Getting Started User Guide (UG105)

All Achronix Devices



Copyrights, Trademarks and Disclaimers

Copyright © 2022 Achronix Semiconductor Corporation. All rights reserved. Achronix, Speedster and VectorPath are registered trademarks, and Speedcore and Speedchip are trademarks of Achronix Semiconductor Corporation. All other trademarks are the property of their prospective owners. All specifications subject to change without notice.

NOTICE of DISCLAIMER: The information given in this document is believed to be accurate and reliable. However, Achronix Semiconductor Corporation does not give any representations or warranties as to the completeness or accuracy of such information and shall have no liability for the use of the information contained herein. Achronix Semiconductor Corporation reserves the right to make changes to this document and the information contained herein at any time and without notice. All Achronix trademarks, registered trademarks, disclaimers and patents are listed at http://www.achronix.com/legal.

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	5
Chapter - 2: Achronix Tool Flow	6
Tool Flow	6
IP Flow	6
Chapter - 3: Launching Synplify Pro	8
GUI Mode	8
Batch Mode	9
Chapter - 4: ACE Execution Modes 1	0
GUI Mode	.0
Command-line Mode	11
Batch Mode	11
Chapter - 5: The Quickstart Design 1	.2
Introduction	2
Synthesizing with Synplify Pro	12
Creating Interface Subsystems and Running Place-and-Route	
Chapter - 6: FAQs	5
Why is my device not listed in Synplify Pro or ACE? 2	25
Which simulators are supported? 2	25
l am in a perspective, but I would like to add a window from a different perspective. How do I do this?	25
I loaded an ACE project, but the information on its implementation is missing. How do I get that information?	25
How can I drive or observe signals when my device is running?	25
l am stuck and my question is not shown here. Who can I contact for help?	25
Revision History	6

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.

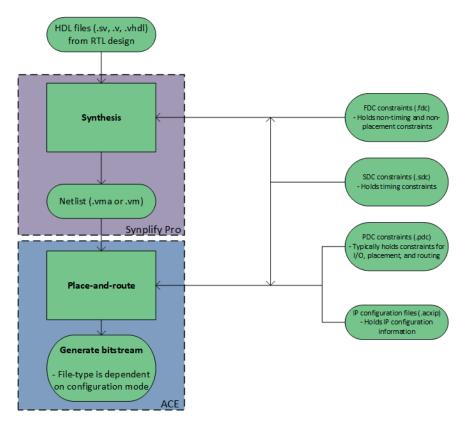


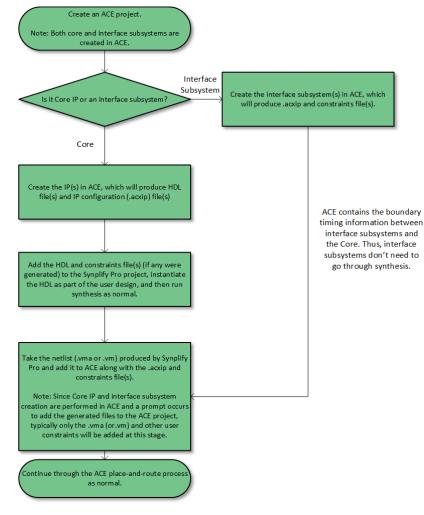
Figure 1: Achronix Design Flow

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. Interface subsystems – traditionally thought of as "hard IP". The I/O ring holds the various interface subsystems that surround the core, including GPIO, PLLs, memory interfaces, and more. Hence the term, "I/O ring".

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). Interface subsystems do not need to be synthesized in Synplify Pro because ACE contains the boundary timing information between them and the Core. However, constraints from interface 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 interface 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

It is 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</version>	<install_path>/fpga/<version>/bin</version></install_path>

When executed, the following window is displayed:

Synplify Pro (R) R-20	021.03X - [<no loaded="" projects="">]</no>						-		×
🤔 🛅 File Edit View Projec	t Run Analysis HDL-Analyst Options Window Web Help								_B×
🚯 🔮 🗉 🙋 🤘 🔦	D 🗈 🗤 🖓 🦀 🛞 🏵 🖬 🐺 🚺 😁 🚥 🧮	()+D							
◎ Run	Synplify Pro® Ready								
Dpen Project	Project Files	Project Status	Impleme	entation Direct	ory Process	View			
Close Project		Θ			Project S	Settings			
Add File		Project Name			Implementatio	in Name			
Change File					Run S	tatus			
Add Implementation		Job Name	Status	0 🛆 🖡	CPU Time	Real Time	Memory	Date/Tim	e
R Add P&R Implementation									
View Log									
Frequency (MHz):									
)[
🤔 <no projects=""></no>									
									www. B ×
License checkout: synplifyp License: synplifypro achror									
Licensed Vendor: achronix									
8		******							
TCL Script Messages									رت

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 does not 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</install_path>

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.



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 a device are included in the respective device overlay.

Synthesizing with Synplify Pro

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

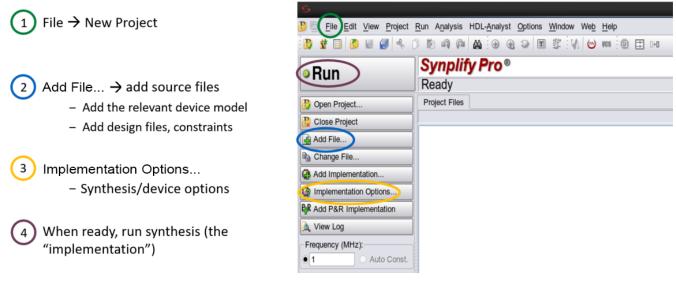


Figure 4: Overview of Synplify Pro Steps

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

 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 8). 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). Table 3: Synplify Project Files

Files	Typical Windows Path	Typical Linux Path	Purpose			
<device_name>_ synplify.sv</device_name>	C:\Program Files\Achronix CAD Environment\Achronix\ libraries\device_models	<install_path>/Achronix- linux/libraries/ device_models</install_path>	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⁾</device_ 	C:\Program Files\Achronix CAD Environment\Achronix\ examples\quickstart\rtl	<install_path>/Achronix- linux/libraries/examples/ quickstart/rtl</install_path>	Design RTL files.			
quickstart_ <device_ name>_ioring.sdc⁽³⁾</device_ 	C:\Program Files\Achronix CAD Environment\Achronix\ examples\quickstart\ <device_name></device_name>	<install_path>/Achronix- linux/libraries/examples/ quickstart/<device_name></device_name></install_path>	The .sdc file type stands for Synopsys Design Constraints. While SDC files are normally capable of also constraining timing, power, and area, the ACE flow differs in that SDCs are only used for timing constraints. In this case, a clock is defined. If the file name does not appear, select Files of type \rightarrow All Files (*.*).			
Table Notes 1. If using a Speedcore device, use the quickstart.v file instead. 2. If using the AC7t1550, also add the quickstart_AC7t1550_top.v file. 3. If using a Speedcore device, use the quickstart.sdc file instead.						

Tip

 \bigcirc

Files with "ioring" in their name are generated as part of the I/O ring 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.

Synplify Pro (R) R-20	21.03X - [C:/Syr	nopsys/fpga_R-2021.	03X/bin/quickstart.pr	* <out date="" of="">]</out>							
🕑 👘 File Edit View Projec	ct Run Analysis HI	DL-Analyst Options Window	Web Help								
🚯 🔮 🗐 🙆 😸 🎒 🤻		A 🕢 🕢 🗩 🖼 😤	🚺 🌝 🛤 💿 🎞 🖻]							
◎Run	Synplify	Pro®									_
- IXull	Ready										
Dpen Project	Project Files	Design Hierarchy			Project Status	Implement	tation Directory	Process	View		
Close Project	quickstart : re	v_1 - Achronix Speedst	er7t : AC7t1500ES0 : F5	53A0 : C1							
Add File		- C:\Synopsys\fpga_R-2021.03	3X\bin\quickstart.prj		Project Name				-	1	mpler
	Verilog	t1500ES0_synplify.sv [work] <s< td=""><td></td><td></td><td>Project Name</td><td></td><td></td><td></td><td></td><td></td><td>npier</td></s<>			Project Name						npier
	- 🔏 coun	nter.v [work]	ysv>		\odot						
Add Implementation		kstart_AC7t1500ES0.v [work]			Job Name		Status) 🛆		CPU
Implementation Options		onstraints (SDC) kstart_AC7t1500ES0_ioring.sdo			Compile Input		Complete				
Add P&R Implementation	- 🥥 rev_1				Premap		out-of-date				
View Log].				Map & Optimize	e	out-of-date				
Frequency (MHz):											
200 Auto Const.											
Automatic Compile Point											
Continue on Error											
FSM Compiler											
Resource Sharing											
Pipelining											
Retiming Automatic Compile Point											
Automatic Compile Point											
1											

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.

Synplify Pro (R) R-20	21.03X - [C:/Synopsys/fpga_R-2021.03X/bi	n/quickstart.prj * <out date="" of="">]</out>			
🕑 🗐 File Edit View Projec	t Run Analysis HDL-Analyst Options Window Web	Help			
🕒 🔮 🔲 🙋 📓 🍕	0 10 49 44 9 39 39 11 22 14 6) KOS 🕘 🎞 🖂			
◎Run	Synplify Pro® Ready				
Dpen Project	Project Files Design Hierarchy		Project Status Implementatio	on Directory Process View	w
Close Project	quickstart : rev_1 - Achronix Speedster7t : A	C7t1500ES0 : F53A0 : C1			Project
Add File	e 🗐 [quickstart] - C:\Synopsys\/pga_R-2021.03X\bin\qui	ckstart.prj	Project Name		Implementation Na
Change File	Verilog Varilog V	Bimplementation Options - quickst		Verilog GCC	Implementations:
Implementation Options	quickstart_AC7t1500ES0_ioring.sdc				rev_1
Adu r art ingrementation	@ rev_1	Technology: Part: Achronix Speedster7t V AC7t1500ES0	Package: S ▼ F53A0 ▼ (ipeed:	1 -
🔍 View Log		Actronix Speedster/t	▼ [P53A0 ▼]	52	1 -
Frequency (MHz):		Device Mapping Options			
Auto Const.		Option		Value	
Automatic Compile Point		Fanout Guide		10000	
Continue on Error		Disable I/O Insertion		v	
FSM Compiler Resource Sharing		Update Compile Point Timing Data			
Pipelining V		Automatic Read/Write Check Insertion for RAM			
Retiming		Retime Registers Forward			
Automatic Compile Point		Annotated Properties for Analyst		◀	
	1	mem_init_file		0	
		Resolve Mixed Drivers			
		Click on an option for description			
	•		OK Can	cel Help	SYNOPSYS*
guickstart.prj * <out date="" of=""></out>					

Figure 6: Device Tab of Implementation Options Window

• Implementation Results Tab – select where synthesis products are output via the Results Directory value. The Implementation Name value determines the final part of the Results Directory path string. This feature allows creating multiple synthesis runs in the same parent directory simply by changing the Implementation Name value for each run.

Synplify Pro (R) R-20	21.03X - [C:/Synopsys/fpga_R-2021.03X/b	in/quickstart.prj * <out date="" of="">]</out>		
🕑 🗐 File Edit View Projec	t Run Analysis HDL-Analyst Options Window Web	Help		
🚯 🔮 🔲 💋 😸 🎒 🤻	0 10 40 40 40 40 40 40 40 40 40 40 40 40 40	👏 WCS 🕘 🎞 🖽		
◎Run	Synplify Pro® Ready			
Doen Project	Project Files Design Hierarchy		Project Status Implementation Directory Process Vie	sw
Close Project	quickstart : rev_1 - Achronix Speedster7t : A	C7t1500ES0 : F53A0 : C1		Project
Add File	[quickstart] - C:\Synopsys\fpga_R-2021.03X\bin\qu	lickstart.prj	Project Name	Implementation Na
Change File		B Implementation Options - quicks	tart : rev_1	Implementations:
BR Add P&R Implementation	quickstart_AC7t1500ES0_ioring.sdc	Implementation Name:		rev_1
View Log		rev_1		- F
Frequency (MHz): 200 Auto Const Automatic Compile Point Continue on Error FSM Compiler Resource Sharing Pipelining V		Results Directory: C:Synopsystlpga_R-2021.03Xbinirev_1 Result Base Name: quickstart_results	Browse Result Format: vm v	
Retiming Automatic Compile Point		Optional Output File Options Write Mapped Verilog Netlist Write Mapped VHDL Netlist Write Vendor Constraint File	OK Cancel Help	Synopsys*

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.

🕑 🗐 File Edit View Proje	ject Run Analysis HDL-Analyst Options Window Web Help	
🐉 🔮 🔲 💋 😸 🎒 🭕	< D 10 4 6 6 3 9 8 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
[⊚] Run	Synplify Pro® Ready	
Copen Project	Project Files Design Hierarchy Project Status Implementation Directory Proc	cess View
Close Project	quickstart : rev_1 - Achronix Speedster7t : AC7t1500ES0 : F53A0 : C1	Proje
Add File	e Di (quickstarl) - C:Synopsyslipga_R-2021.03Xibiniquickstart.prj Project Name	Implementation N
Change File	Verlog Verlog	×
Implementation Options	E Constraints (SDC) Device Options Constraints Implementation Results Timing Report Verilog GCC	Implementations:
Add P&R Implementation Average Construction Automatic Compile Point Continue on Error Resource Sharing Pipelining Automatic Compile Point	System Verligg Push Tristates Allow Duplicate Modules Multiple File Compilation Unit Compiler Directives: e.g. SIZE:::8	
	Library Directories or Files:	edit> Synopsys* Pretabilistatis

Figure 8: Implementation Options Window Verilog Tab

The following tabs are not 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** (value 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** \rightarrow **Save As...** from the menu. See *Synthesis User Guide* (UG018) for more information on Synplify Pro and synthesis.

Creating Interface Subsystems and Running Place-and-Route

Follow these steps to process the Quickstart design in ACE:

 Launch ACE and create a new project (select File → Create Project... from the menu). See Launching ACE (see page 10) for details.

ACE - Achronix CAD Environment - Version 8.6		
File Edit Actions Window Help		
🔗 🔛 🐚 🗁 🖟 🗈 🖄 🕫 😰 🚰 🛅 🔐 💭 🖗 🌾 💡 🗇 🗢 🖉		
🕙 Projects 🛛 📫 🐸 🖓 🕼 📓 🖉 🖓 🖉 🖛 👘 👘	E Options 🛛 🕨 Multiprocess	
	Screate a New Project	– 🗆 X
	Project Creation	
	A project represents a collection of source files, flow settings, and flow outputs.	
	Project directory: C:/designs/quickstart_example	Browse
	Project name: quickstart	
δγ Flow ∞ ► ► = = = = = = = = = = = = = = = = =	Implementation name:	
✓ ■ ³ Prepare		
Run Prepare		
🗹 🔺 Run Estimated Timing Analysis		
Generate Pre-Placed Simulation Netlist		
✓ ■ ❷ Place and Route	⑦ Finish	Cancel
A Run Place		
A Run Post-Placement Timing Analysis		

Figure 9: 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. To see how the interface subsystems were created, see Creating Interface subsystems (see page 20). If only interested in the ACE flow and not interface subsystem creation, continue to the next step.
- 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):

Getting Started User Guide (UG105)

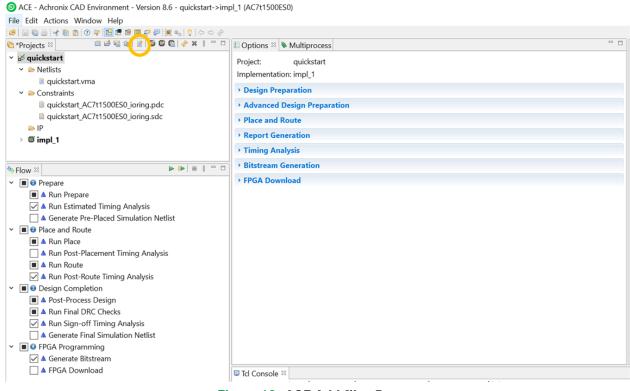


Figure 10: ACE Add files Button

Table 4: ACE Project Files

Files	Typical Windows Path	Typical Linux Path	Purpose
quickstart.vma Of quickstart.vm	C:\Program Files\Achronix CAD Environment\Achronix\ examples\quickstart\ <device_name>⁽¹⁾</device_name>	<install_path>/Achronix- linux/libraries/examples/ quickstart/<device_name></device_name></install_path>	The netlist generated from Synplify Pro.
quickstart_ <device_ name>_ioring.sdc⁽²⁾</device_ 	C:\Program Files\ Achronix CAD Environment\Achronix\ examples\quickstart\ <device_name></device_name>	<install_path>/Achronix- linux/libraries/examples/ quickstart/<device_name></device_name></install_path>	The .sdc file type stands for Synopsys Design Constraints. Only used for timing constraints. While SDC files are normally capable of also constraining power and area, the ACE flow differs in this regard. In this tutorial, a clock is defined.
quickstart_ <device_ name>_ioring.pdc⁽³⁾</device_ 	C:\Program Files\ Achronix CAD Environment\Achronix\ examples\quickstart\ <device_name></device_name>	<install_path>/Achronix- linux/libraries/examples/ quickstart/<device_name></device_name></install_path>	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, this file is named ${\tt quickstart.pdc}.$

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

O ACE - Achronix CAD Environment - Version 8.6 - quickstart->imp	ol_1 (AC7t1500ES0)	
File Edit Actions Window Help		
🎯 🔛 🗞 🎰 i 🛠 🗈 🗈 i 🕐 🛠 📴 🖪 🗃 📾 📰 🖧 🦃 🕅		
🖄 *Projects 🛛 📫 🐸 📽 📽 📓 📽 🕼 🔗 🗱 😫 " 🗖	E Options 🛛 🕨 Multiprocess	
 ✓ guickstart ✓ Potential Potential ✓ Potential ✓ Pote	Project: quickstart Implementation: impl_1	,
✓ ➢ Constraints	- Design Preparation	
quickstart_AC7t1500ES0_ioring.pdc	Target Device AC7t1500ES0	~
quickstart_AC7t1500ES0_ioring.sdc IP	Package F53A0	\checkmark
› @ impl_1	Speed Grade C2	~
8₀s Flow ⊠ ► ► ■ 8 □ □	Core Voltage 0.85	~
 ✓ I I I I I I I I I I I I I I I I I I I	Junction Temperature 0	~
 A Run Prepare A Run Estimated Timing Analysis 	Flow Mode Normal	~
Generate Pre-Placed Simulation Netlist	Incremental Compile	
✓ ■ e Place and Route	Enable Incremental Compile	
 A Run Place A Run Post-Placement Timing Analysis 	Incremental Compile Mode Smart	~
A Run Route	Export All Partitions	
 ✓ ▲ Run Post-Route Timing Analysis ✓ ■ Oesign Completion 	Auto-Select Top Module	

Figure 11: Design Preparation Section of Options Tab

8. Set the Target Device, Package, and Speed Grade values.

For **Speed Grade**, the value is marked on the package. **C3L** 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.

 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.

File Edit Actions Window Help	Ø ACE - Achronix CAD Environment - Version 8.6 - quickstart->impl_1 (AC7t1500ES0)						
 Projects II Projects II Projects II Projects II Project Quickstart Quickstart Quickstart Quickstart AC7t1500ES0_ioring.pdc Quickstart AC7tite Quickstart AC7t1500ES0_ioring.pdc Quicksta	File Edit Actions Window Help						
V jetuits V jetuits V jetuitstatt	🔗 🔄 🗞 🗁 🖟 🗈 🖞 🧭 🕐 📴 🖪 📾 🛄 🗗 🖗 🍕 🤉						
 Netlists quickstart.vma Constraints quickstart_AC7t1500ES0_ioring.pdc quickstart_AC7t1500ES0_ioring.sdc IP IP<!--</td--><td>🖄 *Projects 🛛 🔅 🧀 😂 📽 🕼 🕼 🕼 🤣 🗱 🗧 🗖</td><td>🗉 Options 🛛 🕨 Multipro</td><td>ocess</td><td></td>	🖄 *Projects 🛛 🔅 🧀 😂 📽 🕼 🕼 🕼 🤣 🗱 🗧 🗖	🗉 Options 🛛 🕨 Multipro	ocess				
 Netlists quickstart.wma Constraints quickstart_AC7t1500ES0_ioring.pdc quickstart_AC7t1500ES0_ioring.sdc IP Speed Grade C2 Core Voltage 0.85 Speed Grade C2 Core Voltage 0.85 Speed Grade Speed Grade Core Voltage 0.85 Speed Grade Speed Grade<!--</td--><td>✓</td><td>Project: quicks</td><td>tart</td><td>^</td>	✓	Project: quicks	tart	^			
Image: Problem Imag	 > Netlists 						
Constraints Quickstart_AC7t1500ES0_ioring.pdc Quickstart_AC7t1500ES0_ioring.sdc Pip IP IP IP	🗎 quickstart.vma						
Image: Package Imag	 Constraints 	 Design Preparation 					
 Package Package F53A0 Package F53A0 Speed Grade C2 Core Voltage 0.85 Junction Temperature Junction Temperature Junction Temperature Flow Mode Normal Flow Mode Normal Flow Mode Normal Incremental Compile Increment	quickstart_AC7t1500ES0_ioring.pdc	Target Device	AC7t1500ES0	~			
 Incremental Compile Increm	quickstart_AC7t1500ES0_ioring.sdc		55340				
Speed Glade Speed Glade Speed Glade Speed Glade Speed Glade Speed Glade Speed Glade Speed Glade Speed Glade Speed Glade Speed Glade <td>▷ IP</td> <td>Package</td> <td>F53A0</td> <td>~</td>	▷ IP	Package	F53A0	~			
 Brow 3 Brow 4 Brow 4 Contract of the pare A Run Prepare A Run Prepare A Run Prepare A Run Prepare A Run Prepare 0 Flow Mode Normal Flow Mode Normal Incremental Compile Inc	> @ impl_1	Speed Grade	C2	~			
 A Run Prepare A Run Estimated Timing Analysis A Generate Pre-Placed Simulation Netlist Place and Route Enable Incremental Compile Incremental Compile Incremental Compile Incremental Compile Export All Partitions A Run Post-Route Timing Analysis 	Image: Second secon	Core Voltage	0.85	~			
■ A Run Prepare Flow Mode Normal ▲ A Run Estimated Timing Analysis Incremental Compile ▲ Generate Pre-Placed Simulation Netlist Incremental Compile ■ ④ Place and Route Enable Incremental Compile ▲ Run Place Incremental Compile ▲ Run Post-Placement Timing Analysis Export All Partitions ✓ ▲ Run Post-Route Timing Analysis ✓ Auto-Select Top Module	✓ ■ ⊕ Prepare	Junction Temperature	0	×.			
✓ A Run Estimated Timing Analysis Incremental Compile ▲ Generate Pre-Placed Simulation Netlist Incremental Compile ■ A Run Place Enable Incremental Compile ▲ A Run Post-Placement Timing Analysis Incremental Compile ■ A Run Post-Placement Timing Analysis Export All Partitions ✓ A Run Post-Route Timing Analysis ✓ Auto-Select Top Module		-					
 ▼ ■ Place and Route ■ A Run Place □ A Run Post-Placement Timing Analysis ■ A Run Route □ Export All Partitions □ A Run Post-Route Timing Analysis □ A Run Post-Route Timing Analysis 	✓ ▲ Run Estimated Timing Analysis	Flow Mode	Normal	~			
A Run Place A Run Post-Placement Timing Analysis A Run Post-Route Timing Analysis ✓ A Run Post-Route Timing Analysis ✓ A Run Post-Route Timing Analysis ✓ A Run Post-Route Timing Analysis	☐ ▲ Generate Pre-Placed Simulation Netlist	Incremental Compile	2				
	✓ ■ ❷ Place and Route	Enable Increment	al Compile				
	🔳 🔺 Run Place	Incremental Compile	Mode Smart	~			
✓ ▲ Run Post-Route Timing Analysis	A Run Post-Placement Timing Analysis						
Auto-Select lop Module	A Run Route	Export All Partition	15				
✓ ■ Obsign Completion	🗹 🔺 Run Post-Route Timing Analysis	Auto-Select Top M	Indule				
	 Design Completion 	- Auto-Select top M	iouue				

Figure 12: 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** \rightarrow **Save Project As...** from the menu.

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

Creating Interface Subsystems

The quickstart design uses the following interface subsystems:

- Clock signals enter and exit the device via Clock I/O Banks. In this case, a Clock I/O Bank routes an
 incoming, external clock to a PLL.
- The PLL uses the incoming clock from the Clock I/O Bank (sys_clock) as the reference clock. The PLL synthesizes and outputs the clock used for the core logic (clk).
- A GPIO Bank outputs the counter data from the core to two LEDs (quickstart_gpio_bank_data[0] and quickstart_gpio_bank_data[1]).

👩 Tip

Interface subsystems *are not instantiated* in the user design. Signals are added going between the interface subsystems (which are located in the I/O ring) and core by listing them as part of the top-level design port list, after which ACE makes the connection during place-and-route.

The following steps summarize creating each interface subsystem:

1. Navigate to the IP Configuration Perspective. Perspectives are preset configurations of various windows in ACE.

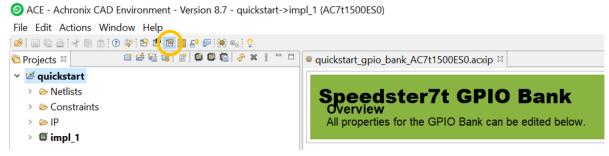


Figure 13: IP Configuration Perspective button

- In the IP Libraries window, under IO Ring, double-click Clock I/O Bank. The file name is the name of the configuration. A common practice is to create a directory called *acxip* at the same level as the ACE project, and then store all .acxip files in that directory. Configure the subsystem to match the screenshot below. Namely,
 - Set **Placement** to CLKIO_NE.
 - Enable the clock I/O on **Ball**, **N16** by checking the box on the same line under **Enabled**. Then, triple-click the **I/O Instance Name** associated with it and change it to sys_clock.
 - Other settings should default to the desired configuration. The screenshot below can be used as a reference for the other fields:

Over	edster7t Cloc view perties for the Clock I/O Bank		¢									
ank Conf	figuration											
Target	Device AC7t1500ES0											~
Placem	CLKIO_NE											×.
VREF S	ource Internal VDD											~
Rank V	oltage Level 1.5											
			1					1			1	
nabled	I/O Instance Name	Ball Name	Bump Name	Ball	Piffere	Signal Type	Port Direction	Reset C	I/O Standard	Vref	Input C	Ou ^
L 9	quickstart_clock_lo_bank_msio_p	CLKIO_NW_MSIO_P	CLKIO_NE_MSIO_P	010								
q	uickstart_clock_io_bank_msio_n	CLKIO_NW_MSIO_N	CLKIO_NE_MSIO_N	U17								
✓ s	ys_clk	CLKIO_NW_REFIO_P_0	CLKIO_NE_REFIO_P_0	N16		Clock •	INPUT -		LVCMOS_15 -		100	
q	quickstart_clock_io_bank_refio_n_(CLKIO_NW_REFIO_N_0	CLKIO_NE_REFIO_N_0	N17								
	wickstart clock in bank rofin n		CIVIO NE DECIO D 1	D16								×
												>
	OLL Bank Reset Configuration (CA											
		Internet Deart from EC										

Figure 14: Clock I/O Bank configuration

- 3. In the **IP Libraries** window, under **IO ring**, double-click **PLL**. Configure it to match the screenshot below. Namely,
 - a. Set **Placement** to PLL_NE_0.
 - b. Set **Reference Clock Name** to sys_clk, the clock entering the device via the Clock I/O Bank.
 - c. Set clkout0 Desired Frequency to 500.
 - d. Set Clkout Output 0 Port Name to clk.
 - e. Other settings should default to the desired configuration. The screenshot below can be used as a reference for the other fields.

quickstart_pll_AC7t1500ES0.	cxip 🛙			-	° 0
Speedster7 Overview Using basic properties,	t PLL this editor attempts to auto-configure the PLL wrapper. For more complicated configurations, see the Adv	anced PLL.			
✓ Target Device	AC7t1500ES0			~	^
✓ Placement	PLL_NE_0			~	
Reference Clock Name	sys_dk			~	
Reference Clock Frequency	100.0 MHz				
 Number of Clock Outputs 	1			~	
🛩 🗌 Force Integer Feedback	Divider for reduced jitter				
Clock Output 0					
Clkout0 Desired Frequencies	ency 500				
clkouto Achieved Fred	uency 500.0 MHz				
Desired - Achieved di	ference 0.0 MHz				
Percentage difference	0.0%				
Clkout Output 0 Port	Name clk				
🖌 🗹 Expose Clock Out	ut to Core Fabric				
✓ Phase Shift Factor	0			~	
VCO Frequency	8000.0 MHz				
V Expose PLL Lock Signal	to Core				~
0		Generate	<< Back	Next >>	
Configuration File Preview					

Figure 15: PLL configuration

- 4. In the **IP Libraries** window, under **IO ring**, double-click **GPIO Bank**. Configure it to match the screenshot below. Namely,
 - a. Set **Placement** to GPIO_N_B0.
 - b. Enable the I/O on Ball, W17 by checking the box on the same line under Enabled. Then, tripleclick the I/O Instance Name associated with it and change it to quickstart_gpio_bank_data [0]. Set Port Direction to OUTPUT.
 - c. Enable the I/O on Ball, W16 by checking the box on the same line under Enabled. Then, tripleclick the I/O Instance Name associated with it and change it to quickstart_gpio_bank_data [1]. Set Port Direction to OUTPUT.
 - d. Other settings should default to the desired configuration. The screenshot below can be used as a reference for the other fields.

quickstart_pll_AC7t1500ES0.acxip Speedster7t G Overview All properties for the GPIO Bar		·											
Bank Configuration													
✓ Target Device	AC7t1500ES0			~~	Placement		GPIO_N_B0					~	~
✔ DDR Mode	No			~ 🗸	SerDes Ratio		1					```	7
< 🗌 Rx Register Mode				~	Tx Register M	ode							
✔ Rx Edge Select	Positive edge			~ 🗸	Tx Edge Select		Positive edg	je					~
 Bank Clock Signal Name 	gpio_bank_1_clk_0			~	Valid GPIO DLL r	efclk Frequen	cy:						
Bank (Serial) Clock Frequency					Bank (Parallel) Clo	ock Frequenc	/						
GPIO DLL Reference Clock Period					GPIO DLL Phase	Shift Increme	nt						
✔ Bank Reset Source	Internal Reset from FCU			~ 🗸	Bank Global Rese	et Signal Nam	e bank_reset						~
✓ VREF Source	Internal VDD			~ 🗸	Bank Voltage Lev	rel	1.1					```	~
Enabled I/O Instance Name	Placement	Ball/Bump Name	Ball	Differe	Port Directio	n ad Clo	I/O Standard	Vref	Pull Type	Sle	Tx Targ	. Dri	^
gpio_bank_1_clk_0	GPIO_N_B0_CLK_0	GPIO_N0_BYTE0_BIT_4	AA17										
gpio_bank_1_clk_1	GPIO_N_B0_CLK_1	GPIO_N0_BYTE0_BIT_5	AA16										
quickstart_gpio_bank_data[0] GPIO_N_B0_DATA_0	GPIO_N0_BYTE0_BIT_0	W17		OUTPUT	•	LVCMOS_11 -		None •	3 •	28.8	• 13	
quickstart_gpio_bank_data[] GPIO_N_B0_DATA_1	GPIO_N0_BYTE0_BIT_1	W16		OUTPUT	-	LVCMOS_11 -		None •	3 -	28.8	13	~
Onfiguration File Preview								Generate	, , <	< Bac	k I	Next >	>>

Figure 16: GPIO Bank Configuration

5. After saving all the .acxip files, click the Generate button in *any* of the interface subsystem configuration windows to initiate the generation process for *all* configured subsystems. A common practice is to create a directory called *ioring* at the same level as the ACE project, and then store all generated interface subsystem files in that directory.

- 6. While each subsystem has a .acxip file associated with it, other files are also generated as part of the interface subsystem flow. In particular, multiple SDC files that cover different PVT points and simulation support files are created. Selecting **Add to active project** adds the generated files relevant to the ACE project (as opposed to generated files that support simulation), and is considered best practice. Doing so adds the following files:
 - <proj>_ioring.sdc: constraints for clock definitions of clocks to core
 - <proj>_ioring_timing_delays_<speedgrade_voltage_temp_corner>.sdc: constraints for timing delay of signals between core and I/O ring
 - <proj>_ioring.pdc: constraints for pin placement for core to interface subsystem interface
 - <proj>_ioring_util.xml: used for bitstream generation
 - <proj>_ioring_bitstream*.hex: the .hex file associated with each interface subsystem is taken and combined to produce two of these files (<proj>_ioring_bitstream0.hex and <proj>_ioring_bitstream1.hex). These two files are used as part of the final bitstream generation for the ACE project.
- 7. Continue with the quickstart steps (see page 17).

For more information on clocking, see the Speedster7t Clock and Reset Architecture Guide (UG083).

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 $Window \rightarrow Show View$ 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

Versio n	Date	Description
1.0	26 Jan 2022	Initial Achronix release.
1.1	19 Jul 2022	 Updated terminology: the I/O ring holds the various interface subsystems. Added instructions on creating the interface subsystems used in the Quickstart de The Device Speed option C3 is being deprecated and replaced by C3L.