просматривать свой код по:
  - Страницам
  - Закладкам
  - Именованным закладкам
  - Связям
  - Контрольным точкам

Browse Options



Навигация (продолжение)

- Вы можете создавать горизонтальные или вертикальные разделенные экраны для навигации по различным частям одного и того же исходного кода.



Закладки

- Закладки помогают облегчить навигацию по длинным документам. Их можно размещать по всему исходному коду и переходить от одной закладки к другой.
- Чтобы вставить закладку:
  - Выберете **Toggle Bookmark** или **Ctrl+F2** 
- Для перемещения межлу закладками:
  - Выберете **Next Bookmark** или **F2** 
  - Выберете **Previous Bookmark** или **Shift+F2** 
- Для удаления всех закладок:
  - Выберете **Clear all bookmarks** 

Примечание: Все команды закладок можно найти в меню поиска Search

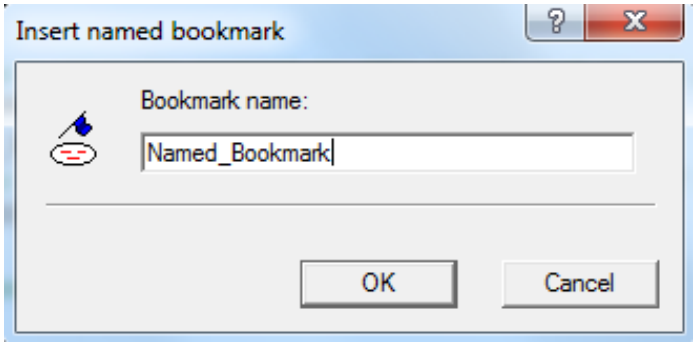
Именованные закладки

- Именованные закладки аналогичны закладкам Toggle, помогая облегчить навигацию по исходному коду. Разница в том, что именованные закладки кодируются специальными строками, вставляемыми непосредственно в текст документа. Эти строки называются *кодами закладок*.
- Например, закладка с именем *jump* будет реализована следующей строкой:
  - --<A NAME="Jump">
- Редактор HDL не отображает коды закладок напрямую. Вместо этого названия закладок выделяются другим цветом. Аналогично тому, как отображается комментарий.
  - --Named bookmark in HDL code:
  - -- Jump

Именованные закладки (продолжение)

- Любые коды закладок, которые появляются за пределами комментариев, редактором HDL игнорируются и отображаются как истинный код.

- Чтобы поместить именованную закладку:
- Выберете значок **Insert Named Bookmark** 
- Укажите имя закладки




```
20      -- Stimulus signals - signals mapped to the input and inou
21      signal CLK : STD_LOGIC;
22      signal RESET : STD_LOGIC;
23      -- Observed signals - signals mapped to the output ports o
24      signal Q : STD_LOGIC_VECTOR(3 downto 0);--Named_Bookmark
25
```

Авто отступ и смарт отступ

- Редактор HDL предоставляет две функции, предназначенные для внесения отступов в редактируемом коде:
- Авто отступ
  - Когда вы нажимаете **Enter**, чтобы начать новую строку, редактор автоматически вставляет символы табуляции или пробелы в новой строке для выравнивания с первым символом в предыдущей строке.
- Смарт отступ
  - Когда вы нажимаете **Enter**, чтобы начать новую строку, редактор автоматически вставляет символы табуляции или пробелы в новой строке для выравнивания последовательных конструкций HDL
- Обеими опциями можно управлять в диалоге настроек **Preferences**.

