

Vivado Design Suite Tutorial

Design Analysis and Closure Techniques

UG938 (v2021.1) July 14, 2021





Revision History

The following table shows the revision history for this document.

Section	Revision Summary					
07/14/2021 \	/ersion 2021.1					
Lab 2: Increasing Design Performance Using Report QoR Suggestions	Added clarifications.					
Lab 3: Running ML Strategies	Added clarifications.					
Lab 4: Intelligent Design Runs	New section added.					



Table of Contents

Revision History	2
Tutorial Overview	5
Introduction	5
Tutorial Description	5
Software Requirements	5
Locating Tutorial Design Files	5
Navigating Content by Design Process	6
Lab 1: Setting Waivers with the Vivado IDE	7
Introduction	7
Step 1: Starting the Vivado IDE	7
Step 2: Generating the CDC Report	8
Step 3: Waiving a Single CDC Violation	
Step 4: Generating a Report for Waived Violations	13
Step 5: Generating a Text Report with Details for Waived Violations	14
Step 6: Waiving Multiple CDC Violations	15
Step 7: Exporting Waivers	18
Step 8: Using the create_waiver Command	19
Step 9: Waiving Multiple CDC Violations	20
Step 10: Waiving Multiple DRC Violations	22
Step 11: Generating a Summary Report for Waived Violations	27
Step 12: Using Waiver Commands	30
Summary	31

Lab 2: Increasing Design Performance Using Report QoR

Suggestions	
Introduction	
Step 1: Understanding the Design	
Step 2: Running Report QoR Suggestions	
Step 3: Understanding the Report	
Step 4: Run with Suggestions	
Step 5: Accumulating Suggestions	46



Summary	
Lab 3: Running ML Strategies	49
Introduction	
Step 1: Generating an ML Strategy RQS File	49
Step 2: Creating ML Strategy Runs	51
Summary	53
Lab 4: Intelligent Design Runs	54
Introduction	
Step 1: Creating Intelligent Design Runs	54
Step 2: Navigating Intelligent Design Runs	57
Step 3: Analyzing the Reports and Log File	59
Step 4: Final Analysis	62
Summary	64
Appendix A: Additional Resources and Legal Notices	
Please Read: Important Legal Notices	65



Tutorial Overview

Introduction

This tutorial uses the Vivado[®] design rules checker (report_drc), clock domain crossing checker (report_cdc), and quality of results enhancer (report_qor_suggestions) to analyze example designs for issues, and shows you how to take corrective actions. It also outlines how to run ML strategies and Intelligent Design Runs (IDRs).

Tutorial Description

Lab 1 walks you through creating waivers for CDC, methodology, and DRC violations.

Lab 2 is a guide to using the report_qor_suggestions (RQS) command.

Note: The designs used in this tutorial are intended to exhibit issues for demonstration purposes, and should not be used as a reference for designs outside this tutorial.

Software Requirements

This tutorial requires that the 2021.1 Vivado[®] Design Suite software release or later is installed.

For a complete list of system and software requirements, see the Vivado Design Suite User Guide: Release Notes, Installation, and Licensing (UG973).

Locating Tutorial Design Files

- 1. Download the reference design files from the Xilinx[®] website.
- 2. Extract the ZIP file contents into any write-accessible location.

This tutorial refers to the location of the extracted ZIP file contents as <Extract_Dir>.



Navigating Content by Design Process

Xilinx[®] documentation is organized around a set of standard design processes to help you find relevant content for your current development task. All Versal[™] ACAP design process Design Hubs can be found on the Xilinx.com website. This document covers the following design processes:

- Hardware, IP, and Platform Development: Creating the PL IP blocks for the hardware platform, creating PL kernels, functional simulation, and evaluating the Vivado[®] timing, resource use, and power closure. Also involves developing the hardware platform for system integration. Topics in this document that apply to this design process include:
 - Lab 2: Increasing Design Performance Using Report QoR Suggestions
 - Lab 3: Running ML Strategies
 - Lab 4: Intelligent Design Runs



Lab 1

Setting Waivers with the Vivado IDE

Introduction

In the Vivado[®] Design Suite, you can use the waiver mechanism to waive clock domain crossing (CDC), design rule check (DRC), or methodology check violations. After a violation is waived, it is no longer reported by the <code>report_cdc</code>, <code>report_drc</code>, or <code>report_methodology</code> commands. Waived checks are also filtered out from the mandatory DRCs run at the start of the implementation commands, such as <code>opt_design</code>, <code>place_design</code>, and <code>route_design</code>. For more information, see this link in the Vivado Design Suite User Guide: Design Analysis and Closure Techniques (UG906).

IMPORTANT! The content of the waiver is built with the objects that exist when the waiver is created. However, if an instance referenced inside a waiver is replicated by Vivado[®], the replicated instance is automatically added to the waiver and saved in subsequent checkpoints and XDC.

This lab shows how to set waivers with the Vivado integrated design environment (IDE) using both menu commands and the Tcl Console. The lab focuses on CDC waivers, but the methods for waiving DRC and methodology violations are similar.

Step 1: Starting the Vivado IDE

This lab uses a Vivado design checkpoint (.dep file), which is a snapshot of a design. When you launch the Vivado IDE using a design checkpoint, a subset of the Vivado IDE functionality is available.

TIP: To launch the Vivado Tcl Shell on Windows, select Start \rightarrow All Programs \rightarrow Xilinx Design Tools \rightarrow Vivado <version> \rightarrow Vivado <version> Tcl Shell.

1. From the command line or the Vivado Tcl Shell, change to the directory where the lab materials are stored:

```
cd <Extract_Dir>/src/Lab1
```

2. To start the Vivado IDE with the design checkpoint loaded, enter the following:

```
vivado my_ip_example_design_placed.dcp
```



TIP: You can disregard the critical warnings about the unbounded GT locations.

Step 2: Generating the CDC Report

In this step, you generate the CDC report to view the associated CDC violations.

- 1. Select **Reports** \rightarrow **Timing** \rightarrow **Report CDC**.
- 2. In the Report CDC dialog box, leave the default settings as-is, and click OK.

À Report CDC X
Report clock domain crossing (CDC) paths between clocks, even if set_false_path or set_clock_groups constraints have been applied.
Res <u>u</u> lts name: cdc_1
Clocks
Erom:
<u>T</u> o:
Report
Report from cells:
Waivers
Apply waivers
Report only waived paths
☐ Ignore all waivers
File Output
Export to file:
<u>O</u> verwrite <u>Append</u>
Options
Suspend message limits during command execution
Ignore command errors (quiet mode)
Command: report_cdc-name cdc_1
✓ Open in a new tab Open in Timing Analysis layout
OK Cancel

The Summary (by clock pair) section of the CDC Report appears as follows.



Q 풒 ♦ C	Q Summa	ry (by clock pair)								
General Information	Severity ^1	Source Clock	Destination Clock	CDC Type	Exceptions	Endpoints	Safe	Unsafe	Unknown	No ASYNC_REG
Summary (by clock pair)	Critical	my_ip_glblclk	my_ip_axi_aclk	No Common Primary Clock	False Path	2	1	1	0	0
Summary (by type)	Oritical	my_ip_axi_aclk	my_ip_drpclk	No Common Primary Clock	False Path	2	0	2	0	0
Summary (by waived endpoints)	Oritical	my_ip_axi_aclk	my_ip_glblclk	No Common Primary Clock	False Path	942	12	351	579	185
CDC Details (928)	Info	my_ip_drpclk	my_ip_axi_aclk	No Common Primary Clock	False Path	1	1	0	0	0
my_ip_drpclk to my_ip_axi_aclk (1)	1 Info	input port clock	my_ip_drpclk	No Common Primary Clock	False Path	2	2	0	0	0
my_ip_glblclk to my_ip_axi_aclk (2)	Info	my_ip_glblclk	my_ip_drpclk	No Common Primary Clock	False Path	6	6	0	0	0
input port clock to my_ip_drpclk (2)	Info	my_ip_drpclk	my_ip_glblclk	No Common Primary Clock	False Path	2	2	0	0	0

The Summary (by CDC type) section appears as follows.

Q ≚ ♦ C		Q Summa	ry (by type)		
General Information	<u>^</u>	Severity ^1	ID	Count	Description
Summary (by clock pair)		Oritical	CDC-1	536	1-bit unknown CDC circuitry
Summary (by type)		Oritical	CDC-4	4	Multi-bit unknown CDC circuitry
Summary (by waived endpoints)		Oritical	CDC-10	187	Combinational logic detected before a synchronizer
CDC Details (928)		\rm Critical	CDC-11	2	Fan-out from launch flop to destination clock
my_ip_drpclk to my_ip_axi_aclk (1)		Oritical	CDC-13	170	1-bit CDC path on a non-FD primitive
my_ip_glblclk to my_ip_axi_aclk (2)		\rm Oritical	CDC-14	5	Multi-bit CDC path on a non-FD primitive
input port clock to my_ip_drpclk (2)		👴 Warning	CDC-15	10	Clock enable controlled CDC structure detected
my_ip_axi_aclk to my_ip_drpclk (2)		1 Info	CDC-3	9	1-bit synchronized with ASYNC_REG property
my_ip_glblclk to my_ip_drpclk (6) my_ip_axi_aclk to my_ip_glblclk (913)		1 Info	CDC-9	5	Asynchronous reset synchronized with ASYNC_REG property

Step 3: Waiving a Single CDC Violation

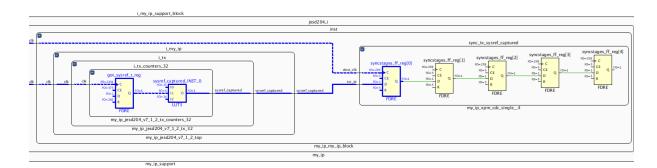
The $my_{ip_glblclk}$ to $my_{ip_axi_aclk}$ clock pair includes one Critical CDC-10 violation due to combinational logic on the CDC path. This step covers how to waive the CDC-10 violation.

Tcl Console Messages Timing ×									? _ 🗆
Q ₹ ≑ C	Q 🗄 🔇	my_ip	_glblclk to my_ip_axi_aclk			🖌 🌗 Critical warning (1)	🖌 🌖 Warning (0) 🛛 🖌	1 🚺 Info (1) Hide A
General Information	Severity ^1	ID	Description	Depth	Exception	Source (From)	Destination (To)		Category
Summary (by clock pair)	Critical	CDC-10	Combinatorial logic detected before a synchronizer	5	False Path	i_my_ip_supporysref_r_reg/C	i_my_ip_suppors_ff_r	eg[0]/D	Unsafe
Summary (by type)	Info	CDC-3	1-bit synchronized with ASYNC_REG property	5	False Path	i_my_ip_supporot_sync_reg/C	i_my_ip_suppos_ff_re	eg[0]/D	Safe
CDC Details (928) my_ip_drpclk to my_ip_axi_aclk (1) my_ip_glblclk to my_ip_axi_aclk (2) input port clock to my_ip_drpclk (2)									

1. To view a schematic of the violation, select the CDC-10 row in the CDC Report, and click the Schematic toolbar button 3.

Note: Alternatively, you can press **F4** to generate the schematic. However, using the toolbar button provides a more detailed schematic that includes all the levels of the downstream synchronizer.





- 2. To waive the violation, select the **CDC-10** row in the CDC Report, right-click, and select **Create Waiver**.
- 3. In the Create Waiver dialog box, enter a description, and click **OK**.

🝌 Create Wa	iver X
Create waiver fo	or 1 cdc path
User:	Xilinx
Description:	This is a safe CDC per review with the team
Tags:	
Tcl Command	Preview
Q	
create_waiver	r-type CDC -id CDC-10 -from [get_pins i_my_ip_support_block/jesd204_i/inst/i_my_ip/i_tx/i_tx_counters
<	>
?	OK Cancel

IMPORTANT! A waiver tracks the date the waiver was added, the user that added the waiver, and a description of why the violation was waived. The date is automatically added by the system. The Tags field is an optional description or list of keywords that can be used for documentation purposes.

4. After the waiver is created, check the CDC Report.

To indicate that a waiver was created, the CDC-10 row is gray and disabled.

Note: Rows are only disabled in the Report CDC result window from which the waivers were created.



Report is out of date because path waivers we	re cha	nged. Rerun							
. <u>¥</u> ≑ C		Q H .	©_ my_ip_	glblclk to my_ip_axi_aclk			🕙 🕒 Critical warning (1) 🛛 🖌 🔮 V	Varning (0) 🕑 🚯 Info (1) 🛛 H	lide All
General Information	^	Severity ^1	ID	Description	Depth	Exception	Source (From)	Destination (To)	Category
Summary (by clock pair)		Critical		Combinational logic detected before a synchronizer	5	False Path	i_my_ip_supportsysref_r_reg/C	i_my_ip_supportes_ff_reg[0]/D	Unsafe
Summary (by type)		Info	CDC-3	1-bit synchronized with ASYNC_REG property	5	False Path	i_my_ip_supporot_sync_reg/C	i_my_ip_suppores_ff_reg[0]/D	Safe
Summary (by waived endpoints)									
CDC Details (928)									
my_ip_drpclk to my_ip_axi_aclk (1)									
my_ip_glblclk to my_ip_axi_aclk (2)									
input port clock to my_ip_drpclk (2)									

5. To see the impact of the CDC-10 waiver, select **Reports** → **Timing** → **Report CDC** to rerun Report CDC.

Note: When a waiver is created or deleted, you must rerun Report CDC, Report DRC, or Report Methodology to see the updated results.

6. See the CDC Report to view the updated information.

The differences from the previous Summary by clock pair and Summary by type sections are highlighted in red in the following figures.

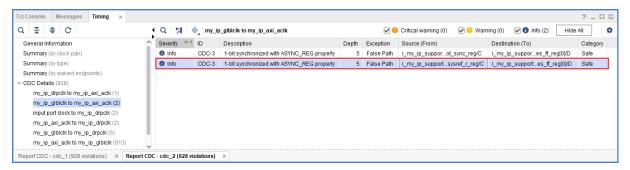
Q 素 ♦ C	Q Summar	y (by clock pair)									
General Information	Severity ^1	Source Clock	Destination Cl	lock CDC	Туре	Exception	s Endpoints	Safe	Unsafe	Unknown	No ASYNC_REG
Summary (by clock pair)	Critical	my_ip_axi_aclk	my_ip_drpclk	No C	Common Prim	ary Clock False Pa	n 2	0	2	0	(
Summary (by type)	Critical	my_ip_axi_aclk	my_ip_glblclk	No C	Common Prim	ary Clock False Pa	n 942	. 12	351	579	18
Summary (by waived endpoints)	Info	my_ip_drpclk	my_ip_axi_ad	i_aclk No Common Prima		ary Clock False Pa	n 1	1	0	0	
V CDC Details (928)	Info	my_ip_glblclk	my_ip_axi_ad	lk No C	Common Prim	ary Clock False Pa	n 2	2	0	0	(
my_ip_drpclk to my_ip_axi_aclk (1)	 Info 	input port clock	my_ip_drpclk	No C	Common Prim	ary Clock False Pa					
my_ip_glblclk to my_ip_axi_aclk (2) input port clock to my_ip_drpclk (2)	 Info 	my_ip_glblclk	my_ip_drpclk		Common Prim	-					
my_ip_axi_aclk to my_ip_drpclk (2)	 Info 	my_ip_drpclk	my_ip_glblclk	No C	Common Prim	ary Clock False Pa	n 2	2	0	0	
Tel Console Messages Tin	ning ×										
	ning ×	۰Q.	Summary (by type)							
Q ¥ ≑ C	ning ×					Description					
Q X ♦ C General Information	ning ×	Sever	ity ^1 ID)	Count	Description					
Q ≚ ≑ C	ning ×		ity ^1 ID			Description 1-bit unknowr	CDC circui	itry			
Q X ♦ C General Information	ning ×	Sever	ity ^1 ID itical C)	Count			· ·	,		
Q Image: A state of the st		Sever Cr Cr	ity ^1 ID itical C itical C	DC-1	Count 536	1-bit unknowr	wn CDC ci	rcuitry		synchror	izer
Q X ♦ C General Information Summary (by clock pair) Summary (by type) Summary (by waived endpoints)		Sever Cr Cr	ity ^1 ID itical C itical C itical C	DC-1 DC-4	Count 536 4	1-bit unknowr Multi-bit unkn	wn CDC ci I logic dete	rcuitry cted b	efore a		nizer
Q X ♦ C General Information Summary (by clock pair) Summary (by type) Summary (by waived endpoints)	5)	Sever G Cr Cr Cr Cr Cr Cr Cr	ity ^1 ID itical C itical C itical C itical C	DC-1 DC-4 DC-10	Count 536 4 186	1-bit unknown Multi-bit unkn Combination	own CDC ci I logic dete aunch flop t	rcuitry cted b to des	efore a tination		izer
Q X ♦ C General Information Summary (by clock pair) Summary (by type) Summary (by waived endpoints ~ CDC Details (928)	;) _aclk (1)	Sever G Cr Cr Cr Cr Cr Cr Cr	ity ^1 ID itical C itical C itical C itical C itical C	DC-1 DC-4 DC-10 DC-11	Count 536 4 186 2	1-bit unknown Multi-bit unkn Combination Fan-out from	wn CDC ci Il logic dete aunch flop t n on a non-f	rcuitry cted b to des FD pri	efore a tination mitive	clock	izer
Q X ♦ C General Information Summary (by clock pair) Summary (by type) Summary (by waived endpoints ~ CDC Details (928) my_ip_drpclk to my_ip_axi	s) _aclk (1) _aclk (2)	Sever Cr Cr Cr Cr Cr Cr Cr Cr Cr C	ity ^1 ID itical C itical C itical C itical C itical C itical C	DC-1 DC-4 DC-10 DC-11 DC-13	Count 536 4 186 2 170	1-bit unknown Multi-bit unknown Combination Fan-out from 1-bit CDC pat	own CDC ci I logic deter aunch flop t n on a non-f path on a no	rcuitry cted b to des FD pri on-FD	efore a tination mitive primitiv	ve	
Q X ♦ C General Information Summary (by clock pair) Summary (by type) Summary (by waived endpoints ✓ CDC Details (928) my_ip_drpclk to my_ip_axi_ my_ip_glblclk to my_ip_axi	3) _aclk (1) _aclk (2) rpclk (2)	Sever Cr Cr Cr Cr Cr Cr Cr Cr Cr C	ity ^1 ID itical C itical C itical C itical C itical C itical C itical C anning C	DC-1 DC-4 DC-10 DC-11 DC-13 DC-14	Count 536 4 186 2 170 5	1-bit unknown Multi-bit unknown Combinationa Fan-out from 1-bit CDC pat Multi-bit CDC	wn CDC ci I logic deter aunch flop t n on a non-f path on a no controlled C	rcuitry cted b to des FD pri on-FD CDC s	efore a tination mitive primitiv tructure	ve detected	
Q X ♦ C General Information Summary (by clock pair) Summary (by type) Summary (by waived endpoints ✓ CDC Details (928) my_ip_drpclk to my_ip_axi, my_ip_glblclk to my_ip_axi, input port clock to my_ip_dr	s) _aclk (1) _aclk (2) rpclk (2) Irpclk (2)	 Sever Ci <l< td=""><td>ity ^1 ID itical C itical C itical C itical C itical C itical C itical C arning C fo C</td><td>DC-1 DC-4 DC-10 DC-11 DC-13 DC-13 DC-14 DC-15</td><td>Count 536 4 186 2 170 5 5</td><td>1-bit unknown Multi-bit unkn Combination Fan-out from 1-bit CDC pat Multi-bit CDC Clock enable</td><td>wn CDC ci I logic dete aunch flop t n on a non-l path on a no controlled C ized with AS</td><td>rcuitry cted b to des FD pri on-FD CDC s SYNC</td><td>efore a tination mitive primitiv tructure _REG p</td><td>ve detected roperty</td><td>1</td></l<>	ity ^1 ID itical C itical C itical C itical C itical C itical C itical C arning C fo C	DC-1 DC-4 DC-10 DC-11 DC-13 DC-13 DC-14 DC-15	Count 536 4 186 2 170 5 5	1-bit unknown Multi-bit unkn Combination Fan-out from 1-bit CDC pat Multi-bit CDC Clock enable	wn CDC ci I logic dete aunch flop t n on a non-l path on a no controlled C ized with AS	rcuitry cted b to des FD pri on-FD CDC s SYNC	efore a tination mitive primitiv tructure _REG p	ve detected roperty	1

You can also view a summary with the list of waived endpoints.



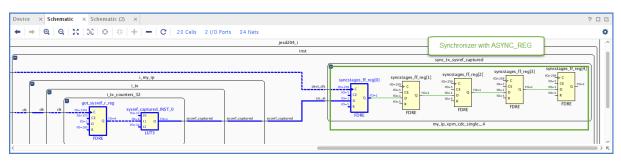
Tcl Console Messages Timing x			
Q ¥ ≑ C	4	Q. Sun	nmary (by waived endpoints)
General Information	2	ID	Waived Endpoints
Summary (by clock pair)		CDC-10	1
Summary (by type)			
Summary (by waived endpoints)			
V CDC Details (928)			
my_ip_drpclk to my_ip_axi_aclk (1)			
my_ip_glblclk to my_ip_axi_aclk (2)			
input port clock to my_ip_drpclk (2)			
my_ip_axi_aclk to my_ip_drpclk (2)			
my_ip_glblclk to my_ip_drpclk (6)			
my_ip_axi_aclk to my_ip_glblclk (913)	~		
Report CDC - cdc_1 (928 violations) × Re	port	CDC - cdc	_2 (928 violations) \times

The detailed section for the $my_{ip_glblclk}$ to $my_{ip_axi_aclk}$ CDC shows that the Critical CDC-10 was replaced with an Info CDC-3.



7. Select the new CDC-3 row, and click the Schematic toolbar button ¹. Double-click the **Q** pin of the output register to expand the schematic to match what is shown in the following figure.

The CDC path includes a 5-level synchronizer on the output of the selected destination register. This is the reason the CDC-10 was replaced with CDC-3 for this topology, as shown in the following figure.





IMPORTANT! By default, Report CDC only reports a single violation per endpoint and per clock pair.
 When multiple violations apply to the same endpoint, only the violation with the highest precedence is reported. Because CDC-10 has a higher precedence than CDC-3, only CDC-10 is reported when both CDC-10 and CDC-3 apply to the same endpoint. For more information on CDC rules precedence, see this link in the Vivado Design Suite User Guide: Design Analysis and Closure Techniques (UG906).

TIP: To report all of the CDC violations for each endpoint regardless of the precedence rules, use the command line option *-all_checks_per_endpoint*. However, unsafe rules are not reported on a register if at least one safe rule on the same register is detected.

Step 4: Generating a Report for Waived Violations

You can generate a report for the CDC, DRC, or methodology check violations that were waived. This step shows how to generate a report for waived CDC violations using the Tcl Console as well as the Vivado IDE menu commands.

Generating a Text Report for Waived Violations

1. In the Tcl Console, enter:

report_cdc -waived

 In the CDC report, verify that a single CDC-10 violation is listed, because only one waiver was created.



Generating a Vivado IDE CDC Report for Waived Violations

- 1. Select **Reports** \rightarrow **Timing** \rightarrow **Report CDC**.
- 2. In the Report CDC dialog box, enable Report only waived paths, and click OK.
- 3. In the CDC Report, check the Summary (by clock pair) and CDC Details to verify that a single CDC-10 violation is listed.



Note: The icon next to the violation shows that the violation was waived *8*.

Tcl Console Messages Timing ×													? _ 🗆
Q ¥ ♦ C	Q Summar	y (by clock pair	r)										
General Information	Severity ^1	Source Clock	Destination Clock	CDC Type	Excepti	ons Endpo	ints Safe	Unsafe	Unknown	No ASYNC	_REG		
Summary (by clock pair)	Oritical	my_ip_glblclk	my_ip_axi_aclk	No Common Primary Clock	False F	ath	1	0	1 ()	0		
Summary (by type)													
Summary (by waived endpoints)													
CDC Details (1)													
my_ip_glblclk to my_ip_axi_aclk (1)													
Report CDC - cdc_1 (928 violations) × Rep	ort CDC - cdc_2 (928	violations) >	Report CDC - cdc_	3 (1 waived) ×									
	oort CDC - cdc_2 (928	i violations) >	< Report CDC - cdc_	3 (1 waived) ×									2 _ 0
Tcl Console Messages Timing ×			< Report CDC - cdc_			¥ 🔒 (Critical warni	ng (1)	🖌 🕕 Warnir	1g (0) 🕑	 Info (0) 	Hide	
Tcl Console Messages Timing ×	Q H «	, my_ip_glbl			Depth E	0 -	Critical warni Source (Fro		_	ng (0) 🕑 Destination		Hide /	
Ccl Console Messages Timing ×	Q H «	my_ip_glbl	clk to my_ip_axi_aclk			Exception	Source (Fro	m)	_	Destination	(To)		All
Tct Console Messages Timing × Q X ♦ C General Information	Q H « Severity ^1	my_ip_glbl	clk to my_ip_axi_aclk	· · · · ·		Exception	Source (Fro	m)	_	Destination	(To)		All d Category
Ccl Console Messages Timing × Q	Q H « Severity ^1	my_ip_glbl	clk to my_ip_axi_aclk	· · · · ·		Exception	Source (Fro	m)	_	Destination	(To)		All
Ccl Console Messages Timing × Q X Image: Color of the start of	Q H « Severity ^1	my_ip_glbl	clk to my_ip_axi_aclk	· · · · ·		Exception	Source (Fro	m)	_	Destination	(To)		All
Tcl Console Messages Timing × Q X ★ + C General Information Summary (by clock pair) Summary (by type) Summary (by walved endpoints)	Q H « Severity ^1	my_ip_glbl	clk to my_ip_axi_aclk	· · · · ·		Exception	Source (Fro	m)	_	Destination	(To)		Category
Tcl Console Messages Timing × Q ₹ ♦ C General Information Summary (by clock pair) Summary (by type) Summary (by waived endpoints) ∨ CDC Details (1)	Q H « Severity ^1	my_ip_glbl	clk to my_ip_axi_aclk	· · · · ·		Exception	Source (Fro	m)	_	Destination	(To)		All
Q ★ ♦ C General Information Summary (by clock pair) Summary (by type) Summary (by waived endpoints) ~ CDC Details (1)	Q H Seventy ^1	my_ip_glbl ID De CDC-10 Co	cik to my_ip_axi_acik escription mbinational logic dete	ected before a synchronizer		Exception	Source (Fro	m)	_	Destination	(To)		All d Category

Step 5: Generating a Text Report with Details for Waived Violations

In this step, you generate text reports with additional details, including a list of all of the rules and all of the violations regardless of the waivers.

Generating a List of Rules with Waived Violations

1. In the Tcl Console, enter:

report_cdc -details -show_waiver

2. Verify that the my_ip_glblclk to my_ip_axi_aclk CDC-10 violation is waived and the two CDC-3 violations are not waived.

Note: In the text report, all of the rules are reported, whether they were waived or not. The Waived column indicates the status of the rule.

CDC Re	port									
ID	Severity	Count	Description							
CDC-13 CDC-14	Critical Critical	10 4 5 186 2 170 5	1-bit usinown CDC circuitsy 1-bit usinown CDC circuitsy Multi-bit unknown CDC circuitsy Asynchronour reset usynchronizad with RONC_REG pr Combinatorial logic detected before a synchroniza Farrout from Lanch flog to destination clock 1-bit CDC path on a nor-FD primitive Libck enable controlled CDC structure detected	operty r						
CDC-10	1									
Destin	Clock: my_ ation Clock >e: No Comm	: ny_ip	_axi_aclk				\$			
Row I) Seve	erity I	lescription	Depth	Exception	Source (From)			Destination (To)	Waived
2 C	0C-3 Info 0C-3 Info 0C-10 Crit) 1	-bit synchronized with ASYNC_REG property -bit synchronized with ASYNC_REG property combinatorial logic detected before a synchronizer	5 5 5	False Path False Path False Path	i_my_ip_suppor i_my_ip_suppor i_my_ip_suppor	t_b) t_ t_b.	s_32/got_sync_reg/C _32/got_sysref_r_reg/C s_32/got_sysref_r_reg/C	i_my_ip_support_block/jesd204_i/inst/sync_tx_sync/syncstages_ff_reg[0]/D i_my_ip_support_block/jesd204_i/inst/sync_tx_synrf_captured/syncstages_ff_reg[0]/D i_my_ip_support_block/jesd204_i/inst/sync_tx_synrf_captured/syncstages_ff_reg[0]/D	N N Y



Generating a List of All Violations Regardless of the Waivers

1. In the Tcl Console, enter:

report_cdc -no_waiver

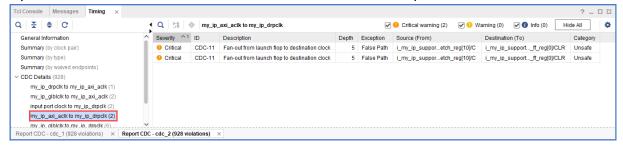
2. In the text report, verify that the table matches the original report from Report CDC before the CDC-10 waiver was created.

CDC Repor	t								
Severity	Source Clock	Destination Clock	CDC Type	Exceptions	Endpoints	Safe	Unsafe	Unknown	No ASYNC_REG
Critical Critical	my_ip_glblclk my_ip_axi_aclk	my_ip_axi_aclk my_ip_drpclk	No Common Primary Clock No Common Primary Clock		2	1	1	0	0
Critical Info	my_ip_axi_aclk my_ip_drpclk	my_ip_glblclk my_ip_axi_aclk	No Common Primary Clock No Common Primary Clock	False Path	942 1	12 1	351	579 0	185 0
Info Info	input port clock	my_ip_drpclk my_ip_drpclk	No Common Primary Clock No Common Primary Clock	False Path	2 6	2	Ŏ	Ŏ	Ŏ
Info	my_ip_drpclk	my_ip_glblclk	No Common Primary Clock		2	2	Ŏ	Ŏ	ò

TIP: You can also generate a list of all violations regardless of the waivers from the Vivado IDE. Select **Reports** \rightarrow **Timing** \rightarrow **Report CDC**. In the Report CDC dialog box, enable **Ignore all waivers**, and click **OK**.

Step 6: Waiving Multiple CDC Violations

The my_ip_axi_aclk to my_ip_drpclk CDC includes two Critical CDC-11 violations. This step covers how to waive both CDC-11 violations simultaneously.



1. To waive the violations, select the **CDC-11** rows in the CDC Report, right-click, and select **Create Waiver**.



Tcl Console Messages Timing ×								? _	0 6
Q ₹ ≑ C	Q 🔄 🔄 🗞 my_ip	_axi_aclk to my_ip_drpclk			🖌 🌗 Critical warning (2) 🛛 🖌	Warning (0)	🕑 🚯 Info (0) 🛛 Hi	de All	•
General Information	Severity ^1 ID	Description	Depth	Exc	ception Source (From)	Destination	(To)	Category	
Summary (by clock pair)	Critical CDC-11	Fan-out from launch flop to destination clock	ş	Fo	los Dath i mu in aunnar atab raal10		nnortff_reg[0]/CLR	Unsafe	
Summary (by type)	Oritical CDC-11	Fan-out from launch flop to destination clock	5		Path Properties	Ctrl+E	ortff_reg[0]/CLR	Unsafe	
Summary (by waived endpoints)			·		Elide Setting	,	•		
V CDC Details (928)				۰.	Highlight	,			
my_ip_drpclk to my_ip_axi_aclk (1)					Unhighlight				
my_ip_glblclk to my_ip_axi_aclk (2)				~					
input port clock to my_ip_drpclk (2)				\otimes	Mark	,			
my_ip_axi_aclk to my_ip_drpclk (2)					Unmark	Ctrl+Shift+N			
my_ip_glblclk to my_ip_drpclk (6)				Н	Schematic				
my_ip_axi_aclk to my_ip_glblclk (913)					View Path Report				
my_ip_drpclk to my_ip_glblclk (2)					Report Timing on Source to Destination.				
					Set Maximum Delay				
					Set Bus Skew	1			
				2)	Create Waiver				
· · · · · · · · · · · · · · · · · · ·				-	Export to Spreadsheet				
Report CDC - cdc_1 (928 violations) × Report CDC	C - cdc_2 (928 violations)	×							

2. In the Create Waiver dialog box, enter a description, and click **OK**.

🝌 Create W	aiver ×
Create waivers	for 2 cdc paths
User:	Xilinx
Description:	Safe fanout. Circuitry has been reviewed
Tags:	
Tcl Command	1 Preview
Q	
	r -type CDC -id CDC-11 -from [get_pins {i_my_ip_support_block/jesd204_i/inst/i_my_ip_reset_block/stre r -type CDC -id CDC-11 -from [get_pins {i_my_ip_support_block/jesd204_i/inst/i_my_ip_reset_block/stre
<	>
?	OK Cancel

In the Timing Report, the two selected rows are disabled when the waivers are created.

Note: One waiver is created for each selected row. In this example, two waivers are created.

Severity \land 1	ID	Description	Depth	Exception	Source (From)	Destination (To)	Category
Oritical	CDC-11	Fan-out from launch flop to destination clock	5	False Path	i_my_ip_suppotch_reg[10]/C	i_my_ip_supporff_reg[0]/CLR	Unsafe
Critical	CDC-11	Fan-out from launch flop to destination clock	5	False Path	i_my_ip_suppotch_reg[10]/C	i_my_ip_supporff_reg[0]/CLR	Unsafe

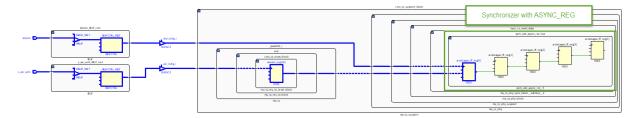
- 3. Select **Reports** → **Timing** → **Report CDC** to rerun Report CDC. In the Report CDC dialog box, make sure that *Report only waived paths* is unchecked, and click **OK**.
- 4. In the CDC Report, look at the my_ip_axi_aclk to my_ip_drpclk CDC.



The two Critical CDC-11 violations were replaced with two Info CDC-9 violations. Based on the CDC precedence rules, waiving CDC-11 unmasks CDC-9 for this circuit.

Q ≚ ≑ C	Q H (⊗_ my_ip	p_axi_aclk to my_ip_drpclk		V 🕛 🗘	Critical warning (0) 🛛 🕑 🕘 Warnin	g (0) 🕑 🚺 Info (2) 🛛 Hide Al	
General Information	Severity ^1	ID	Description	Depth	Exception	Source (From)	Destination (To)	Category
Summary (by clock pair)	Info	CDC-9	Asynchronous reset synchronized with ASYNC_REG property	5	False Path	i_my_ip_supporetch_reg[10]/C	i_my_ip_supportff_reg[0]/CLR	Safe
Summary (by type)	Info	CDC-9	Asynchronous reset synchronized with ASYNC_REG property	5	False Path	i_my_ip_supporetch_reg[10]/C	i_my_ip_supportff_reg[0]/CLR	Safe
Summary (by waived endpoints)								
CDC Details (928)								
my_ip_drpclk to my_ip_axi_aclk (1)								
my_ip_glblclk to my_ip_axi_aclk (2)								
input port clock to my_ip_drpclk (2)								
my_ip_axi_aclk to my_ip_drpclk (2)								
my_ip_axi_aclk to my_ip_drpclk (2) my_ip_glblclk to my_ip_drpclk (6)								

- 5. To view a schematic of the violation, select the CDC-9 row in the CDC Report, and click the Schematic toolbar button 3.
- 6. Verify that there is a 5-level synchronizer on the destination clock domain.



7. Compare the new Summary (by type) information with the information from the previous CDC Report.

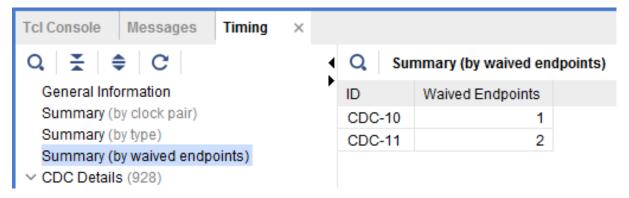
In the updated CDC Report, the two CDC-11 violations are no longer listed. Instead, there are two new CDC-9 violations.

Q ₹ ♦ C	Q Summa	ry (by type)		
General Information	Severity ^1	ID	Count	Description
Summary (by clock pair)	Critical	CDC-1	536	1-bit unknown CDC circuitry
Summary (by type)	Critical	CDC-4	4	Multi-bit unknown CDC circuitry
Summary (by waived endpoints)	Critical	CDC-10	186	Combinational logic detected before a synchronizer
CDC Details (928)	Critical	CDC-13	170	1-bit CDC path on a non-FD primitive
my_ip_drpclk to my_ip_axi_aclk (1)	Oritical	CDC-14	5	Multi-bit CDC path on a non-FD primitive
my_ip_glblclk to my_ip_axi_aclk (2)	👴 Warning	CDC-15	10	Clock enable controlled CDC structure detected
input port clock to my_ip_drpclk (2)	1 Info	CDC-3	10	1-bit synchronized with ASYNC_REG property
my_ip_axi_aclk to my_ip_drpclk (2)	Info	CDC-9	7	Asynchronous reset synchronized with ASYNC_REG proper
my_ip_glblclk to my_ip_drpclk (6)				
my_ip_axi_aclk to my_ip_glblclk (913)				
my_ip_drpclk to my_ip_glblclk (2)				

8. Look at the Summary (by waived endpoints) information.



In the updated CDC Report, there are three waived endpoints. This number is different from the number of waived violations (2), because CDC-11 is a multi-bit violation.



9. Generate different text reports and compare the results with previous reports.

For example, you can run the following Tcl commands:

report_cdc -details
report_cdc -details -waived
report_cdc -details -show_waiver
report_cdc -details -no_waiver

The following report was generated using the report_cdc -details -waived Tcl command and shows that three violations were waived.

CDC Report ID Severity Count Description	
CBC-10 Critical 1 Combinatorial logic detected before a synchronizer CBC-11 Critical 2 Fan-out from launch flop to destination clock	
ID Valued CDC-10 1 CDC-11 2	
Source Clock: mg_ip_glblclk Bestination Clock: mg_ip_axi_aclk CDC fage: No comen Primary Clock	
Row ID Severity Description Depth Exception Source (Fr	om) Destination (To)
1 CDC-10 Critical Combinatorial logic detected before a synchronizer 5 False Path i_my_ip_su	pport/s_32/got_sysref_r_reg/Ci_my_ip_support_block/jesd204_i/inst/sync_tx_sysref_captured/syncstages_ff_reg[0]/D
Source Clock; ng_ip_axi_aclk Bestination Clock; ng_ip_dpclk CDF ige: No comen Primary Clock	
Row ID Severity Description Depth Exception Source (From)	Destination (To)
1 CIC-11 Critical Farrout from launch flop to destination clock 5 False Path i_mg_ip_support 2 CIC-11 Critical Farrout from launch flop to destination clock 5 False Path i_mg_ip_support T	-block/ reg[10]/C i.ng.ip.support.block/i.jesd204.phg/inst/jesd204.phg.block.i/sgrc.rx.reset.data/sgr.cd.asgrc.rst.inst/arststages.ff.reg[0]/CR _block

Step 7: Exporting Waivers

In this step, you export waivers with the write_waivers Tcl command.

Note: The XDC output file can be imported using the read_xdc or source Tcl commands.

1. To export the CDC waivers, enter: write_waivers -type cdc waivers.xdc.

TIP: Alternatively, because there are no DRC or methodology waivers, you can enter:

write_waivers waivers.xdc Of write_xdc -type waiver waivers.xdc.

2. Open the waivers.xdc file to view the three waivers.



Note: The following example is reformatted to better show the different command line options.

```
create_waiver -type CDC -id {CDC-10} -user "Xilinx" \
  -desc "This is a safe CDC per review with the team"
  -from [get_pins i_my_ip_support_block/jesd204_i/inst/i_my_ip/i_tx/
i_tx_counters_32/got_sysref_r_reg/C] \
  -to [get_pins {i_my_ip_support_block/jesd204_i/inst/
sync_tx_sysref_captured/syncstages_ff_reg[0]/D}] \
  -timestamp "<timestamp>" ;#1
create_waiver -type CDC -id {CDC-11} -user "Xilinx" \
  -desc "Safe fanout. Circuitry has been released"
  -from [get_pins {i_my_ip_support_block/jesd204_i/inst/
i_my_ip_reset_block/stretch_reg[10]/C}] \
  -to [get_pins {i_my_ip_support_block/i_jesd204_phy/inst/
jesd204_phy_block_i/sync_rx_reset_data/xpm_cdc_async_rst_inst/
arststages_ff_reg[0]/CLR}] \
  -timestamp "<timestamp>" ;#1
create_waiver -type CDC -id {CDC-11} -user "Xilinx" \
  -desc "Safe fanout. Circuitry has been released" \setminus
  -from [get_pins {i_my_ip_support_block/jesd204_i/inst/
i_my_ip_reset_block/stretch_reg[10]/C}] \
  -to [get_pins {i_my_ip_support_block/i_jesd204_phy/inst/
jesd204_phy_block_i/sync_tx_reset_data/xpm_cdc_async_rst_inst/
arststages_ff_reg[0]/CLR}] \
 -timestamp "<timestamp>" ;#2
```

Step 8: Using the create_waiver Command

Waivers added from the Report CDC dialog box are created using the create_waiver command. You can view these commands as follows.

Note: You can use the create_waiver command line command for CDC, DRC, and methodology waivers. The options differ slightly depending on whether you are creating a CDC, DRC, or methodology waiver. For more information, including information on the different options, see the create_waiver command in the *Vivado Design Suite Tcl Command Reference Guide* (UG835).

- 1. Open the Vivado journal file (vivado.jou) to see the three distinct create_waiver commands issued by the Vivado IDE.
- 2. Scroll through the history of the Tcl Console to see the same three create_waiver commands.

```
TIP: The -from and -to options are used to specify the startpoints and endpoints. When a waiver is set from the Report CDC dialog box, both -from and -to are specified to match the exact violation. However, you can specify a CDC waiver using only the -from option or only the -to option, but more paths might be waived than expected.
```



Step 9: Waiving Multiple CDC Violations

In this step, you waive multiple CDC violations simultaneously.

1. In the CDC Report, view the my_ip_axi_aclk to my_ip_glblclk CDC under CDC Details.

This crossing has five CDC-14 violations, which are multi-bit violations. The five CDC-14 violations all start from the same two register clock pins:

```
i_my_ip_support_block/jesd204_i/inst/tx_cfg_test_modes_reg[2:1]/C
```

TIP: You can sort the table by the column ID to more easily see the five CDC-14 violations.

λ ≚ ≑ C	1	Q H &	my_ip_	axi_aclk to my_ip_glblclk		😪 🜖 Critical	warning (901) 🛛 🕑 😣 Warning (1	.0) 🕑 🚯 Info (2) 🛛 Hide	All
General Information	^	Severity	ID ^ 1	Description	Depth	Exception	Source (From)	Destination (To)	Category
Summary (by clock pair)		Warning	CDC-15	Clock enable controlled CDC structure detected	0	False Path	i_my_ip_supportmodes_reg[1]/C	i_my_ip_suppod_nls_r_reg/D	Safe
Summary (by type)		Oritical	CDC-14	Multi-bit CDC path on a non-FD primitive	0	False Path	i_my_ip_supportodes_reg[2:1]/C	i_my_ip_supp/TXDATA[2:1]	Unknown
Summary (by waived endpoints)	- 11	Oritical	CDC-14	Multi-bit CDC path on a non-FD primitive	0	False Path	i_my_ip_supportodes_reg[2:1]/C	i_my_ip_supp/TXDATA[2:1]	Unknown
CDC Details (928)	- 11	Critical	CDC-14	Multi-bit CDC path on a non-FD primitive	0	False Path	i_my_ip_supportodes_reg[2:1]/C		Unknown
my_ip_drpclk to my_ip_axi_aclk (1)	- 11	O Critical	CDC-14	Multi-bit CDC path on a non-FD primitive	0	False Path	i my ip supportodes reg[2:1]/C		Unknown
my_ip_glblclk to my_ip_axi_aclk (2)	- 11	Critical	CDC-14	Multi-bit CDC path on a non-FD primitive	0	False Path	i_my_ip_supportodes_reg[2:1]/C		Unknown
input port clock to my_ip_drpclk (2)	- 11	Oritical	CDC-13	1-bit CDC path on a non-FD primitive	0	False Path	i_my_ip_supportmodes_reg[2]/C	i_my_ip_suppoT/TXCTRL2[0]	Unsafe
my_ip_axi_aclk to my_ip_drpclk (2)	- 11	Oritical	CDC-13	1-bit CDC path on a non-FD primitive	0	False Path	i_my_ip_supportmodes_reg[2]/C	i_my_ip_suppoT/TXCTRL2[1]	Unsafe
my_ip_glblclk to my_ip_drpclk (6) my_ip_axi_aclk to my_ip_glblclk (913)	- 11	Critical	CDC-13	1-bit CDC path on a non-FD primitive	0	False Path	i_my_ip_supportmodes_reg[2]/C	i_my_ip_suppoT/TXCTRL2[3]	Unsafe
my ip drpclk to my ip glblclk (2)		Oritical	CDC-13	1-bit CDC path on a non-FD primitive	0	False Path	i my ip supportmodes reg[1]/C	i my ip suppST/TXDATA[0]	Unsafe

Because i_my_ip_support_block/jesd204_i/inst/

 $tx_cfg_test_modes_reg[*]/C$ matches five pins and you only need to target two of those five pins, construct the list of startpoints as follows:

```
set startpoints [list \
   [get_pins i_my_ip_support_block/jesd204_i/inst/
tx_cfg_test_modes_reg[1]/C] \
   [get_pins i_my_ip_support_block/jesd204_i/inst/
tx_cfg_test_modes_reg[2]/C] \
  ]
```

3. To waive the five CDC-14 violations, use the create_waiver Tcl command with the -from option:

```
create_waiver -type {CDC} -id {CDC-14} -user {Xilinx} -desc {No more CDC
14!} -from $startpoints
```

- 4. From the Vivado IDE, select **Reports** \rightarrow **Timing** \rightarrow **Report CDC** to rerun Report CDC.
- 5. In the CDC Report, verify that the CDC-14 violations are no longer reported in the Summary section.



. <u>≭</u>	1	Q Summar	ry (by type)		
General Information	^	Severity ^1	ID	Count	Description
Summary (by clock pair)		\rm Critical	CDC-1	536	1-bit unknown CDC circuitry
Summary (by type)		\rm Critical	CDC-4	4	Multi-bit unknown CDC circuitry
Summary (by waived endpoints)		Critical	CDC-10	186	Combinational logic detected before a synchronizer
CDC Details (923)		Critical	CDC-13	170	1-bit CDC path on a non-FD primitive
my_ip_drpclk to my_ip_axi_aclk (1)		👴 Warning	CDC-15	10	Clock enable controlled CDC structure detected
my_ip_glblclk to my_ip_axi_aclk (2)		1 Info	CDC-3	10	1-bit synchronized with ASYNC_REG property
input port clock to my_ip_drpclk (2)		 Info 	CDC-9	7	Asynchronous reset synchronized with ASYNC_REG property
my_ip_axi_aclk to my_ip_drpclk (2)	~				

6. To report only the waived violations, enter:

report_cdc -details -waived

The following figure shows the waived CDC violations in two different tables. The first table shows the 5 CDC-14 violations waived as multi-bit violations. The second table shows the 10 single-bit violations, calculated by multiplying the 5 multi-bit violations by 2 bits per multi-bit violation.

	Critical Critical	 1 2	Description Combinatorial logic detected b Ean-out from launch floe to de Multi-bit CDC path on a non-FI	estination_cloc										
ID 6 CDC-10 CDC-11 CDC-14	Waived 1 2 10													
Destinati CDC Type:	: No Commo	<: my_i non Pri	p_axi_aclk mary Clock											
Row ID	Sever	erity .	Description		1	Jepth E	Exception	Source (Fro	m)					
			Combinatorial logic detected bef	fore a synchron	izer	5 F	False Path	i_my_ip_sup	port_bloo	k/jesd204	_i/inst/i	_my_ip/i_t:	x/i_tx_c	ounters_32/gou
Source Cl Destinati	lock: my_i ion Clock: : No Commo	_ip_axi <: my_i mon Prim	_aclk	fore a synchron		5 F		i_my_ip_sup ce (From)	port_bloo	k/jesd204	_i/inst/i	_my_ip/i_t:	x/i_tx_c	ounters_32/gou
Source Cl Destinati CDC Type: Row ID 1 CDC-	Clock: my_i tion Clock: No Commo Sever 11 Criti	_ip_axi <: my_i mon Pri erity tical	_aclk p_drpclk mary Clock	 tination clock	Depth 5	Except False	tion Sour Path i_my.	ce (From) ip_support_			st/i_my_i	p_reset_bl		ounters_32/gou tch_reg[10]/C tch_reg[10]/L
Source Cl Destinati CDC Type: Row ID 1 CDC- 2 CDC- Source Cl Destinati	ilock: my_i ion Clock: No Commo -11 Criti -11 Criti ilock: my_i ion Clock:	_ip_axi <: my_i mon Pri erity tical tical tical _ip_axi. <: my_i	_aclk p_drpclk mary Clock Description Fan-out from launch flop to dest Fan-out from launch flop to dest _aclk	 tination clock	Depth 5	Except False	tion Sour Path i_my.	ce (From) ip_support_			st/i_my_i	p_reset_bl		 tch_reg[10]/P
Source Cl Destinati CDC Type: Row ID 1 CDC- 2 CDC- Source Cl Destinati	Clock: my_i ion Clock: : No Commo Sever 	_ip_axi. <: my_i mon Prin erity tical tical _ip_axi <: my_i mon Prin	_aclk p_drpclk mary [Jock Description Fan-out from launch flop to dest Fan-out from launch flop to dest _aclk _aclk	tination clock tination clock	Depth 5 5	Except False	tion Sour Path i_my, Path i_my,	ce (From) _ip_support_ _ip_support_			st/i_my_i	p_reset_bl		 tch_reg[10]/P

7. To export all the waivers inside a script and verify that a total of four waivers were added, enter:

write_waivers -type cdc waivers.xdc -force

Note: Because the waivers.xdc file already exists, the -force option must be specified to override the file.



or

TIP: Alternatively, because there are no DRC or methodology waivers, you can enter: write_waivers waivers.xdc -force

write_xdc -type waiver waivers.xdc -force

The list of waivers inside waivers.xdc appears as follows.

```
WRITE CDC Maivers

# URITE CDC Maivers

# cmd; write_waivers -type cdc -file waivers.xdc -force

current_instance -quiet

create_waiver -type CDC -id (CDC-11) -user "Xilinx" -desc "Safe fanout, Circuitry has been released" -from [get_pins i_mw_ip_support_block/jesd204_i/inst/i_my_ip_rest_block/stretch_reg[1]/C]

create_waiver -type CDC -id (CDC-11) -user "Xilinx" -desc "Safe fanout, Circuitry has been released" -from [get_pins i_mw_ip_support_block/jesd204_i/inst/i_my_ip_rest_block/stretch_reg[1]/C]

create_waiver -type CDC -id (CDC-14) -user "Xilinx" -desc "No more CDC-14!" -from [list [get_pins {i_my_ip_support_block/jesd204_i/inst/tx_cfg_test_modes_reg[1]/C] [ge
```

8. To import the waivers.xdc file, enter:

read_xdc waivers.xdc

The following warnings show that duplicate waivers were not added to the existing waivers. Only waivers that are exact duplicates of existing waivers are rejected.

```
WARNING: [Vivado_Tcl 4-935] Waiver ID 'CDC-10' is a duplicate and will
not be added again.
WARNING: [Vivado_Tcl 4-935] Waiver ID 'CDC-11' is a duplicate and will
not be added again.
WARNING: [Vivado_Tcl 4-935] Waiver ID 'CDC-11' is a duplicate and will
not be added again.
WARNING: [Vivado_Tcl 4-935] Waiver ID 'CDC-14' is a duplicate and will
not be added again.
```

Step 10: Waiving Multiple DRC Violations

In this step, you waive multiple DRC violations simultaneously.

- 1. Select **Reports** \rightarrow **Report DRC**.
- 2. In the Report DRC dialog box, leave all settings at their default, and click OK.



🍌 Report DRC			\times
Check design against sele	cted rule decks and/or individual desig	ın rules.	4
<u>R</u> esults name: <u>Interactive report file:</u> <u>Export to file:</u>	drc_1		
Waivers Apply waivers Display only w Ignore all waivers Rule Decks		Bulae (2440 of 2425)	
Vivado Rule Decks	s ecks checks checks	Rules (2119 of 2135) Q X All Rules (2135) > Netlist (737) > Pin Planning (110) > Clocking (4) > Memory (148) > Floorplan (10) > Implementation (239) > Physical Configuration (842)	
 ✓ Open in a new tab ? 			Cancel

3. In the DRC Report, right-click **UCIO#1**, and select **Create Waiver** to create a waiver for the UCIO-1 violations.

Note: The UCIO#1 violation combines 125 individual violations into a single violation. Similarly, the NSTD#1 violation covers 113 ports.



Tcl Console Messages DR	C × Timing			
Q ≚ ♦ 📲 🗎	🖌 🖖 3 Critical Warnings 🛛 🖓 🕛 1 Warning 🛛 Hide All			
Name	Details			
V () NSTD-1 (1)				
0 NSTD #1	113 out of 125 logical ports use I/O standard (IOSTANDARD) value 'DEFAULT', instead of a user as components to which it is connected. To correct this violation, specify all I/O standards. This design command: set_property SEVERITY {Warning} [get_drc_checks NSTD-1]. NOTE: When using the V s axi araddr[10], s axi araddr[9], s axi araddr[8], s axi araddr[7], s axi araddr[6], s axi araddr	n will f ivado	ail to generate a bitstream unles Runs infrastructure (e.g. launch	s all logic _runs Tcl (
UCIO-1 (1)				
	125 out of 125 logical ports have no user assigned specific location constraint (LOC). This may car	use I/(O contention or incompatibility w	ith the boa
0 UCIO #1	correct this violation, specify all pin locations. This design will fail to generate a bitstream unless a [get_drc_checks UCIO-1]. NOTE: When using the Vivado Runs infrastructure (e.g. launch_runs To	•=	Сору	d
	correct this violation, specify all pin locations. This design will fail to generate a bitstream unless a	•=		d
 UCIO #1 Implementation (1) 	correct this violation, specify all pin locations. This design will fail to generate a bitstream unless a [get_drc_checks UCIO-1]. NOTE: When using the Vivado Runs infrastructure (e.g. launch_runs To	•=		Ctrl+E
	correct this violation, specify all pin locations. This design will fail to generate a bitstream unless a [get_drc_checks UCIO-1]. NOTE: When using the Vivado Runs infrastructure (e.g. launch_runs To	•=	Violation Properties	d
 Implementation (1) 	correct this violation, specify all pin locations. This design will fail to generate a bitstream unless a [get_drc_checks UCIO-1]. NOTE: When using the Vivado Runs infrastructure (e.g. launch_runs To	•=	Violation Properties Auto-Select Objects Auto-Mark Objects	d
 Implementation (1) Routing (1) 	correct this violation, specify all pin locations. This design will fail to generate a bitstream unless a [get_drc_checks UCIO-1]. NOTE: When using the Vivado Runs infrastructure (e.g. launch_runs To	•=	Violation Properties Auto-Select Objects Auto-Mark Objects Wrap Lines	d
 Implementation (1) Implementation (1) Routing (1) Chip Level (1) 	correct this violation, specify all pin locations. This design will fail to generate a bitstream unless a [get_drc_checks UCIO-1]. NOTE: When using the Vivado Runs infrastructure (e.g. launch_runs To	✓ ✓	Violation Properties Auto-Select Objects Auto-Mark Objects	d

4. In the Create Waiver dialog box, look at the output in Tcl Command Preview, and click OK.

🙏 Create Wa	aiver	\times
Create waiver f	or 1 violation	
User:	Xilinx	9
Description:	Waive UCIO DRC violations	9
Tags:		
Tcl Command	1 Preview	
Q		
create_waive	r -of_objects [get_drc_violations -name drc_1 {UCIO-1#1}] -user Xilinx -description {Waive UCIO DRC vio	D
<	· · · · · · · · · · · · · · · · · · ·	
?	OK Cancel	
<u> </u>		

5. To generate the drc_waivers.xdc file and verify that the waiver is waiving all 125 objects, enter:

write_waivers -type DRC drc_waivers.xdc

6. In the XDC file, look at the expanded port list, and notice that some of the strings from the violations message were converted to wildcards (*).



Strings are automatically converted to wildcards for UCIO-1, NSTD-1, TIMING-15, and TIMING-16 type violations. For UCIO-1, the numbers of objects in the violations are replaced with wildcards, because the numbers of elements are not meaningful.

* WRITE DRC WAIVERS
cmd: write_waivers -type DRC drc_waivers.xdc
current_instance -quiet
create_waiver -type DRC -id {UCIO-1} -user "Xilinx" -desc "Waive UCIO DRC violations" -objects [get_ports { refclkOp glblclkp refclkOn tx_start_of_frame[3] tx_start_of_multiframe[3] glblclkn
tx_reset drpclk tx_start_of_multiframe[2] txp[3] tx_start_of_multiframe[0] tx_start_of_frame[2] tx_start_of_frame[1] s_axi_rready tx_sync tx_sysref tx_aresetn s_axi_rdata[1] s_axi_rvalid
s_axi_rresp[0] s_axi_rresp[1] s_axi_rdata[0] s_axi_rdata[2] s_axi_rdata[3] s_axi_rdata[4] s_axi_rdata[9] s_axi_rdata[5] s_axi_rdata[10] s_axi_rdata[6] s_axi_rdata[7] s_axi_rdata[8]
s_axi_rdata[15] s_axi_rdata[11] s_axi_rdata[12] s_axi_rdata[13] s_axi_rdata[16] s_axi_rdata[17] s_axi_rdata[18] s_axi_rdata[14] s_axi_rdata[20] s_axi_rdata[21] s_axi_rdata[22] s_axi_rdata[27]
s_axi_rdata[23] s_axi_rdata[19] s_axi_rdata[25] s_axi_rdata[26] s_axi_arready s_axi_rdata[28] s_axi_rdata[24] s_axi_rdata[30] s_axi_rdata[31] s_axi_araddr[2] s_axi_arvalid s_axi_rdata[29]
s_axi_araddr[7] s_axi_araddr[3] s_axi_araddr[4] s_axi_araddr[5] s_axi_araddr[6] s_axi_araddr[9] s_axi_araddr[10] s_axi_bready s_axi_wstrb[0] s_axi_wstrb[1] s_axi_araddr[8]
s_axi_bresp[1] s_axi_wready s_axi_wvalid s_axi_wdata[0] s_axi_bresp[0] s_axi_wstrb[2] s_axi_araddr[11] s_axi_wdata[4] s_axi_wdata[1] s_axi_wstrb[3] s_axi_wdata[2] s_axi_wdata[3]
s_axi_wdata[9]s_axi_wdata[5]s_axi_wdata[6]s_axi_wdata[7]s_axi_wdata[8]s_axi_wdata[14]s_axi_wdata[10]s_axi_wdata[11]s_axi_wdata[12]s_axi_wdata[13]s_axi_wdata[19]s_axi_wdata[15]
s_axi_wdata[16] s_axi_wdata[17] s_axi_wdata[18] s_axi_wdata[24] s_axi_wdata[20] s_axi_wdata[28] s_axi_wdata[25] s_axi_wdata[26] s_axi_wdata[27] s_axi_awready s_axi_awvalid txp[4]
tx_start_of_multiframe[1] txp[1] tx_start_of_frame[0] txp[2] txn[1] txn[3] txn[2] s_axi_awaddr[11] txn[0] txp[0] e_axi_aclk_e_axi_areasta_e_axi_awaddr[7] s_axi_awaddr[10] s_axi_awaddr[8]
s_axi_awaddr[9] s_axi_awaddr[2] txn[4] s_axi_awaddr[5] s_axi_awaddr[6] s_axi_awaddr[3] s_axi_awaddr[4] }] -strings { "*" } -strings { "*" } -timestamp "Wed Mar 14 22:57:14 GMT 2018" ;#1

7. To delete the DRC waiver and rewrite the waiver using wildcards to target a subset of the ports objects, enter:

```
delete_waivers [get_waivers -type drc]
create_waiver -type DRC -id {UCIO-1} -user "Xilinx" -desc "Waive
selected UCIO violations" -objects [get_ports { s_axi_rdata[*]
s_axi_wdata[*] s_axi_araddr[*] } ] -strings { "*" } -strings { "*" }
```

Note: This command only covers a subset of the original 125 objects.

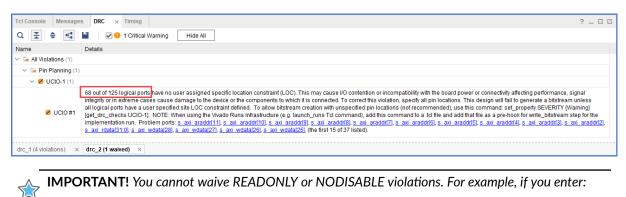
- 8. Select **Reports** \rightarrow **Report DRC** to rerun Report DRC.
- 9. In the Report DRC dialog box, select **Display only waived violations** to report only waived violations, and click **OK**.





🝌 Report DRC		×
Check design against sele	cted rule decks and/or individual design rules.	4
Results name:	drc_2 Overwrite Append	
Waivers Apply waivers Apply waivers Jisplay only w Ignore all waivers Rule Decks		
 ✓ ■ Vivado Rule Deck ✓ Ø default Ø opt_check Ø placer_che Ø router_che Ø bitstream_ Ø incr_eco_e Ø eco_check 	s All Rules (2136 > All Rules (2136 > Netlist (737 > Implement checks > Implement	7) ng (110) +) 48) 11)
 ✓ Open in a new tab ? 		OK Cancel

In the DRC Report, verify that only 68 violations are waived out of 125.



create_waiver -type DRC -id RTSTAT-1 -description "Waive RTSTAT-1"



The Vivado tools issue the following error:

```
ERROR: [Vivado_Tcl 4-934] Waiver ID 'RTSTAT-1' is READONLY or
NODISABLE and cannot be waived.
These Factory designations specify that a check is required and may
not be overridden by user action.
```

Step 11: Generating a Summary Report for Waived Violations

This step covers how to use the report_waivers Tcl command to generate a summary report for CDC, DRC, and methodology waivers.

MPORTANT! Before running the *report_waivers* command, you must rerun Report CDC, Report DRC, or Report Methodology to ensure that added or removed waivers are included in the statistics reported by *report_waivers*.

1. To rerun Report CDC, enter:

report_cdc

2. To rerun Report DRC, enter:

report_drc

Note: You do not need to rerun Report Methodology, because no methodology waivers were set.

3. To create a summary report, enter:

report_waivers

By default, report_waivers reports only waived violations. The following figure shows the UCIO-1, CDC-10, CDC-11, and CDC-14 rules, which have defined waivers.



Table Of Contents 1. REPORT SUMMARY 2. REPORT DETAILS (DRC) 3. REPORT DETAILS (METHODOLOGY: no waivers) 4. REPORT DETAILS (CDC)									
1. REPORT SUMMARY									
Waiver Type Total Vios Remaining Vios Waived Vios Used Waivers Set Waivers DRC 240 172 68 1 1 METHODOGY 0 0 0 0 0 CDC 957 944 13 4 4 Note: This report is based on the most recent report_drc/report_methodology/report_cdc runs.									
2. REPORT DETAILS (DRC)									
Rule Severity Description Total Vio	s Remaining Vios	Waived Vios	Used Waivers	Set Waivers					
UCIO-1* Critical Warning Unconstrained Logical Port 125	57	68	1	1					
4. REPORT DETAILS (CDC)									
Rule Severity Description	Total Vios	Remaining Vios	Waived Vios	Used Waivers	Set Waivers				
CDC-10 Critical Combinational logic detected before a synchron CDC-11 Critical Fan-out from launch flop to destination clock CDC-14 Critical Multi-bit CDC path on a non-FD primitive	izer 187 2 10	186 0 0	1 2 10	1 2 1	1 2 1				
Note: Any 'Rule' which is flagged by '*' is an aggregating message and its counts are based on the number of objects represented, rather than the number of messages.									

Note the number of waived objects and total violations:

- The aggregating DRCs are reported as 1 violation per object inside the violation. Because there are 113 objects in NSTD-1, 125 objects in UCIO-1, 1 in RTDAT-13, and 1 in AVAL-326, a total number of 240 DRC violations are reported in the Summary table.
- The Report Summary table reports all of the violations.
- The Report Details tables only report the check IDs that have one or more waivers.
- 4. To generate detailed tables with all of the rules, including rules with no waivers, enter:

report_waivers -show_msgs_with_no_waivers

The following figure shows the report with all DRC and CDC rules reported in the Report Details.



REPORT E	ontents SUMMARY DETAILS (DRC)	DOLOGY: no waive	ers)								
1. REPORT S	SUMMARY										
		-	Waived Vios Us			ivers					
DRC METHODOLOGY CDC	240 7 0 957	172 0 944	68 1 0 0 13 4 recent report_dro		1 0 4	y/report_c	de runs.				
	DETAILS (DRC)	Description		Total Vice	Pomai	ning Wice	Waiwed Vice	Ucod	Waiwars	Sat Waiward	
		Description		10081 0108							
AVAL-326 NSTD-1* RTSTAT-13	Critical Warn: Critical Warn: Critical Warn: Critical Warn:	ing Unconstrain ing Hard_block_ ing Unspecified ing Insufficien	ed Logical Port must_have_LOC I I/O Standard t Routing	125 1 113 1	1 113 1		0 0 0 0	0 0 0		1 0 0 0	
	DETAILS (CDC)										
Rule Sev	verity Descrip	otion								ios Used Waivers	Set Waivers
CDC-10 Critical Combinational logic detected before a synchronizer 187 186 1 1 CDC-11 Critical Fan-out from launch flop to destination clock 2 0 2 2 2 CDC-14 Critical Multi-bit CDC path on a non-FD primitive 10 0 10 1 1 CDC-14 Critical Multi-bit CDC clicultry 536 536 0 0 0 CDC-3 Info 1-bit unknown CDC circuitry 9 9 0 0 0 CDC-4 Critical Multi-bit unknown CDC circuitry 28 28 0 0 0 CDC-4 Critical Multi-bit unknown CDC circuitry 28 28 0 0 0 CDC-4 Critical Multi-bit unknown CDC circuitry 170 170 0 0 0 CDC-4 Critical Multi-bit unknown CDC circuitry 170 170 0 0 0 CDC-3 Info Asy											
Note: Any '	'Rule' which is		is an aggregati							jects represented,	

5. To run Report Methodology, enter:

report_methodology

6. To generate detailed tables with all of the rules, including rules with no waivers, enter:

report_waivers -show_msgs_with_no_waivers

The exact statistics are reported, as shown in the following figure.

Note: This figure does not include the Report Details (CDC) section.

1. REPORT S	UMMARY											
Waiver Type	e Total Vios Re	maining Vios Waived Vios	Used Wai	vers	Set Wai	vers						
DRC METHODOLOGY CDC Note: This	957 94	9 0 1 13	1 0 4 drc/report	t meth	1 0 4 nodology	/report c	dc runs					
Note: This report is based on the most recent report_drc/report_methodology/report_ddc runs.												
		Description	Total	Vios	Remain	ing Vios	Waived	Vios	Used W	Jaivers	Set Wa	ivers
AVAL-326 NSTD-1*	Critical Warning Critical Warning	Unconstrained Logical Po Hard_block_must_have_LO Unspecified I/O Standard Insufficient Routing	2 1		57 1 113 1		68 0 0 0 0		1 0 0 0		1 0 0 0	
3. REPORT DETAILS (METHODOLOGY)												
Rule	Severity	Description		Tota	al Vios	Remainin	g Vios	Waive	d Vios	Used W	aivers	Set Waivers
AVAL-324 LUTAR-1* TIMING-9 TIMING-10 TIMING-18*	Critical Warning Warning Warning Warning Warning Warning	LUT drives async reset		1		1 40 1 1 115		0 0 0 0 0 0		0 0 0 0 0		0 0 0 0 0 0



Step 12: Using Waiver Commands

In this step, you run additional commands related to the waivers.

1. To return a collection of CDC waiver objects, enter:

```
get_waivers -type cdc
```

The following CDC waivers are returned:

```
CDC-10#1 CDC-11#1 CDC-11#2 CDC-14#1
```

2. To filter the list of waivers to only return CDC-14 waivers, enter:

```
get_waivers -filter {ID == CDC-14}
CDC-14#1
```

3. To report all of the properties on a CDC waiver object, enter:

report_property [lindex [get_waivers -type cdc] end]

The following properties are returned:

Property CLASS DESCRIPTION	Type string string	Read-only true false	Value cdc_waiver No more CDC-14!
ID	string	true	CDC-14
INDEX	string	true	1
IS_SCOPED	bool	true	0
NAME	string	true	CDC-14#1
OBJECT_COUNTS	string	true	pins:2
SCOPE	string	true	
TAGS	string	false	
TIME	string	true	<timestamp></timestamp>
TYPE	string	true	CDC
USED_CNT	string	true	10
USER	string	true	Xilinx

Note: You cannot retrieve the design objects attached to a waiver object.

4. To delete all of the previously created CDC-14 waivers, enter:

delete_waivers [get_waivers -filter {ID == CDC-14}]

Note: After a waiver object is deleted, the waiver no longer applies and the violations that it waived are reported again.

5. To delete all of the remaining CDC waivers, enter:

delete_waivers [get_waivers -type cdc]





Summary

In this lab, you accomplished the following:

- Waived CDC and DRC violations
- Generated reports for waived violations
- Exported waivers
- Used waiver commands





Lab 2

Increasing Design Performance Using Report QoR Suggestions

Introduction

The report_qor_suggestions (RQS) command enables the Vivado[®] Design Suite tools to analyze a design and provide automated solutions for enhancing QoR. Built into this command when using the Vivado IDE is the report_qor_assessment (RQA) command. The command can be run on an open design after synthesis or after any stage in the implementation flow.

The RQA command assesses the likelihood that a design will meet its timing requirements and provides a quick summary of design metrics. The RQS command evaluates the design in five key areas and suggests fixes or improvements in these areas. The five areas are utilization, clocking, constraints, congestion, and timing. Recommendations from RQS can take the following forms:

- RQS objects. These can add:
 - Switches to a given command.
 - Properties to a given design object.
 - Implementation strategies customized for the design using machine learning algorithms.
- Text recommendations that require user intervention.

This lab covers how to:

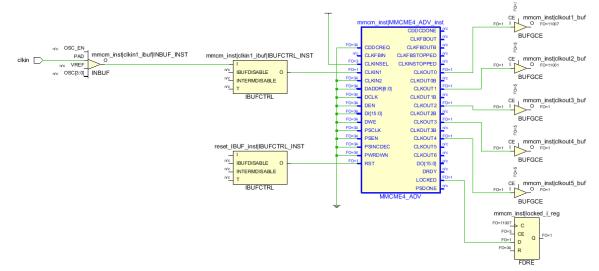
- Analyze the QoR assessment report.
- Analyze the QoR suggestion report.
- Export suggestions to an RQS file.
- Add an RQS file to synthesis and implementation runs.
- Accumulate suggestions in an RQS file by generating suggestions at different implementation stages or on different runs.



Step 1: Understanding the Design

This lab uses an architected design to demonstrate some of the features of RQS. Suggestions are triggered by the design of the RTL and the placement of blocks using floorplanning. The design contains the following modules:

• Clocking Module: The main clocking circuit for the design resides in clocking_module.vhd. For simplicity, RST is tied to GND. LOCKED is registered and tied to an output port. The structure of this block is shown in the following figure.



- **Reg CLKA to CLKB Module:** This module contains a synchronous CDC for a large bus. It registers input data using CLKA and then passes it to a register on the CLKB domain to be passed to the output. Registering large buses on different related clock domains can impact hold slack (WHS/THS) and setup slack (WNS/TNS).
- **Bit Expander and Bit Reducer Modules:** These modules enable the expansion and contraction of internal data widths so that the design does not run out of I/Os. The modules take an arbitrary data width and expand or contract it to or from a desired size. The contraction logic creates many logic levels.

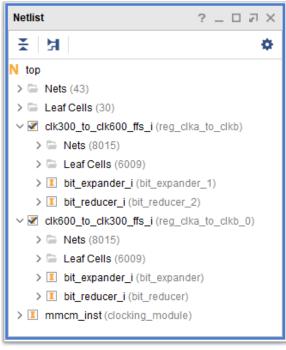
The following steps cover opening the project and examining the placement of the floor-planned modules.

In the Vivado Design Suite, go to File → Project → Open and select the project located in <extract_Dir>/Lab2/project_2.



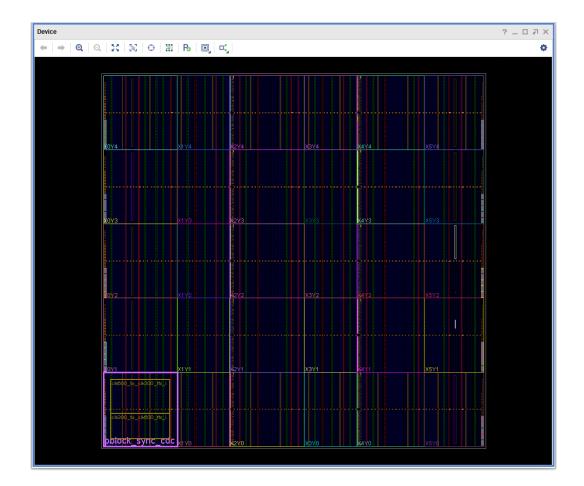
<u>F</u> ile	Flow	Tools	<u>W</u> indo	w	<u>H</u> elp	Q- Quic	k Access
	Project		•		<u>N</u> ew		
	<u>C</u> onstrai	ints	÷		Open		
	Simul <u>a</u> ti	on Waveforr	m→		Open F	_	•
	Chec <u>k</u> pc	pint	+		OpenE	Ex <u>a</u> mple	-
	<u>I</u> P		Þ				
	I <u>m</u> port		÷	art			
	Exit						
-	Create						

- 2. In the Flow Navigator, click Run Synthesis and wait for synthesis to complete.
- 3. In the Flow Navigator, click **Open Synthesized Design**.
- 4. In the Netlist view, look at the hierarchy.



5. In Device view, look at the pblock. This has been added to control placement of the reg_clka_to_clkb modules and force a poor clock skew.





Step 2: Running Report QoR Suggestions

This step covers running the <code>report_qor_suggestions</code> command to generate a report. The command can be run on an open design at any stage of the implementation flow after synthesis. In project mode, this is typically after synthesis or implementation. In non-project mode, this can be after <code>synth_design,link_design,opt_design,place_design,phys_opt_design, or route_design.</code>

1. In the Vivado IDE, from the pull down menus, click **Reports** → **Report QoR Suggestions...** to bring up the dialog box shown in the following figure.



🝌 Report QoR Suggestic	ons		×
Report design and tool results (QoR)	option changes to imp	rove the quality of	4
Number of paths for	suggestion analysis:		100 🌲
Export to file:			
		ОК	Cancel

2. Click **OK** to run the command. The report opens automatically in the integrated design environment (IDE). Due to the interactive nature of the report, only one instance of the report can be open at any time.

Built into the running of this command is the report_qor_assessment command when run from within the Vivado IDE. This command uses the design metrics and the same timing paths to make an assessment of how likely the design is to meet timing. In Tcl, this command is *not* called automatically, so must be called separately. The equivalent Tcl commands are as follows:

report_qor_assessment -max_paths 100 -file rqa.rpt
report_qor_suggestions -max_paths 100 -file rqs.rpt

The command will:

- Generate an assessment report.
- Examine the design and generate new suggestions.
- Generate a report on the suggestions.

Note: By default, the RQS command reports on the 100 worst failing paths per clock group. You can change the number of paths that RQS uses for the analysis of timing-critical paths by modifying the - max_paths switch. Increasing this number generates more suggestions, but on paths that are reducing in criticality.

Step 3: Understanding the Report

This step explains the different sections of the generated QoR Suggestions report. On the left of the report window, you can navigate to the different sections of the report; on the right, more information is provided.



 The report is broken down into two main sections, the QoR Assessment and the Suggestion Report. All items below the Suggestion Report are also related to suggestions. First click on the RQA Summary under the QoR Assessment section of the report. This shows the QoR Assessment Score and the Flow Guidance.

QoR Suggestions		_ D 7 ×
Q ≚ ≑ C	Q RQA Summary	
General	QoR Assessment Score Flow Guidar	nce
v QoR Assessment	3 - Design runs have a small chance of success Run report	_qor_suggestions to generate suggestions
RQA Summary		
Assessment Details		
ML Strategy Availability		
 Suggestion Report 		
GENERATED		
✓ Timing		
RQS_TIMING-33-1		
RQS_TIMING-44-1	× <	>
	Reset Default Export Suggestions Add Suggestions to P	roject

The QoR Assessment score is 3. This means that there is some chance of meeting timing but it is unlikely. The flow guidance is recommending to see if QoR suggestions can improve the design.

Note: The report_qor_assessment command runs using the latest data available to it. For example, clock skew is only accurate after logic is placed so less emphasis is given to the clock skew score at the pre-place stage in the implementation flow.

2. Click on the **Assessment Details** section of the report.

QoR Suggestions						_	
Q ₹ ♦ C	1	Q Assessment Details					
General	^	Name	Threshold	Actual	Used	Available	Status
✓ QoR Assessment		Utilization					ок
RQA Summary		Clocking					ок
Assessment Details		Congestion					ок
ML Strategy Availability		Timing					
 Suggestion Report 		WNS	-0.100	-0.396	- 2	- 22	REVIEW
GENERATED		TNS	-0.100	-0.396			REVIEW
 Timing RQS_TIMING-33-1 		Number of paths above Max Net/LUT Budgeting	0	2	529.	-	REVIEW
RQS_TIMING-44-1	~	<					
	F	eset Default Export Suggestions Add Sugg	estions to Pr	oject			

The detailed table shows the categories that have passed the assessment and individual failed metrics. The score is an aggregate of the failed metrics.

The metrics failing are timing based, and all the others are passing. Items marked as OK can be ignored, but items marked REVIEW should be compared against the threshold to see how severe the failure is.

Note: For paths failing Net and LUT budgeting, refer to the QoR suggestion RQS_XDC-1 for more details on the timing paths.

3. In the Suggestion Report, select **GENERATED**. This brings up the report section shown in the following figure.

Q 😤 🌲 C 📢	Q 🔮 🌲 🚺 GE	NER/	ATED				
General	ID		GENERATED_AT	APPLICABLE_FOR	AUTOMATIC	SCOPE	Incremental Frie
✓ QoR Assessment	∼ 🚍 Timing	1					
RQA Summary	RQS_TIMING-33-1	1	synth_design	synth_design	Yes	GLOBALSCOPE	No
Assessment Detail	RQS_TIMING-44-1	1	synth_design	synth_design	Yes	GLOBALSCOPE	No
ML Strategy Availal	RQS_TIMING-19-1		synth_design	synth_design	Yes	GLOBALSCOPE	No
 Suggestion Report 	RQS_TIMING-27-1		synth_design	synth_design	Yes	GLOBALSCOPE	No
GENERATED	✓ □ XDC						
✓ Timing	RQS_XDC-1-1		synth_design	synth_design	No	GLOBALSCOPE	No
RQS_TIMING-33-1	V 🖴 Clocking						
RQS_TIMING-44-1	RQS_CLOCK-9-1		synth_design	place_design	Yes	GLOBALSCOPE	Yes
RQS_TIMING-19-1 ~ XDC	RQS_CLOCK-21-1		synth_design	place_design	No	GLOBALSCOPE	No
RQS_XDC-1-1							
 Clocking RQS_CLOCK-9-1 RQS_CLOCK-21-1 							
	K MI Strategies are availab	lo on	u in default/ovnlor	a at successfully re	uted design		
	ML Strategies are availab	_				7	
	Reset Default	Ex	port Suggestions	Add Suggesti	ons to Project		

The GENERATED section provides a list of all the suggestions that have been generated at this stage of the current run. Each suggestion has a description that details the reason for the suggestion. