Оглавление

Описание. 2

Язык. 3

Используемое ПО. 4

Структура файлов проекта. 5

Общие требования к стилистике исходных кодов. 6

# Описание.

В данном документе собраны общие рекомендации при разработке проектов для FPGA.

# Язык.

На данный момент основным языком является **Verilog**. Применение **VHDL** возможно, но нежелательно.

При работе с САПР **Xilinx Vivado** возможно применение языка **SystemVerilog** или его элементов. Например, при большом количестве однотипных портов удобнее их группировать в многомерные массивы.

# Используемое ПО.

Для проектирования используется связка нескольких программных продуктов:

1. **Aldec Active-HDL** версии 9.3 или новее – используется для написания кода (удобный редактор) и его первоначальной функциональной отладки.

Ссылка для скачивания:

<https://www.aldec.com/en/downloads>

2. **Xilinx Vivado** 2015.2 или новее – используется для синтеза и имплементации проекта, для генерации IP-ядер, для задания констрейнов. Применяется для FPGA Xilinx начиная с 7 серии.

Ссылка для скачивания:

<http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools.html>

3. **Xilinx ISE** 14.7 - используется для синтеза и имплементации проекта, для генерации IP-ядер, для задания констрейнов. Применяется для FPGA Xilinx до 6 серии включительно.

Ссылка для скачивания:

<http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html>

4. **Lattice Diamond** 3.4.1.213 или новее - используется для синтеза и имплементации проекта, для генерации IP-ядер, для задания констрейнов. Применяется для PLD Lattice.

Ссылка для скачивания:

<http://latticesemi.com/Products/DesignSoftwareAndIP/FPGAandLDS/LatticeDiamond.aspx>

5. **MATLAB** – используется для проработки и косимуляции DSP-алгоритмов.

Ссылка для скачивания:

<http://www.mathworks.com/downloads/web_downloads>

# Структура файлов проекта.

Независимо от текущего кристалла проекты имеют одинаковую структуру. В качестве примера к документу прилагается небольшой проект под **Xilinx Vivado** (***example.rar***).

В папке ***%PROJECT\_DIR%*** фактически находятся два проекта – один под **Active-HDL** (файл ***.asw***), второй под **Vivado** (файл ***.xpr***).

* Все текстовые исходники создаются в проекте **Active-HDL** и содержатся в папке:

***%PROJECT\_DIR%\aldec\_src\src\***

* Сторонние IP-ядра располагаются по адресу:

***%PROJECT\_DIR%\aldec\_src\src\cores\***

* Сгенерированные в **Vivado** IP-ядра располагаются по адресу:

***%PROJECT\_DIR%\%PROJECT\_NAME%.srcs\sources\_1\ip\***

* Констрейны расположены в папке:

***%PROJECT\_DIR%\%PROJECT\_NAME%.srcs\constrs\_1\new\***

В случае использования иерархичного дизайна модулей файлы подмодулей желательно располагать в папке с интуитивно понятным названием. Например, подмодули для файла ***example.sv*** лучше располагать в папке ***example\_inc\***, расположенной в той же директории, что и ***example.sv***.

Самым верхним уровнем проекта всегда является файл ***top\_level.sv***. Соответственно все другие файлы проекта располагаются по адресу ***%PROJECT\_DIR%\aldec\_src\src\top\_level\_inc\***

Из этого правила на данный момент есть три исключения:

* Сгенерированные в **Vivado** IP-ядра, они располагаются по своему стандартному пути. Требуемые исходники подключаются к проекту как ссылки.
* Файл ***glbl.v***, который требуется для симуляции некоторых IP-ядер, он распложен в той же директории, что и ***top\_level.sv***.
* Файл ***vers\_info.v***. Этот файл содержит в себе дату последнего синтеза проекта и обновляется автоматически с помощью TCL-скрипта. Распложен в корневой папке проекта. Подключается к проекту в виде ссылки.

# Общие требования к стилистике исходных кодов.

1. Каждый файл начинается с заголовка следующего вида:

*//-----------------------------------------------------------------------------*

*//*

*// Title : <TITLE>*

*// Design : <DESIGNNAME>*

*// Author : <AUTHOR>*

*// Company : <COMPANY>*

*//*

*//-----------------------------------------------------------------------------*

*//*

*// Description : <DESCRIPTION>*

*//*

*//-----------------------------------------------------------------------------*

Данный формат заголовка может быть задан в ПО Active-HDL в разделе ***Tools→Preferences→Generation→FileHeaders→Verilog***.

Описание заполняется по возможности.

2. Порты модуля должны быть сгруппированы по функциональному признаку, например:

* общий тактовый вход и сброс
* порты управления и статистики
* входной интерфейс модуля
* порты подключения к дополнительным модулям
* выходной интерфейс модуля

Имена портов группы желательно начинать с одной аббревиатуры, отражающей их принадлежность (***ctrl\_\****, ***in\_\****, ***out\_\****, ***stat\_\**** и т.д.).

Каждую группу желательно сопровождать кратким комментарием о ее назначении.

Имена портов различного направления (**in**, **out**, **inout**) располагаются с помощью табуляции на разном уровне для удобства визуального восприятия (см. прилагаемый проект).

3. *Input*-порт общего сброса ***rst*** желательно пропустить через один триггер для уменьшения его fanout.

Все пересинхронизации для сигналов управления/статистики должны быть выполнены внутри модуля, при необходимости во входные порты добавляется *input*-порт ***ctrl\_clk***.

4. Размер чистого кода модуля должен быть ограничен в разумных пределах. При очень большом функционале необходимо использовать подмодули, вынесенные в отдельные файлы.

5. Вложенные модули желательно выделять комментариями.

6. Описание сигналов для вложенного модуля лучше размещать сразу перед этим модулем, но перед комментарием. Исключение – если данный сигнал должен использоваться ранее, в данном случае он должен быть описан перед первым использованием.

7. Имена внутренних сигналов должны отражать их сущность и принадлежность. Для иерархичного дизайна возможны варианты ***module\_function***, ***moduleFunction*** и тому подобное. Для модулей без вложений возможно именовать сигналы только по функциональному признаку.

8. По возможности уменьшать количество внутренних сигналов с помощью многомерных массивов.