ATHENa v. 0.3.1 Tutorial

December 10, 2009

Part 1 – Tools Installation

ATHENa 0.3 requires three main types of programs to support the execution of its scripts: a Perl interpreter, Synthesis Tool(s), and Implementation Tool(s). Generally, basic versions of synthesis and implementation tools are provided freely by FPGA vendors. However, these free tools will not be able to perform synthesis and implementation targeting all FPGA devices available on the market. In order to generate results for a full range of FPGA devices, you will need to purchase appropriate licenses.

As of version 0.3.1, only Windows versions of Xilinx and Altera tools are supported. Linux is currently not supported.

The free versions of tools can be downloaded from the following web sites:

Perl

Perl Interpreter - http://www.activestate.com/downloads/

Synthesis and Implementation Toolsets

Xilinx

ISE - http://www.xilinx.com/tools/webpack.htm

Altera

Quartus - https://www.altera.com/support/software/download/altera_design/quartus_we/dnl-quartus_we.jsp

Synthesis and Implementation tools do not support all operating systems. For a list of supported operating systems see :

Xilinx - http://www.xilinx.com/ise/ossupport/index.htm

Altera - http://www.altera.com/literature/po/ss_quartussevswe.pdf

Once the tools are installed, you will need to run ***ATHENa_setup.bat*** to set the version of tools for Xilinx and/or Altera. Once finished, you must save and exit before any changes become effective.

This process generates ***ATHENa.bat*** in your root directory, and ***tool_config.txt*** in the **config** folder located inside of your root directory. The root directory is the directory called ATHENa, where you unpacked the ATHENa toolset. ***ATHENa.bat*** is a batch file that you use for starting ATHENa operation. ***tool_config.txt*** is a log file containing your tool settings.

The generated script should be used for all future runs of ATHENa with your preselected versions of tool. If changing a version of Xilinx and/or Altera tools is required, simply rerun the setup script. Sometimes, the setup script may not be able to detect your installed tools. You may need to specify the path to the root directory of your installed tools by choosing: 2. Add another version to the list, when you run the script.

Part 2 – General Project Setup and Reports

In order to prepare your code for evaluation using ATHENa, edit the file **design.config.txt** located in the **config** subdirectory of your root directory.

Inside the file, specify the desired workspace directory using the variable ***WORK_DIR***. This is a directory used as a root for all intermediate and result file directories. Then, you need to specify the location of the source folder using the variable ***SOURCE_DIR***. This is a folder containing your VHDL source files. Specify the remaining parameters according to the meaning of options as explained in Appendix A.

Once the preparation of the configuration file is completed, double click on **ATHENa.bat** located in your root folder to start. The results generated by the scripts will be shown in your command line window, as well as stored in text files located in the directory ***WORK_DIR*** under the following subfolder **${application}\${date}_${projectname}_${ instance_no}**.

${application}* is the type of application as specified in **design.config.txt**. *${date} is the date at which the project is run. *${projectname}* is the name of the project as specified in **design.config.txt**. *${instance_no}* is an instance number of the project. An instance number is used in order to distinguish folders created on different time during the same day.

In each run of ATHENa, four report files are generated. These are option, resource utilization, timing, and execution time reports. The option report contains information about the specific options of tools used in a given run (default tool options are not listed). The resource utilization report contains information about the use of FPGA resources. Timing report contains timing related results. Execution time report contains the time taken by each tool.

Part 3 – Application Setup

As of version 0.3.1, there are three supported applications :

single_run, placement_search and **exhaustive_search.**

3.1 single_run

single_run is the most basic ATHENa's application. It performs a single run through synthesis and implementation for all target FPGA devices specified in **design.config.txt.** Options of synthesis and implementation used for all these target devices are provided in the file:

options.<OPTIONS>_<OPTIMIZATION_TARGET>.txt

located in the subdirectory **config** of the root directory.
OPTIONS and **OPTIMIZATION TARGET** are variables defined in **design.config.txt.**
OPTIONS = default or **user. OPTIMIZATION_TARGET = speed**, **area**, or **balanced.**

See Appendix D for a list of options that can be specified in the files

options.<OPTIONS>_<OPTIMIZATION_TARGET>.txt

3.2 placement_search

placement_search is an application that allows running implementation automatically for multiple values of an option that determines a starting point of placement within a given FPGA device. These options include: **Cost Table** for Xilinx and **Seed** for Altera.

If

APPLICATION = **placement_search**

is specified in the **design.config.txt**, then the tool will look into the file **placement_search.txt** located in the **config** subfolder of the root directory. User can specify the range of the given option values using the following format:

<number>; **or** <start_number>: <step> : <end_number>;

The first format allows specifying a single value of an option. The second format specifies a sequence of values starting from start_number, and ending with end_number, with the value incremented by step in each subsequent run.
The specification of a value or a range has to be terminated by a semi-colon. A user can combine multiple specifications together, separated by semicolons. For instance,

XILINX_COST_TABLE_VALUES = 6; 13:6:30; 55;

means the values of the cost table equal to 6, 55 and the range from 13 to 30 with the step of 6. Thus, the implementation will be repeated for the following values of the cost table: 6, 13, 19, 25, and 55.

See Appendix E for a full list of variables of **placement_search.txt**.

3.3 exhaustive_search

exhaustive_search extends placement_search to allow a user to specify a range of option values for several types of options (beyond options that can be specified in placement_search).

In general, **exhaustive_search** is used to find optimum synthesis and implementation options for a given source code, target device, tools, tool versions, and optimization target (speed, area, or balanced).

The description of the exhaustive search strategy is given in the file:
exhaustive.<strategy_name>.txt, where the string <strategy_name> can be chosen arbitrarily.

For each option, several possible values used in the exhaustive search are specified. All options are grouped into Level 1 options and Level 2 options. Level 1 options are investigated first in order to determine a number of best combinations given by the variable BEST_LEVEL_1_OPTION_SETS. For each set of options selected as best at Level 1, an exhaustive search is performed for all options at Level 2.

To select which strategy to use, simply use the **strategy_name** as a parameter to EXHAUSTIVE_SEARCH_STRATEGY option.

See Appendix F for a list of options of **exhaustive.<strategy_name>.txt.**

Part 4 – Example Run

We have included example projects in our ATHENa package. These files are located inside of the folder *examples* of the root directory. In order to perform a test run, proceed to *config/design.config.txt* and make sure that this file contains the lines as shown in our basic and advanced examples. Alternatively, you can overwrite design.config.txt by the provided template inside the example directories and modify *WORK_DIR* and *SOURCE_DIR* path to reflect your work directory and source directory, respectively. Once completed, you can navigate to package folder and click on ***ATHENa.bat*** to start running the project.

Note: replace %ROOT% with the full path to ATHENa's root folder.

Basic example: Multiplier

WORK_DIR = %ROOT%\workspace
SOURCE_DIR = %ROOT%\examples\mult
SOURCE_LIST_FILE = source_list.txt
PROJECT_NAME = multiplier
TOP_LEVEL_ENTITY = main
TOP_LEVEL_ARCH = main
CLOCK_NET = clk

OPTIMIZATION_TARGET = speed
OPTIONS = default

APPLICATION = single_run

FPGA_VENDOR = Xilinx

FPGA_FAMILY = Spartan3
FPGA_DEVICES = best_match
SYN_CONSTRAINT_FILE = default
IMP_CONSTRAINT_FILE = default
REQ_SYN_FREQ = default
REQ_IMP_FREQ = default
MAX_SLICE_UTILIZATION = 0.8
MAX_BRAM_UTILIZATION = 1
MAX_DSP_UTILIZATION = 1
MAX_MUL_UTILIZATION = 1
MAX_PIN_UTILIZATION = 0.9

END FAMILY

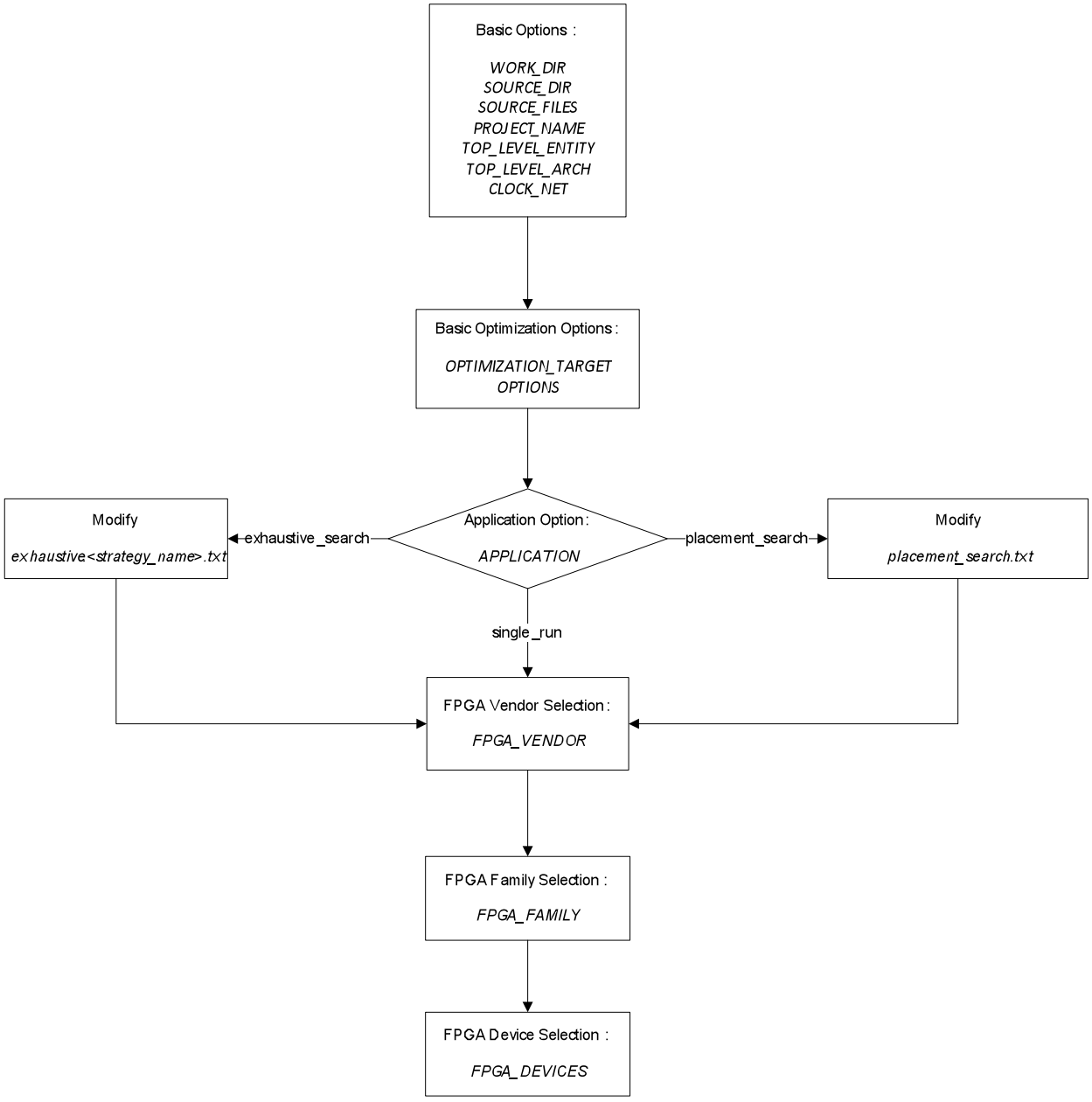
END VENDOR

Advanced example : SHA2-256

```
WORK_DIR = %ROOT%\workspace
SOURCE_DIR = %ROOT%\examples\sha256
SOURCE_LIST_FILE = source_list.txt
PROJECT_NAME = sha256
TOP_LEVEL_ENTITY = sha256
TOP_LEVEL_ARCH = bl_arch
CLOCK_NET = clk

LATENCY = TCLK*65
THROUGHPUT = 512/(TCLK*65)

OPTIMIZATION_TARGET = speed
OPTIONS = default

APPLICATION = single_run

FPGA_VENDOR = xilinx

        FPGA_FAMILY = spartan3
        FPGA_DEVICES = best_match
        SYN_CONSTRAINT_FILE = default
        IMP_CONSTRAINT_FILE = default
        REQ_SYN_FREQ = default
        REQ_IMP_FREQ = default
        MAX_SLICE_UTILIZATION = 0.8
        MAX_BRAM_UTILIZATION = 1
        MAX_DSP_UTILIZATION = 1
        MAX_MUL_UTILIZATION = 1
        MAX_PIN_UTILIZATION = 0.9
        END FAMILY

END VENDOR

FPGA_VENDOR = altera

        FPGA_FAMILY = Cyclone
        FPGA_DEVICES = all
        REQ_IMP_FREQ = default
        END FAMILY

END VENDOR
```

Part 5 – Basic ATHENa Project Flow

Below is a flow that a user might use to create an ATHENa project using ***design.config.txt*** file. All options specified in the given below flow are required. The parameters listed here are the absolute minimum options the tool requires to operate.



Part 6 – Workspace

Workspace is the place where ATHENa creates an instance of your project. An instance of your project is created under the path described in Part 2:

$workspace\$application\${date}_${projectname}_${instance_no}

An ATHENa benchmarking project may contain many runs for each device. Hence, to view original information as created by vendor's tool before ATHENa processes this information, the user needs to navigate to the following path of the project folder:

${vendor}\${family}\${device}\${run_no}

${vendor}, ${family}* and *${device} are the name of the vendor, family and device used in a given run, respectively. *${run_no}* is the run number, starting from 1. This folder numbering is used to distinguish between subsequent runs using different options.

Part 7 – Constraint File

As of version 0.3, constraint file is supported for Xilinx only. The file must be located inside of your source folder. The user can specify the name of the constraint file in the design.config.txt under **SYN_CONSTRAINT_FILE** and **IMP_CONSTRAINT_FILE** for synthesis and implementation, respectively.

Part 8 – Project Termination and Workspace Clearing

In the case that user accidentally starts an ATHENa project, user can stop the run by pressing **CTRL+C** in the console window. Please note that a folder corresponding to the task that the user accidentally started is created inside of the specified workspace. To clean up the undesired directories, the user can either delete them manually or by running the ***clean_workspace.bat*** script located inside your root directory. Please be aware that **clean_workspace.bat** file will also remove directories corresponding to any uncompleted projects and projects that do not produce any results. Hence, be careful when you are running the script as you may accidentally delete a run that you would like to use as a source for debugging.

Appendix A : "design.config.txt" Options

Option	Explanation
WORK_DIR	Directory of your workspace.
SOURCE_DIR	Directory of your source files for the project.
SOURCE_LIST_FILE	Contains the list of files to be compiled starting from the lowest level to the highest level of file hierarchy. Each file name should be separated by new line.
TESTBENCH_DIR	<currently unsupported> To be used for an automated verification of the circuit functionality in the future.
TESTBENCH_FILES	<currently unsupported> To be used for an automated verification of the circuit functionality in the future.
PROJECT_NAME	Project's name. The name of the project is directly associated with the generated project name in your specified workspace. For Altera, project's name **must** be the same as your top level entity's name.
TOP_LEVEL_ENTITY	Name of top level entity.
TOP_LEVEL_ARCH	Name of the architecture of your top level entity.
CLOCK_NET	Name of the global clock in your design.
LATENCY	The equation for calculating circuit's latency. This should be in terms of TCLK, where TCLK is the minimum clock period of the circuit.
THROUGHPUT	The equation for calculating circuit's throughput. This should be in terms of TCLK, where TCLK is the minimum clock period of the circuit.
OPTIMIZATION_TARGET	Synthesis and implementation strategy. Optimization for area, speed, or balanced.
OPTIONS	Option mode (default or user). In the default mode, default options of tools, as specified in the file options.default_<OPTIMIZATION_TARGET> will be used. If you want to use non-default options of tools, please change this variable to user, and modify the file options.user_<OPTIMIZATION_TARGET>.
APPLICATION	Name of an application. single_run, placement_search, and exhaustive_search are currently supported. single_run performs a single run through synthesis and implementation for all specified FPGA devices. Placement search performs multiple runs with different values of the parameters determining starting point of the placement, as specified in the file config/placement_search.txt.
FPGA_VENDOR … END VENDOR	Target FPGA vendor, i.e. Xilinx or Altera
FPGA_FAMILY … END FAMILY	Target FPGA family of a specified vendor, i.e. Spartan3
FPGA_DEVICES ... END DEVICES	List of target FPGA devices based on a specified family and vendor. Device names must be separated by comma. Two special modes of operation exist for this option, ***best_match*** and ***all***. For ***best_match***, the script will search for the smallest device that passes all criteria as specify by the user. For ***all***, the script will go through all the available devices of the specified family. See Table 2 for details of available parameters. Note : The special modes will search through the FPGA devices specified in the library file located in the ***device_lib*** folder.

Appendix B : Xilinx FPGA_DEVICES specific options

Options	Explanation
MAX_SLICE_UTILIZATION	Maximum slice utilization ratio
MAX_BRAM_UTILIZATION	Maximum BRAM utilization ratio
MAX_DSP_UTILIZATION	Maximum DSP utilization ratio
MAX_MUL_UTILIZATION	Maximum multiplier utilization ratio
MAX_PIN_UTILIZATION	Maximum pin utilization ratio
SYN_CONSTRAINT_FILE	Path to a synthesis constraint file (*.xcf). If a contraint file is not used, specify *default*
IMP_CONSTRAINT_FILE	Path to an implementation constraint file (*.ucf). If a contraint file is not used, specify *default*
REQ_SYN_FREQ	Requested synthesis clock frequency
REQ_IMP_FREQ	Requested implementation clock frequency

Note : These options are located inside “FPGA_DEVICES ... END DEVICES” clause

Appendix C : Altera FPGA_DEVICES specific options

Options	Explanation
REQ_IMP_FREQ	Requested implementation frequency

Note : These options are located inside “FPGA_DEVICES ... END DEVICES” clause

Appendix D : “option.<option>_<optimization_target>.txt” options

Options	Explanation
XILINX_SYNTHESIS_TOOL	Xilinx synthesis tool (SYNPLIFY or XST) <only XST is currently supported>
ALTERA_SYNTHESIS_TOOL	Altera synthesis tool (synplify pro or quartus_map) <only quartus_map is currently supported>
ACTEL_SYNTHESIS_TOOL	<currently unsupported>
XILINX_SYNPLIFY_OPT ... END_OPT	<currently unsupported>
XILINX_XST_OPT ... END_OPT	Options for Xilinx’s XST (synthesis)
ALTERA_QUARTUS_MAP_OPT ... END_OPT	Options for Altera Mapping Tool (synthesis)
ALTERA_QUARTUS_FIT_OPT ... END_OPT	Options for Altera Fitting Tool (implementation)
ACTEL_SYNPLIFY_OPT ... END_OPT	<currently unsupported>
XILINX_NGDBUILD_OPT ... END_OPT	Options for Xilinx’s NGDBUILD
XILINX_MAP_OPT ... END_OPT	Options for Xilinx’s MAP
XILINX_PAR_OPT ... END_OPT	Options for Xilinx’s PAR

XILINX_TRACE_OPT ... END_OPT	Options for Xilinx's PAR

Appendix E : "placement_search.txt" options

Options	Explanation
XILINX_COST_TABLE_VALUES	Xilinx's cost table. Possible range is from 1 to 100
ALTERA_SEED_VALUES	Altera's seed. Possible range is from 1 to 2^{32}-1

Appendix E : "exhaustive.<strategy_name>.txt" options

Options	Explanation
General Options	
TARGET_CLK_FREQ	Target synthesis and implementation frequency. If Altera is used, only implementation frequency is used.
RUN_ALL_OPTIONS	The tool will loop through all specified options if **YES** is selected. Otherwise, it will stop whenever the target clock frequency is reached.
BEST_LEVEL_1_OPTION_SETS	Number of best combinations of options from Level 1 that will be used for runs at Level 2.
Level 1 Options	
XILINX_SYNTHESIS_TOOL	<currently unsupported>
XILINX_SYNPLIFY_OPT	<currently unsupported>
XILINX_XST_OPT	Xilinx's XST options
XILINX_MAP_OPT	Xilinx's MAP options
XILINX_PAR_OPT	Xilinx's PAR options
Level 2 Options	
XILINX_COST_TABLE_VALUES	Xilinx's parameter determining the starting placement point

Note : All options inside Level 1 (with the exception of synthesis tool option) must end with "END_OPT" clause.

Options	Explanation
General Options	
TARGET_CLK_FREQ	Target implementation frequency .
RUN_ALL_OPTIONS	The tool will loop through all specified options if **YES** is selected. Otherwise, it will stop whenever the target clock frequency is reached.
BEST_LEVEL_1_OPTION_SETS	Number of best combinations of options from Level 1 that will be used for runs at Level 2.
Level 1 Options	
ALTERA_SYNTHESIS_TOOL	<currently unsupported>
ALTERA_SYNPLIFY_OPT	<currently unsupported>
ALTERA_QUARTUS_MAP_OPT	Altera mapping tool options
ALTERA_QUARTUS_FIT_OPT	Altera fitting tool options
Level 2 Options	

ALTERA_SEED_VALUES	Altera's parameter determining the starting point for placement.

Note : All options inside Level 1 (with the exception of synthesis tool option) must end with "END_OPT" clause.