Getting Started User Guide (UG105)

All Achronix Devices

Achronix®
Data Acceleration

Copyrights, Trademarks and Disclaimers

Achronix Semiconductor Corporation

2903 Bunker Hill Lane
Santa Clara, CA 95054
USA

Website: www.achronix.com
E-mail : info@achronix.com

Table of Contents

Chapter - 1: Overview

This guide serves as a concise introduction to the Achronix tool flow. After an overview of the tool and IP flow, the Quickstart design included with all ACE installations is used to illustrate the tool flow.

The Achronix design flow uses two software tools:

- Synplify Pro – used for design synthesis
- ACE – used for place-and-route of the design elements and for bitstream generation

Synplify Pro operates in GUI and batch modes and ACE operates in GUI, command-line, and batch modes.

Note

This user guide assumes that Synplify Pro and ACE are already installed. For help installing the tools or if issues are being encountered, such as problems with launching the tools or the desired device is missing, see the *ACE Installation and Licensing Guide* (UG002).

The following documents provide additional information on topics covered in this guide:

- *Synthesis User Guide* (UG018)
- *ACE User Guide* (UG070)

The following document provides useful information on topics not covered in this guide:

- *Simulation User Guide* (UG072)

Chapter - 2: Achronix Tool Flow

Tool Flow

The Achronix design flow uses two tools:

- Synplify Pro – used for design synthesis
- ACE – used for place-and-route of design elements and for bitstream generation

The primary output of Synplify Pro is the design netlist which is imported into ACE for place-and-route. ACE also performs IP configuration, floorplanning, timing analysis, and more.

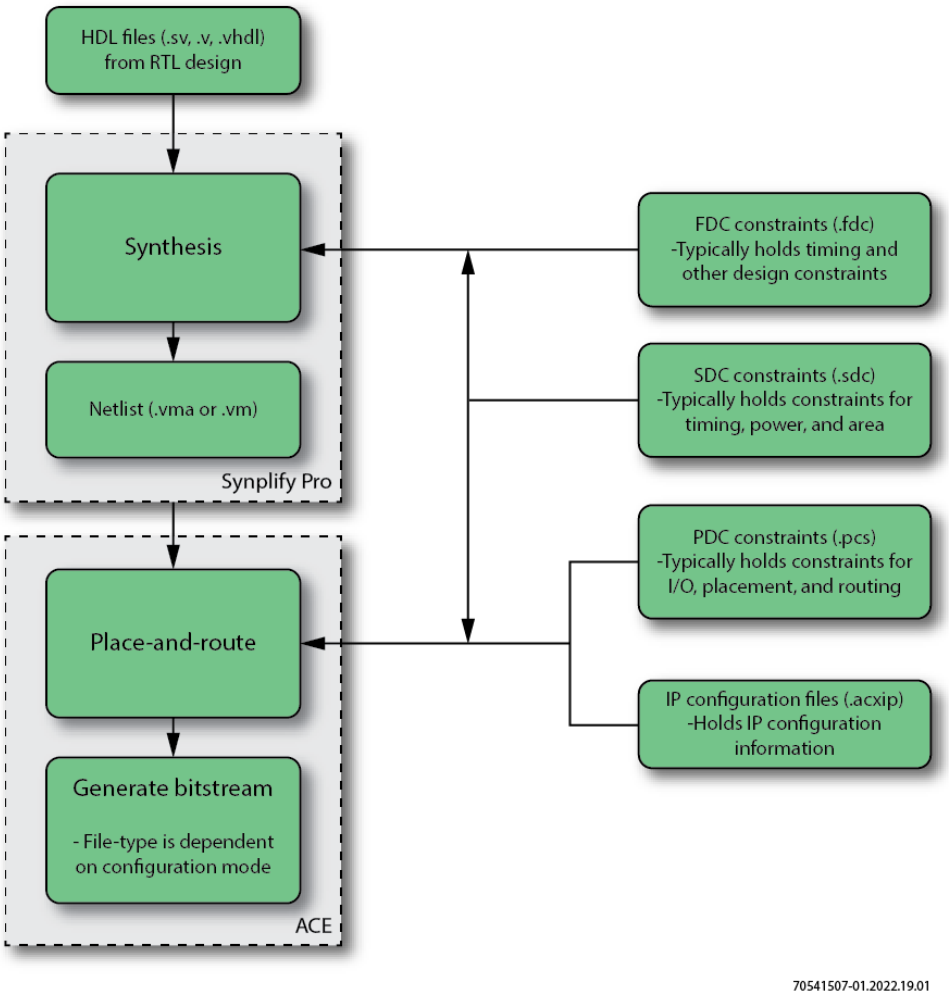

Figure 1: ***Achronix Design Flow***

IP Flow

Achronix offers two types of IP:

- Core IP – traditionally thought of as "soft IP". In Achronix terminology, "core" is short for "fabric core" and is synonymous with what is traditionally thought of as "configurable fabric". It follows that these are IP cores located in the fabric of the device.
- IORing subsystems – traditionally thought of as "hard IP". These interface subsystems surround the core and include GPIO, PLLs, memory interfaces, and more. Hence the term, "IORing".

The natural question is, "How does IP fit into the tool flow?" The answer depends on which type of IP is being used, though both processes begin in ACE. If using Core IP, the synthesis run must include the output products of the IP generation. Namely, the Core IP HDL and constraints (if any are generated). IORing subsystems do not need to be synthesized in Synplify Pro because ACE contains the boundary timing information between the IORing and Core. However, constraints from IORing subsystems are often useful in synthesis as they define things that would otherwise be needed (e.g., clocks are defined as part of the PLL configuration process).