Автоформатирование

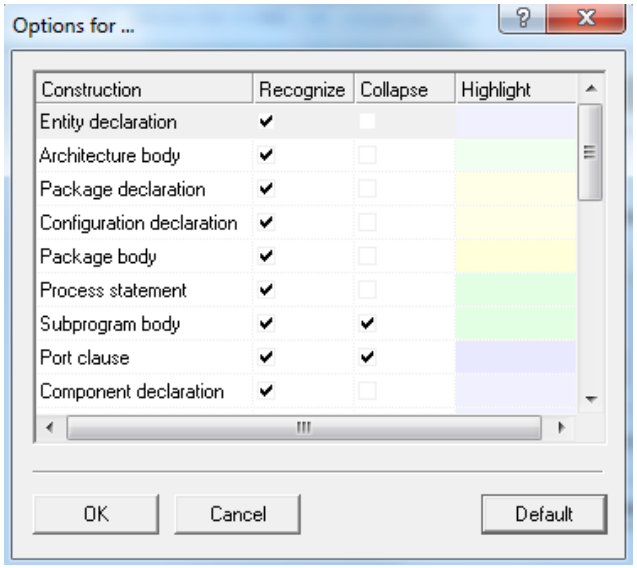
- Если вы находите, что ваш код не выровнен должным образом, можно использовать функцию **Autoformat Text**  Эта функция анализирует код и делает отступ последовательных строк текста на основе тех же принципов, что и опция **Smart Indent**.

```
22
23  architecture Counter of Counter is
24      begin
25      -- <<enter your statements here>>
26
27  process (CLK, RESET)
28  variable Qint: STD_LOGIC_VECTOR (3 downto 0);
29  begin
30  if RESET='1' then
31  Qint := "0000";
32  else
33  if CLK'event and CLK='1' then
34  if Qint<9 then
35  Qint:=Qint+1;
36  else
37  Qint:="0000";
38  end if;
39  end if;
40  end if;
41  Q <= Qint;
42  end process;
43  end Counter;
44
```

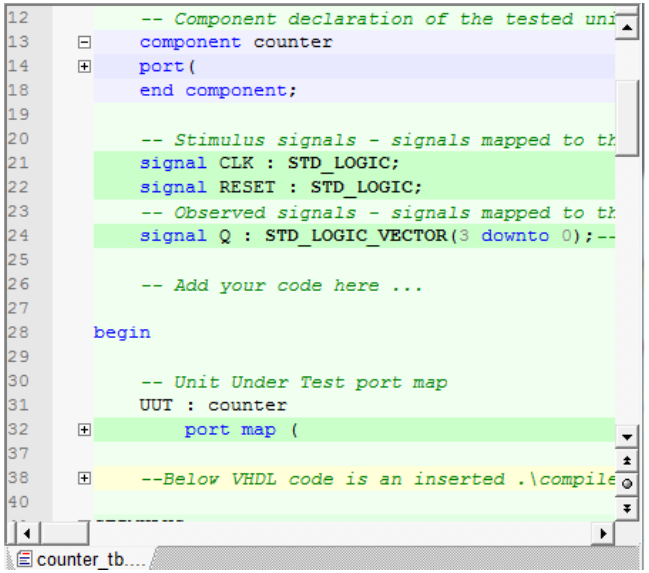
```
23  architecture Counter of Counter is
24      begin
25      -- <<enter your statements here>>
26
27      process (CLK, RESET)
28          variable Qint: STD_LOGIC_VECTOR (3 do
29          begin
30          if RESET='1' then
31              Qint := "0000";
32          else
33              if CLK'event and CLK='1' then
34                  if Qint<9 then
35                      Qint:=Qint+1;
36                  else
37                      Qint:="0000";
38                  end if;
39              end if;
40          end if;
41          Q <= Qint;
42      end process;
43  end Counter;
44
```

Генерация структуры

- Вы можете разделить ваш исходный код на группы, используя функцию **Generate Structure** . Данная функция группирует части кода в соответствии с синтаксисом HDL.
- Вы можете настроить эту команду в **Tools|Preferences|Editors|HDL Editor|Languages|<language>|Document**



Примечание: Можно настроить, какие конструкции HDL должны быть сгруппированы и цвет для каждой конструкции.



Структура текста

- Когда генерируете структуру своего кода, вы можете использовать кнопки + - , чтобы развернуть и свернуть группы операторов HDL.
- Если не хотите, чтобы весь исходный код был сгруппирован, можете создать собственные меньшие групповые структуры.
  - Выберите часть кода и нажмите кнопку **Create Group** 

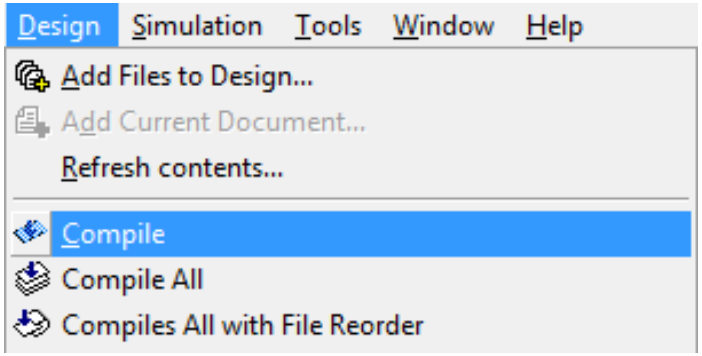
```
        if RESET='1' then
            Qint := "0000";
        else
            if CLK'event and CLK='1' then
                if Qint<9 then
                    Qint:=Qint+1;
                else
                    Qint:="0000";
                end if;
            end if;
        end if;
        Q <= Qint;
    end process;
end Counter;
```

- Чтобы вернуться к исходному макету документа, нажмите кнопку **Remove Groups** 

Компиляция проекта

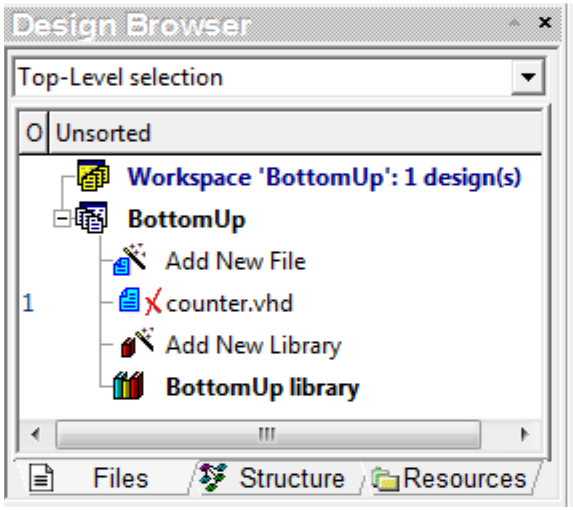
- Active-HDL позволяет вам компилировать исходные файлы проекта тремя способами:
- Compile
  - Обычно для отдельных файлов
- Compile All
  - Для нескольких файлов в папке или во всём проекте
- Compile All with File Reorder
  - Переупорядочивает исходные файлы, чтобы обеспечить правильную последовательность компиляции модулей

Примечание: Все опции могут быть найдены в меню **Design**.



Статус компиляции

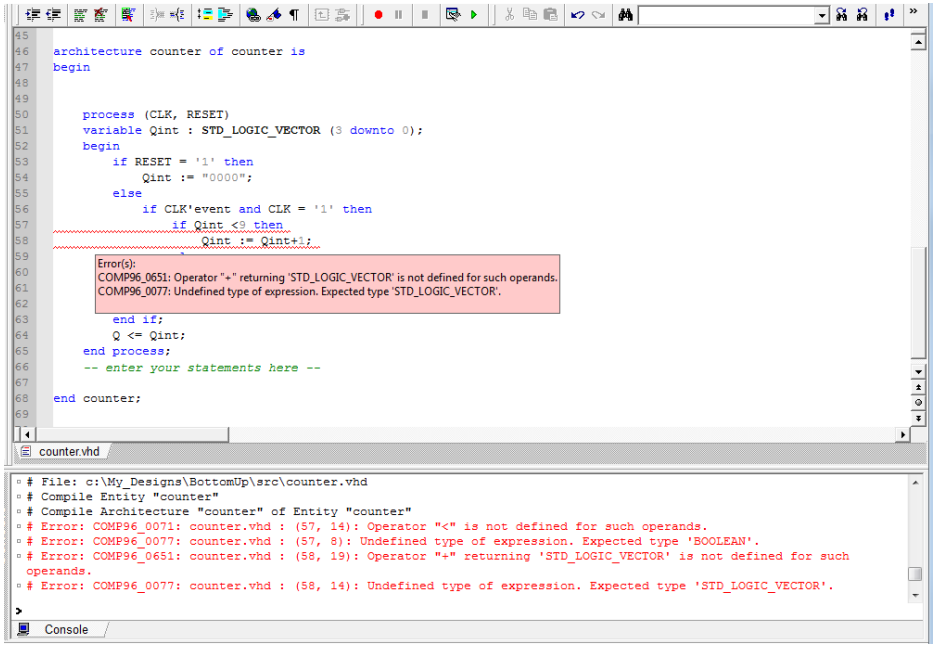
- Статус каждого скомпилированного файла будет показан в браузере проекта и может иметь одно из следующих значений:
- Во время последней компиляции произошли ошибки
- Во время последней компиляции были предупреждения
- Не скомпилировано или изменено после последней компиляции
- Успешно скомпилировано



Отслеживание ошибок

- В окне редактора HDL вы сможете отслеживать любые ошибки, которые произошли во время компиляции кода (подчеркнуты красным).

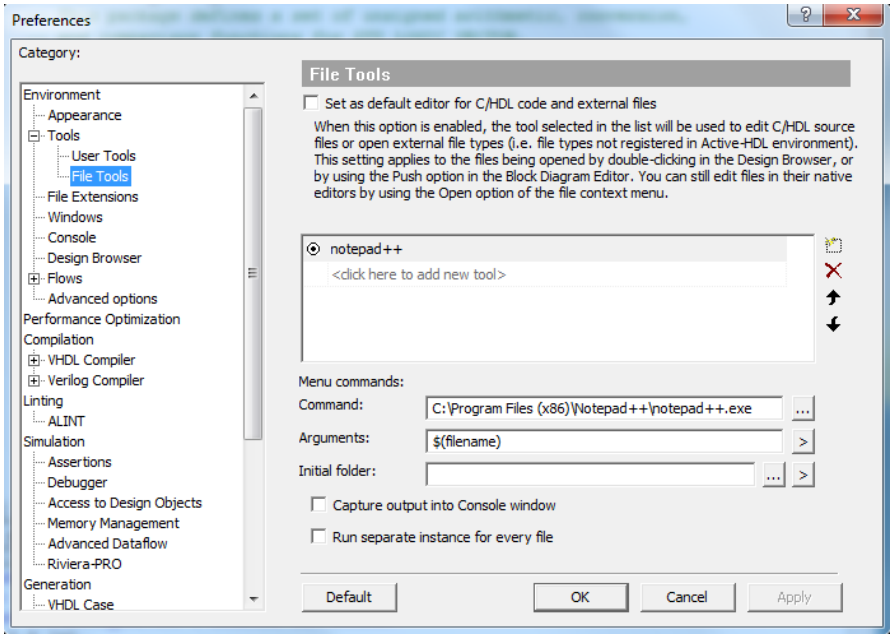
- В окне консоли отобразятся все ошибки с краткими описаниями. Если вы дважды щелкнете на сообщении об ошибке, то попадете прямо на строку кода с ошибкой.



- Если навести курсор на подчеркнутую ошибку в редакторе HDL, вы также увидите её краткое описание.

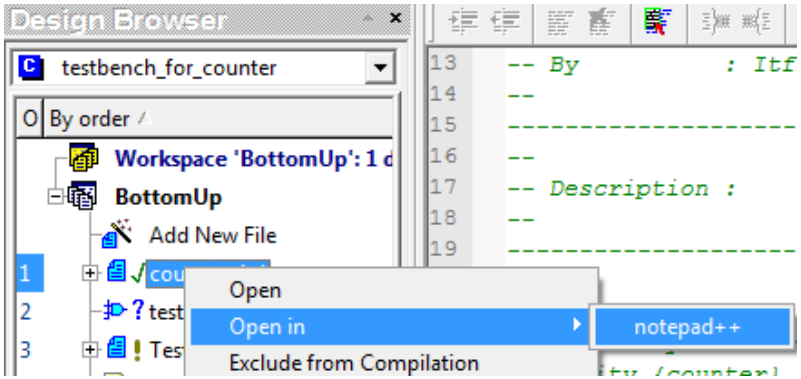
Открытие файлов во внешних редакторах HDL

- Active-HDL позволяет вызывать внешние редакторы из своей среды
- Для настройки:
  - В диалоговом окне **Preferences** перейдите в **Environment|Tools|File Tools**
  - Введите имя редактора, где написано **<click here to add new tool>**
  - В поле **Command** укажите расположение файла .exe для внешнего редактора.



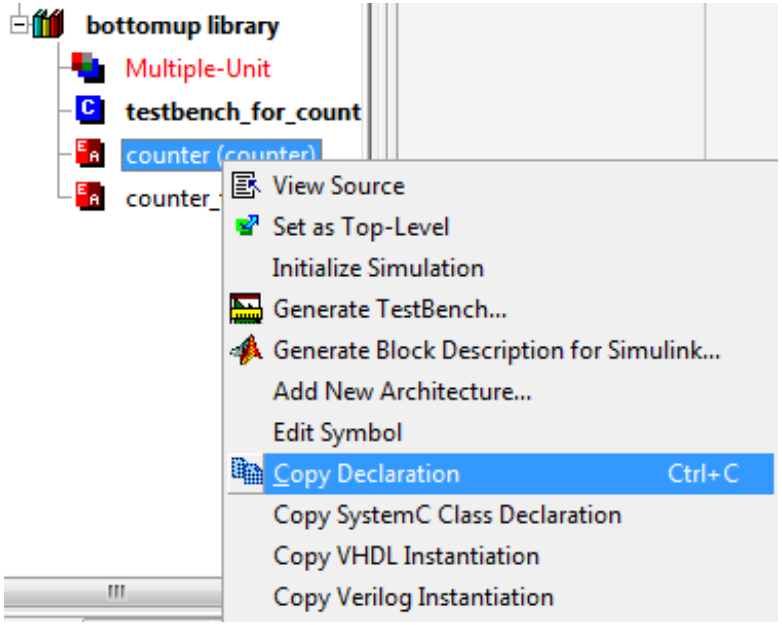
Внешние редакторы HDL (продолжение)

- Чтобы открыть исходный файл в заданном вами внешнем редакторе:
  - Щелкните правой кнопкой мыши на исходном файле в вашем браузере проекта **Design Browser**
  - Выберете **Open in**
  - Выберете редактор для открытия вашего файла



Декларация компонентов

- Active-HDL предоставляет утилиты для ускорения декларации и инстанцирования компонентов. Вы можете скопировать декларацию компонента из рабочей библиотеки или библиотеки в окне диспетчера библиотек **Library Manager**.
  - Разверните содержимое библиотеки в браузере **Design Browser**, щелкните правой кнопкой мыши на компоненте и выберите **Copy Declaration**.
  - Перейдите в окно редактора HDL и вставьте объявление, щелкнув правой кнопкой мыши и выбрав **Paste**.



Примечание: Это только скопирует декларацию компонента. Вы должны будете самостоятельно установить назначение портов и обобщающих констант объекта.

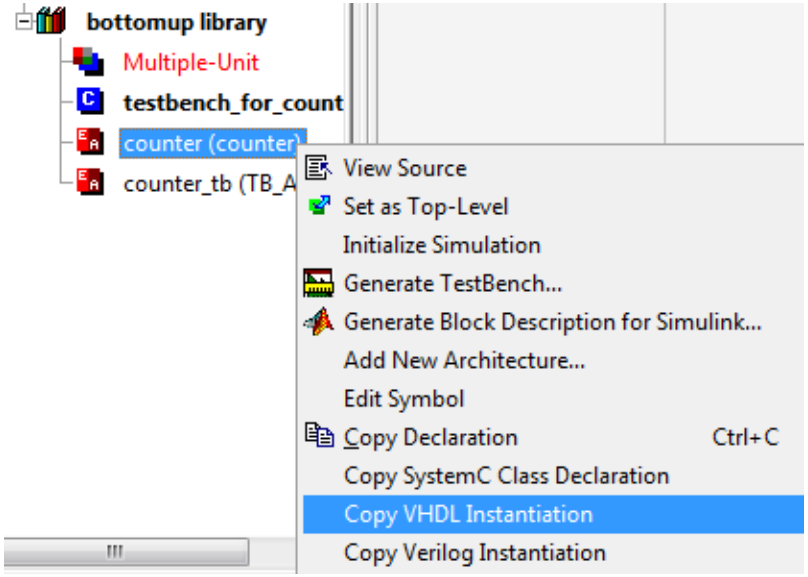
Инстанцирование компонентов

- Active-HDL позволяет работать с многофайловыми проектами.
- Таким образом, вы можете создавать требуемые модели в отдельных файлах и проверять их по отдельности, а не помещать их в один большой файл проекта.
- Создав объект верхнего уровня, вы можете проверить функциональность всего проекта. Чтобы сделать это, вы должны инстанцировать компоненты проекта.
- Инстанцирование компонентов похоже на подключение аппаратного компонента к гнезду на плате.

```
23  architecture MODULATOR of Modulator is
24
25  ---- Signal declarations used on the diagram ----
26
27  signal COS1 : real ;
28  signal SAW1 : real ;
29  signal SIN1 : real ;
30
31  ---- Component declarations -----
32
33  component COSINUSGENERATOR
34    port (
35          CLK : in BIT;
36          CosEnable : in BIT;
37          CosFreq : in INTEGER;
38          COS1 : out REAL
39    );
40  end component ;
41
42  component MULTIPLIER
43    port (
44          IN1 : in REAL;
45          IN2 : in REAL;
46          IN3 : in REAL;
47          clk : in bit;
48          out1 : out real
49    );
50  end component ;
```

Инстанцирование компонентов (продолжение)

- Active-HDL также предоставляет утилиту для ускорения инстанцирования компонентов. Вы можете получить экземпляр компонента для VHDL или Verilog.

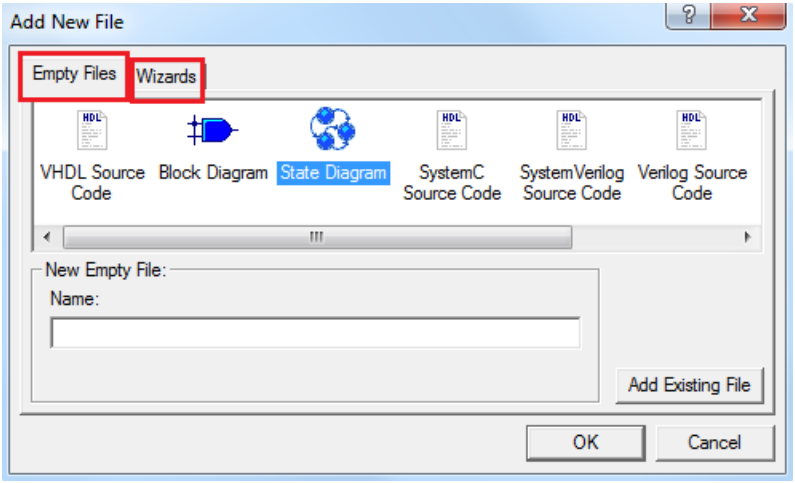


- Разверните содержимое библиотеки в браузере **Design Browser**, щелкните правой кнопкой мыши на компоненте и выберите **Copy Instantiation** для соответствующего исходного файла VHDL или Verilog.
- Перейдите в окно редактора HDL и вставьте экземпляр, щелкнув правой кнопкой мыши и выбрав **Paste**.

Примечание: Метка экземпляра компонента и фактические данные, сопоставленные с портами экземпляра, можно настроить в окне **Preferences**.

Создание диаграмм состояний

- Чтобы создать диаграмму состояний:
  - Дважды щёлкните **Add New File** в браузере **Design Browser**.
  - Вы можете создать автомат состояний без какой-либо помощи, выбрав **State Diagram** на вкладке **Empty Files**.
  - Или вы можете создать автомат состояний с пошаговым руководством, выбрав **State Diagram** на вкладке **Wizards**.



Создание блок-схем

- Чтобы создать блок-схему:
  - Дважды щёлкните **Add New File** в браузере **Design Browser**.
  - Вы можете создать блок-схему без какой-либо помощи, выбрав **Block Diagram** на вкладке **Empty Files**.
  - Или вы можете создать блок-схему с пошаговым руководством, выбрав **Block Diagram** на вкладке **Wizards**.