See the image below for IORing subsystem and Core IP flows:

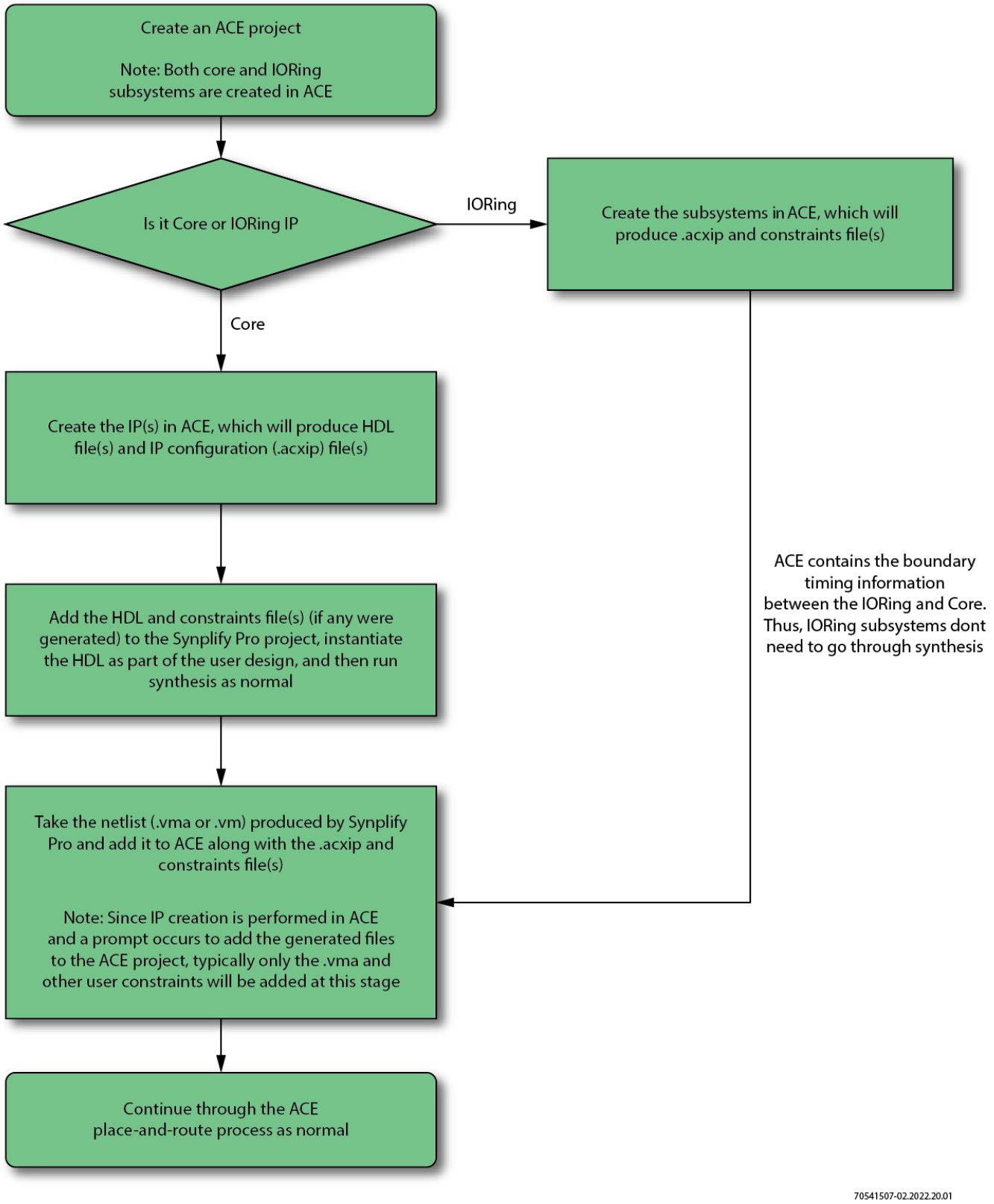

Figure 2: ***IP Flow***

Chapter - 3: Launching Synplify Pro

Synplify Pro can be run in either the GUI or batch mode. Refer to the *ACE Installation and Licensing Guide* (UG002) for details on installing Synplify Pro and *Synthesis User Guide* (UG018) for more details on synthesis.

> **Note**
>
> To list the several options available when launching Synplify Pro, use the `-h` argument with the `synplify_pro` command.

GUI Mode

If a Synplify Pro shortcut or alias was not created, see the following table for the path to the program file.

Table 1: *Synplify Pro File Paths*

Executable File	Typical Windows Path	Typical Linux Path
`synplify_pro`	`C:\Synopsys\<version>\bin`	`<install_path>/fpga/<version>/bin`

When executed, the following window is displayed:

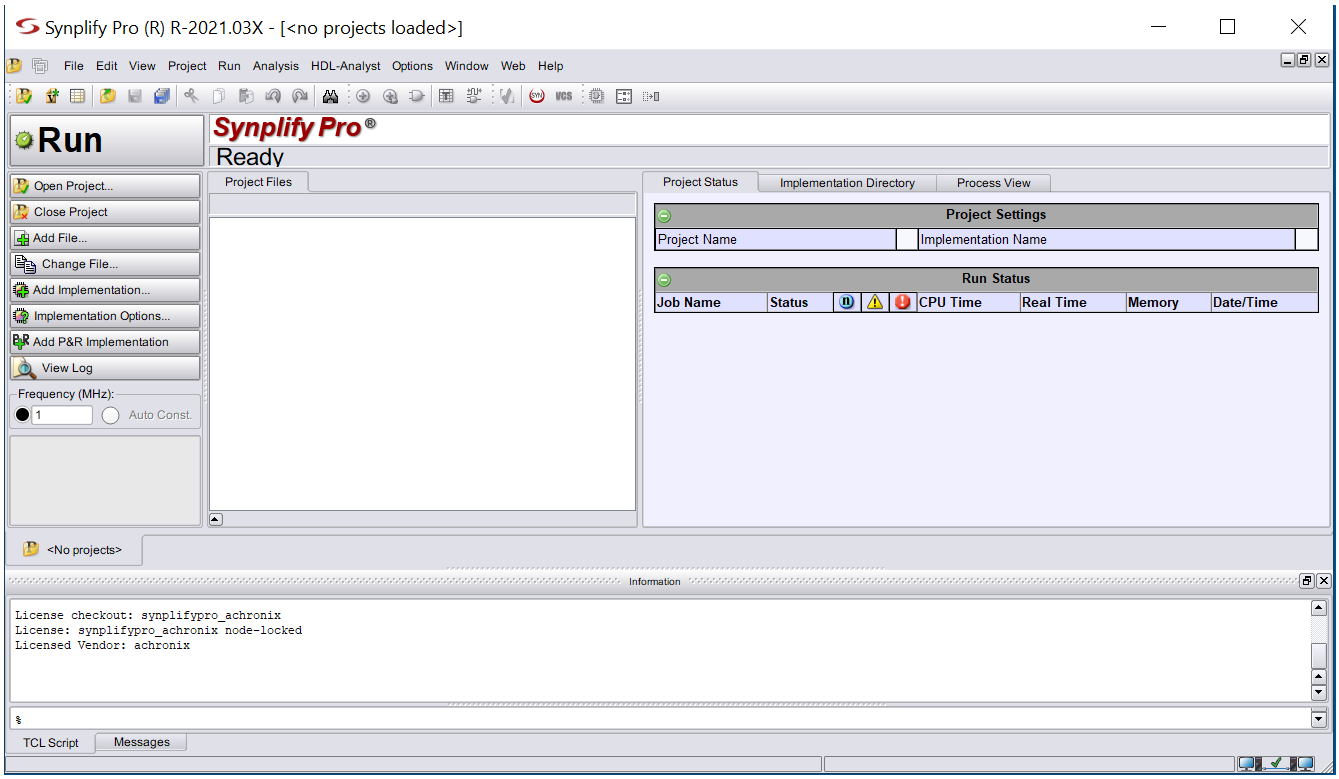

Figure 3: *Synplify Pro Opening Display*

Batch Mode

To run Synplify Pro in batch mode, use the `-batch` argument with the `synplify_pro` command. Reference designs support batch mode and have specific scripts for batch runs.

Note

A floating license is required in order to use batch mode with Synplify Pro.

Chapter - 4: ACE Execution Modes

ACE can be run with full functionality in three different modes:

1. GUI Mode
2. Command-line Mode
3. Batch Mode

For details beyond that covered in this guide, refer to the *ACE User Guide* (UG070).

GUI Mode

If an ACE shortcut or alias doesn't exist, see the following table for the path to the executable program file.

Table 2: *ACE Program File Path*

Executable File	Typical Windows Path	Typical Linux Path
ace	C:\Program Files\Achronix CAD Environment\Achronix	<install_path>/Achronix-linux

To run in GUI mode, invoke the **ace** executable either with no options or with the **-gui** option. GUI mode launches the interactive window from which all commands are issued.

```
Starting ACE in GUI Mode, Implicit

% ./ace
```

or

```
Starting ACE in GUI Mode, Explicit

% ./ace -gui
```

Command-line Mode

To run in command-line mode, invoke the `ace` executable with the `-b` option from a console. Command-line mode takes control of the console and allows interactive entry of Tcl commands at the command prompt.

Starting ACE in Command-line Mode

```
% ./ace -b
-- ACE -- Achronix CAD Environment -- Version 5.4 -- Build  84486- -- Date 2015-02-11 19:58
-- (c) Copyright 2006-2015 Achronix Semiconductor Corp.  All rights reserved.
-- all messages logged in file /home/username/.achronix/ace_2015_02_13_11_00_11.log, created at 11
:00:11 on 02/13/2015
INFO: License ace-v1.0 on server acxlicense (9 of 10 licenses available). Running on docs.
achronix.local (x86_64).
ACE>
```

Batch Mode

To run in batch mode, invoke the **`ace`** executable with the **`-b`** option and the **`-script_file`** option. Reference designs support batch mode and have scripts to refer to as examples.

Starting ACE in Batch Mode

```
% ./ace -b -script_file <path_to_script_file>.tcl
```

Chapter - 5: The Quickstart Design

Introduction

The simple Quickstart design features a 2-bit binary up-counting LED display to indicate that the board and FPGA are operating properly upon powerup. The Quickstart design RTL along with the device-specific netlists and constraints are available under the `<ACE_install_dir>Achronix/examples/quickstart` directory.

The *Introduction to Achronix FPGA Design Tool Flow* video parallels this tutorial and offers a helpful addition to the learning process.

While this tutorial is intended to help gain familiarity with the tool flow, some details are omitted to keep it concise. Relevant user guides that provide additional details on the tools can be found at the end of each section.

> **Note**
>
> Screenshots in this tutorial use the AC7t1500ES0 part with the F53A0 package and C2 speed grade. However, Quickstart design source files for all devices are included in ACE installations.

The Quickstart Design

Synthesizing with Synplify Pro

The Synplify Pro portion of this tutorial is summarized as follows:

1. File → New Project
2. Add File... → add source files
   - Add the relevant device model
   - Add design files, constraints
3. Implementation Options...
   - Synthesis/device options
4. When ready, run synthesis (the "implementation")

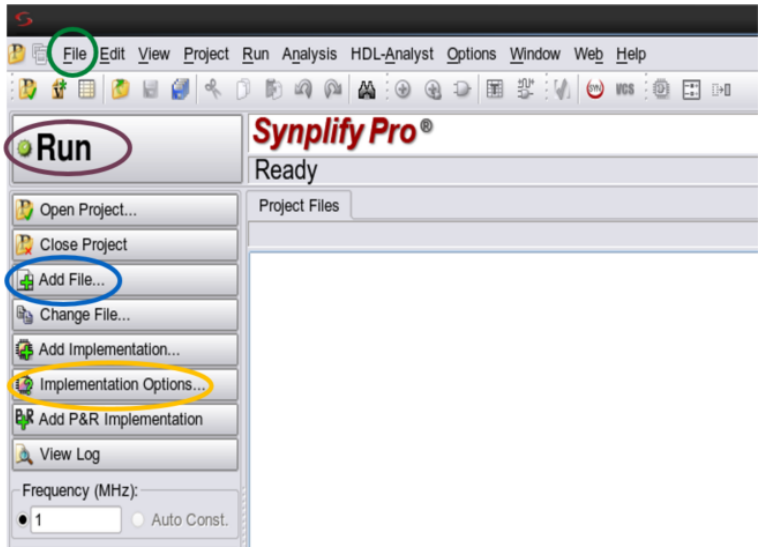

Figure 4: ***Overview of Synplify Pro Steps***

Follow these steps to synthesize the Quickstart design with Synplify Pro:

1. Launch Synplify Pro and create a new project (select **File** → **New Project** from the menu). For details on launching Synplify Pro, see Launching Synplify Pro (see page 9).
2. Add the source files (click **Add File...** at the left-side of the Synplify Pro window). If already familiar with Synplify Pro, the `quickstart.vma` file contains the pre-generated design netlist. In this case, skip to Running Place-and-Route in ACE (see page 18).

Table 3: *Synplify Project Files*

Files	Typical Windows Path	Typical Linux Path	Purpose
`<device_name>_synplify.sv`	`C:\Program Files\Achronix CAD Environment\Achronix\libraries\device_models`	`<install_path>/Achronix-linux/libraries/device_models`	Makes Synplify Pro aware of the device information necessary to perform synthesis. This is NOT design-specific and is required for all designs.
`quickstart_<device_name>.v` and `counter.v`(1) (2)	`C:\Program Files\Achronix CAD Environment\Achronix\examples\quickstart\rtl`	`<install_path>/Achronix-linux/libraries/examples/quickstart/rtl`	Design RTL files.
`quickstart_<device_name>_ioring.sdc`(3)	`C:\Program Files\Achronix CAD Environment\Achronix\examples\quickstart\<device_name>`	`<install_path>/Achronix-linux/libraries/examples/quickstart/<device_name>`	The .sdc file type stands for Synopsys Design Constraints. Typical usage is to constrain timing, power, and area. In this case, a clock is defined. If the file isn't shown, select **Files of type** in the window and add files using **All Files (*.*)**.

Table Notes

1. If using a Speedcore device, the file is named `quickstart.v` instead.
2. If using the AC7t1550, also add `quickstart_AC7t1550_top.v`.
3. If using a Speedcore device, the file is named `quickstart.sdc` instead.

Tip

Files with "ioring" in their name are generated as part of the IORing flow. In this tutorial, these files have already been generated in order to expedite the introduction to the tools.

Note

Synplify Pro compiles files from the top down. In practice, this means the design hierarchy should be inverted: the lowest level files should be at the top (this is usually the device model file, `<device_name>_synplify.sv`) and files are hierarchically organized *downward* towards their top-level file at the bottom. See the screenshot below for an example using this design.

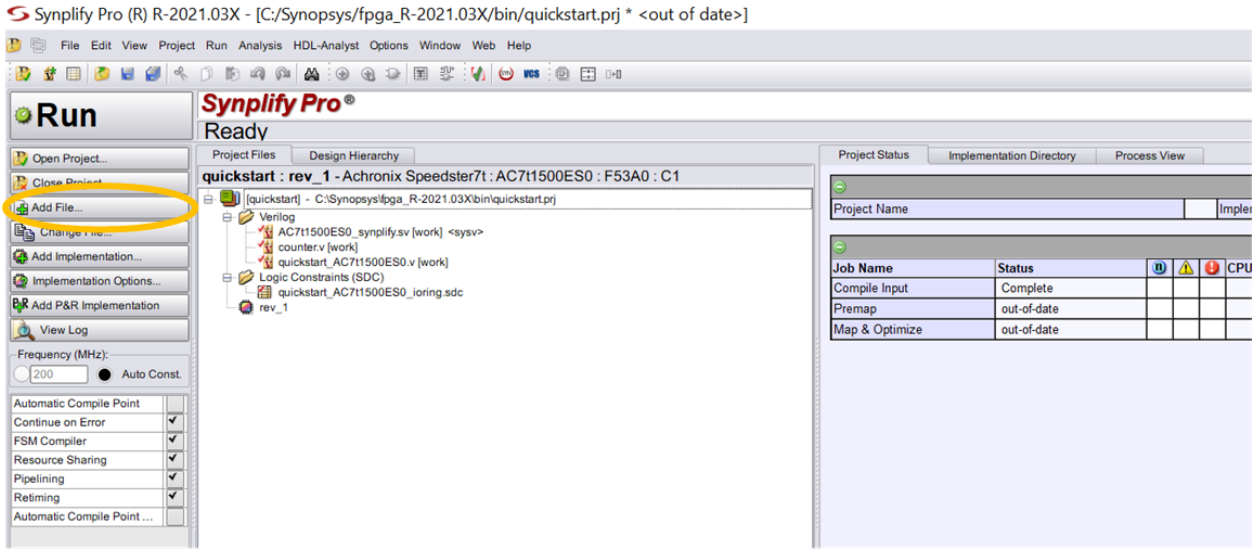

Figure 5: ***Source Files Added to Synplify Pro Project***

3. Select the device and set other house-keeping items (click **Implementation Options...** at the left-side of the Synplify Pro window):
    - **Device** Tab – select the **Technology**, **Part**, **Package**, and **Speed** parameters that match the desired device.

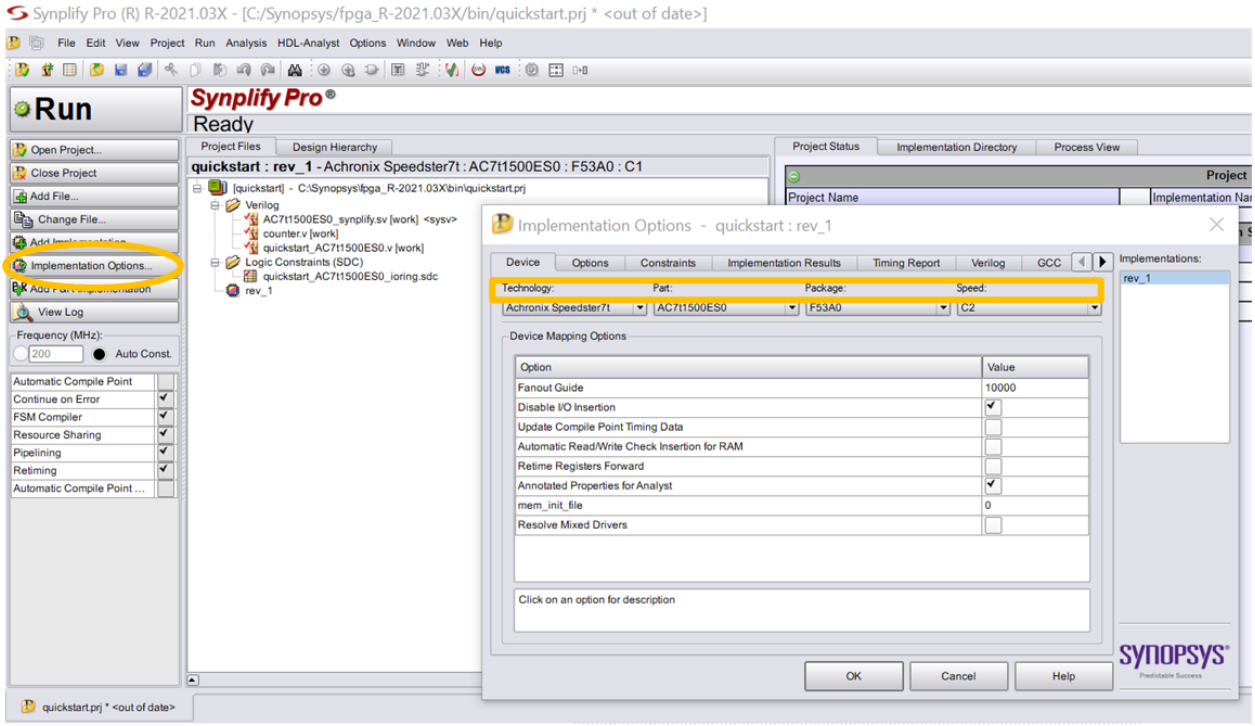

Figure 6: ***Device Tab of Implementation Options Window***

- **Implementation Results** Tab – select where synthesis products are output via the **Results Directory** value. Observe that if the **Implementation Name** value is changed, the final part of the **Results Directory** path changes too. This can be a convenient way of creating multiple synthesis runs in the same parent directory, simply by changing the **Implementation Name** value each time.

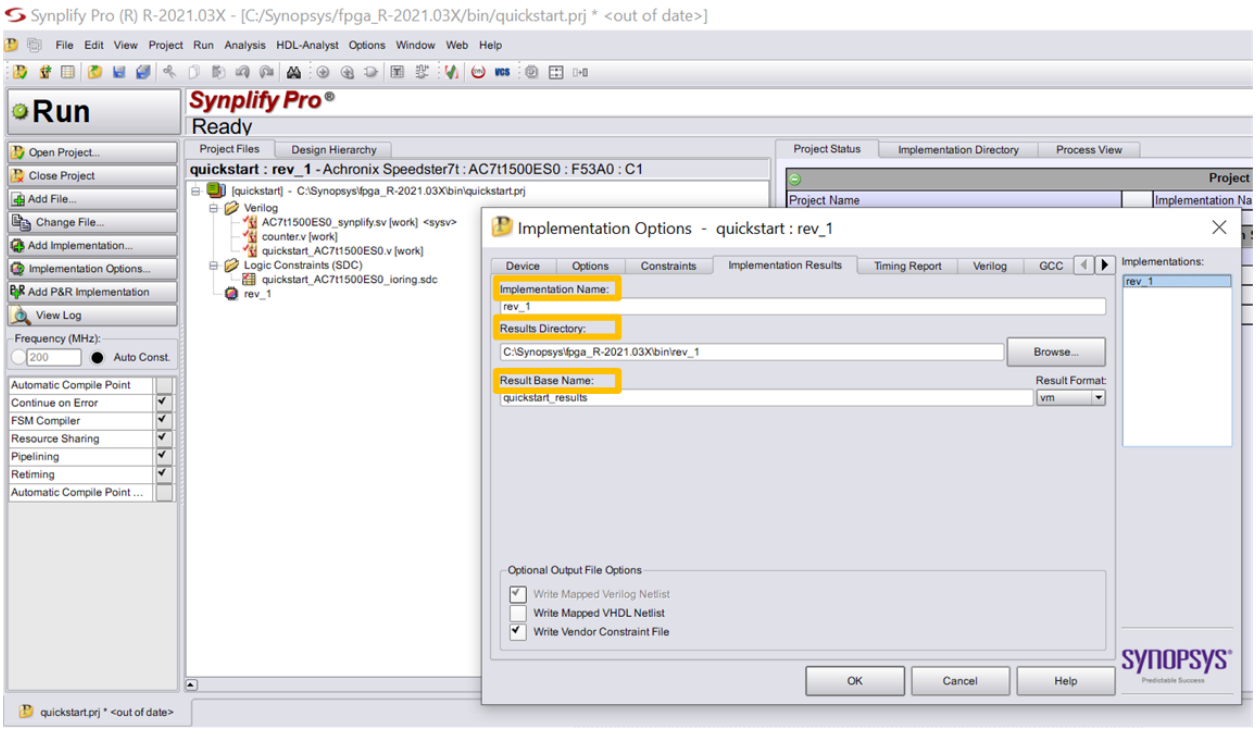

Figure 7: ***Implementation Options Window Results Tab***

- **Verilog** Tab – set the **Top Level Module** to the name of the top-level module (the file type is not necessary, just the name).
    - Set **Include Path Order:** to the libraries directory in your ACE installation path. Typically this is at `C:\Program Files\Achronix CAD Environment\Achronix\libraries\` for Windows and `<install_path>/Achronix-linux/libraries/` for Linux. This enables Synplify Pro to use the Achronix libraries.
    - The **Compiler Directives** option is not used for the Quickstart design. In future designs, add desired compiler directives here.

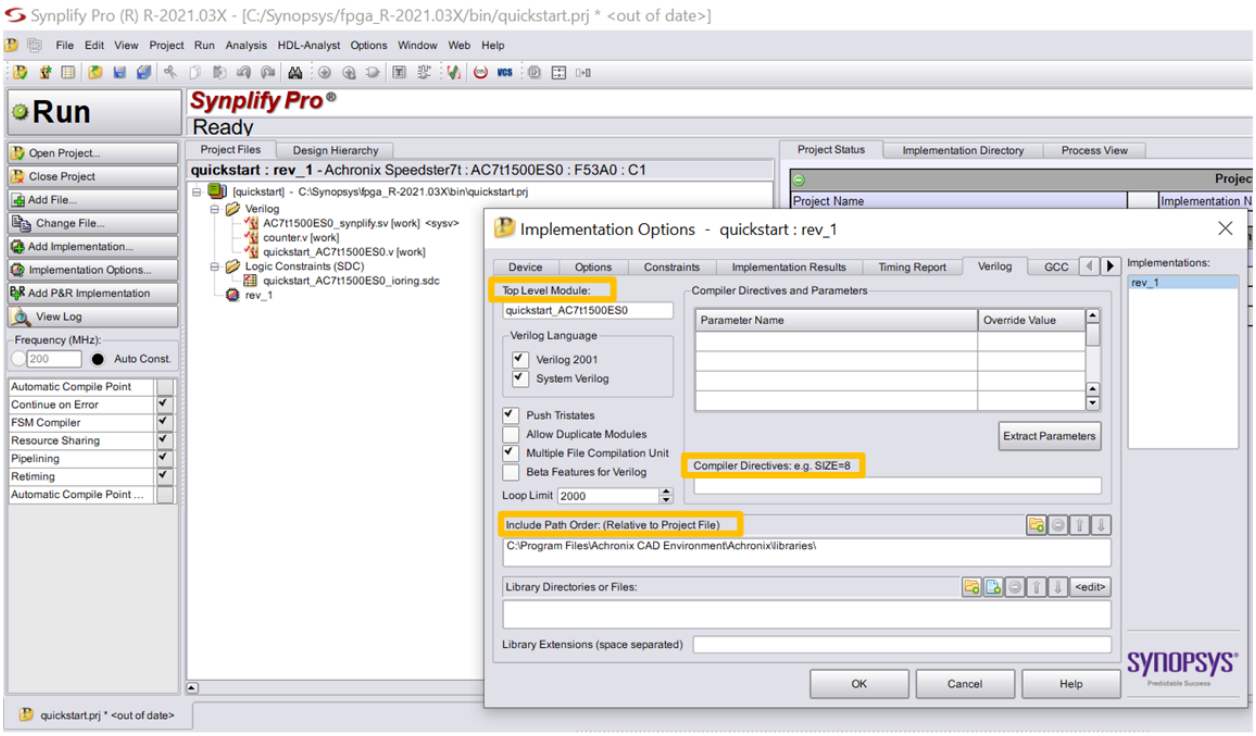

Figure 8: ***Implementation Options Window Verilog Tab***

The following tabs aren't used in this tutorial but might be useful in future designs:

- **Options** Tab – contains options like pipelining, retiming, and continuing through synthesis errors.
- **Constraints** Tab – offers an alternative to adding constraints instead of adding them through the **Add File...** button.
- **Timing Report** Tab – specify the number of reported critical paths and start/end paths. Does not set a limit on the number of critical paths or start/end paths considered by the tool, only the ones reported.
- **GCC** Tab – specifies how gated and generated clocks are handled.
- **Place and Route** Tab – not used in the Achronix tool flow because place-and-route is performed in ACE.

4. Synthesize the design (click the big **Run** button on the top-left of the window).

When synthesis finishes, a synthesized netlist (denoted by the .vma or .vm file-type) is generated in the **Results Directory** set earlier. Any errors, warnings, and informational messages can be found in the **Project Status** window. If desired, the project can be saved by selecting **File → Save As...** from the menu. See *Synthesis User Guide* (UG018) for more information on Synplify Pro and synthesis.

Running Place-and-Route in ACE

The ACE portion of this tutorial is summarized as follows:

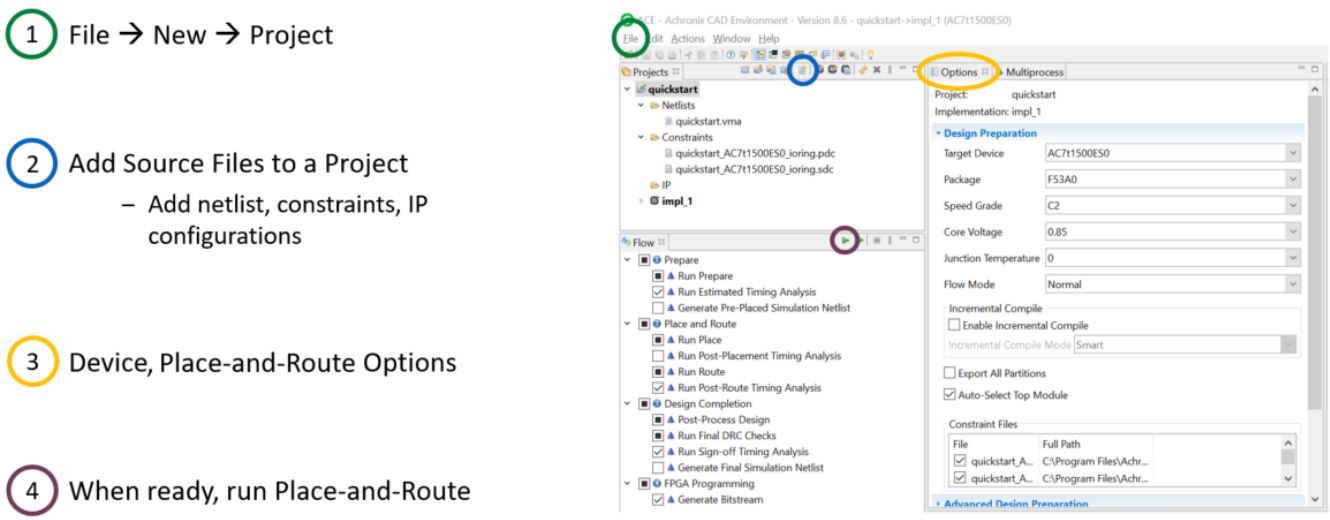

Figure 9: *Overview of ACE Steps*

Follow these steps to process the Quickstart design in ACE:

1. Launch ACE and create a new project (select **File** → **Create Project...** from the menu). See Launching ACE (see page 11) for details.

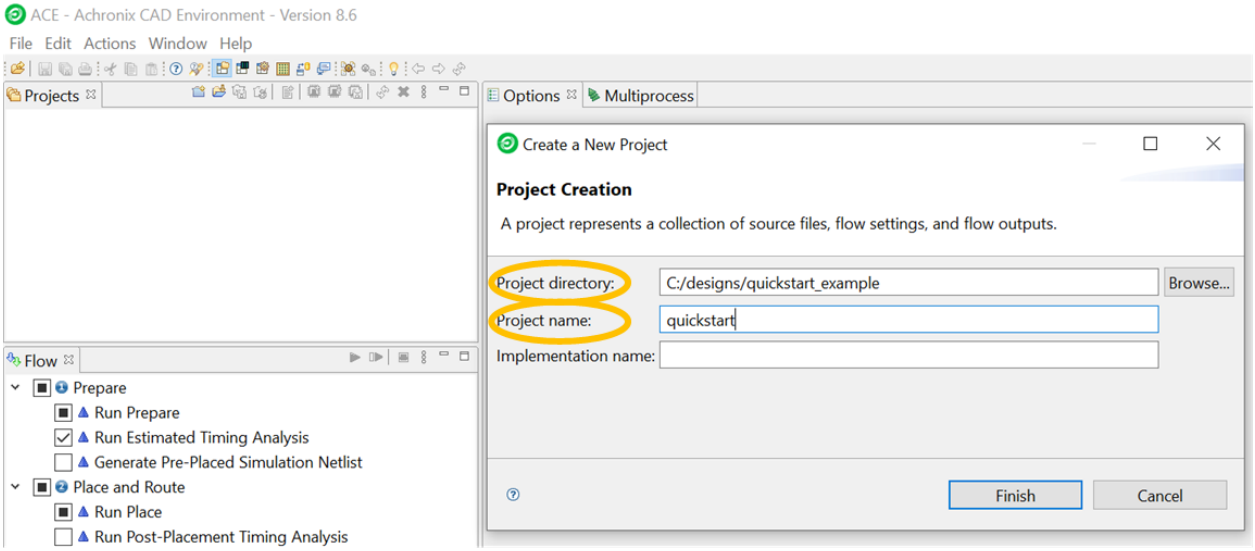

Figure 10: *New Project Creation Prompt*

2. Set the **Project directory** value to the desired directory in which to store the project.
3. Set the **Project name** value to "quickstart".
4. If the **Implementation name** value is left blank, the name defaults to `impl_1`.

5. Add the following files for ACE (select **File** → **Add Project Source Files...** from the menu, or click the circled button as shown in the image below):
   Table 4: *ACE Project Files*

Files	Typical Windows Path	Typical Linux Path	Purpose
`quickstart.vma` or `quickstart.vm`	`C:\Program Files\Achronix CAD Environment\Achronix\examples\quickstart\<device_name>`(1)	`<install_path>/Achronix-linux/libraries/examples/quickstart/<device_name>`	The netlist generated from Synplify Pro.
`quickstart_<device_name>_ioring.sdc`(2)	`C:\Program Files\Achronix CAD Environment\Achronix\examples\quickstart\<device_name>`	`<install_path>/Achronix-linux/libraries/examples/quickstart/<device_name>`	The .sdc file type stands for Synopsys Design Constraints. Typical usage is to constrain timing, power, and area. In this tutorial, a clock is defined.
`quickstart_<device_name>_ioring.pdc`(3)	`C:\Program Files\Achronix CAD Environment\Achronix\examples\quickstart\<device_name>`	`<install_path>/Achronix-linux/libraries/examples/quickstart/<device_name>`	The .pdc file type stands for Physical Design Constraints. Typical usage is to constrain I/O, placement, and routing. Since these are physical constraints, they are only used in the ACE tool as part of place-and-route. In this tutorial, pins are defined and their placement set.

Table Notes

1. Use the path shown if already familiar with Synplify Pro. Otherwise, use the path set for **Results Directory**.
2. If using a Speedcore device, this file is named `quickstart.sdc`.
3. If using a Speedcore device, the file is named `quickstart.pdc`.

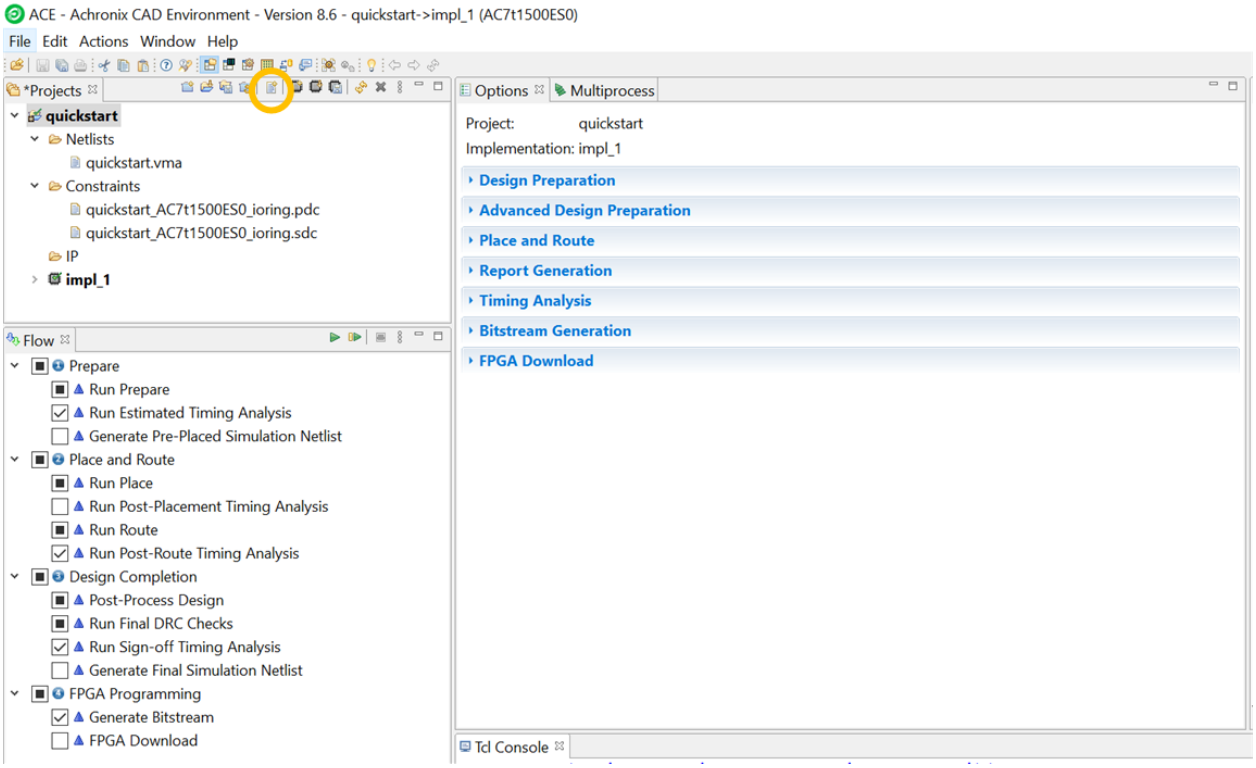

Figure 11: ***ACE Add files Button***

6. Select the **Design Preparation** section within the **Options** tab.

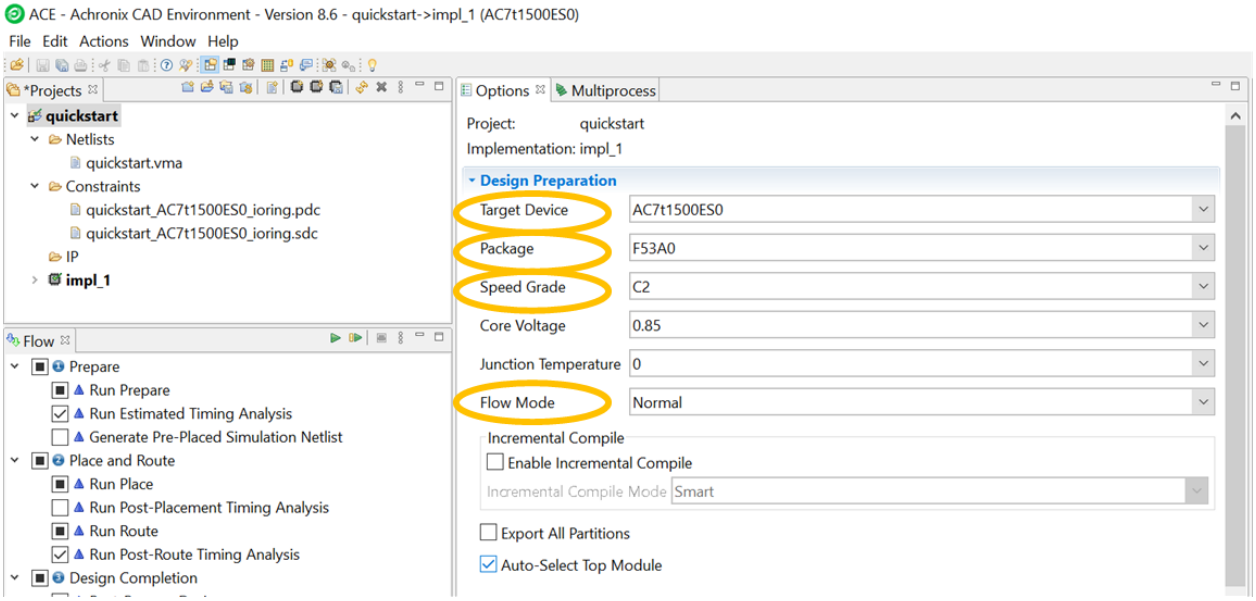

Figure 12: ***Design Preparation Section of Options Tab***

7. Set the **Target Device**, **Package**, and **Speed Grade** values.
   For **Speed Grade**, **C3** is the slowest, **C2** is in the middle, and **C1** is the fastest.
   Flow Mode can be set to **Evaluation**, **Normal**, or **Strict**. Of the three options, **Evaluation** results in the fastest time to produce a placed-and-routed design but bitstream generation requires either **Normal** or **Strict**. If the goal is to generate a bitstream, set **Flow Mode** to **Normal**.
   Other than the settings above, this design uses the default values. There are other fields that are useful in future designs:
   - **Package** – the package of the device.
   - **Core Voltage** – the operating voltage of the device.
   - **Junction Temperature** – the expected junction temperature. While many factors affect timing, this value directly affects which timing libraries are used.
8. Click the play button (circled) to run the place-and-route flow which, upon completion, produces a bitstream. If **Flow Mode** was set to **Evaluation**, ACE completes place-and-route and timing analysis, but does not produce a bitstream.

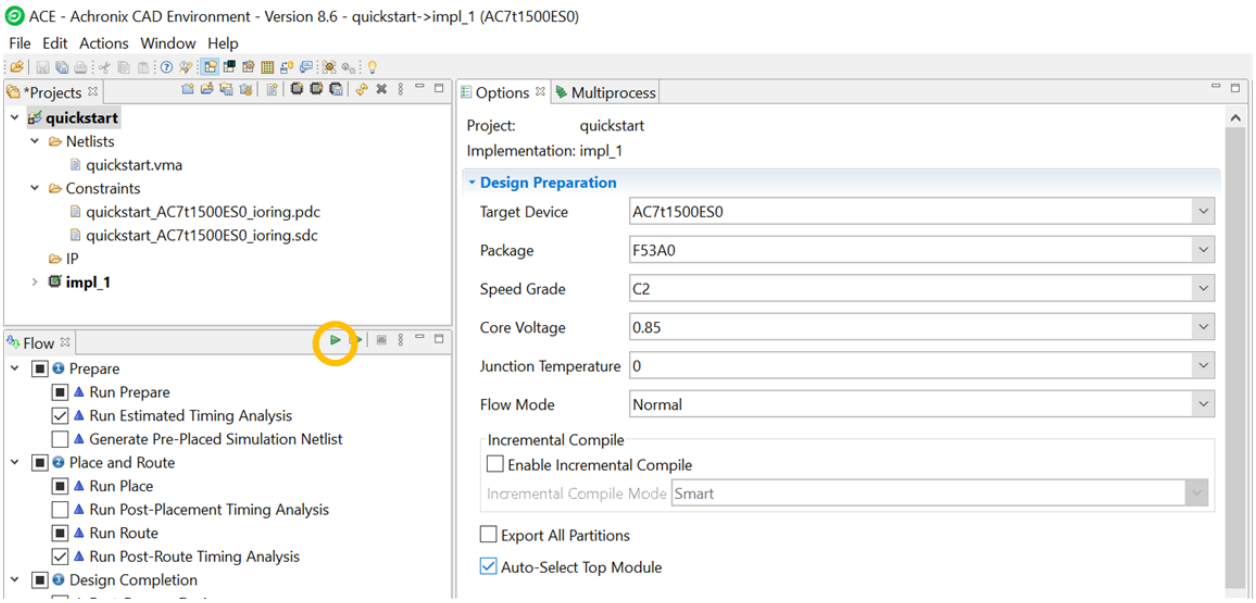

Figure 13: ***Place-and-route Play Button***

This completes the tutorial. A number of reports generated from the run are displayed such as timing, utilization, and pin assignments. If desired, the project can be saved by selecting **File → Save Project As...** from the menu.

See *ACE User Guide* (UG070) for more information on ACE.

Chapter - 6: FAQs

Why is my device not listed in Synplify Pro or ACE?

This is likely because the device overlay was not correctly installed. See ACE Installation and Licensing Guide (UG002).

Which simulators are supported?

VCS (Synopsys), QuestaSim (Mentor), Incisive (Cadence), and Riviera (Aldec). Achronix does not distribute or provide these simulator tools.

I am in a perspective, but I would like to add a window from a different perspective. How do I do this?

Most windows can be added by selecting **Window** → **Show View** from the menu. Select the desired window or select **Other** to see all windows that can be added.

I loaded an ACE project, but the information on its implementation is missing. How do I get that information?

Loading a project does not load its implementation run. After loading the project, right-click the implementation and select **Restore Implementation**.

How can I drive or observe signals when my device is running?

The Snapshot Debugger, found within ACE, can both drive and observe signals. See *Snapshot User Guide* UG016 for more details.

I am stuck and my question is not shown here. Who can I contact for help?

Contact Achronix at support@achronix.com

Revision History

Version	Date	Description
1.0	26 Jan 2022	• Initial Achronix release.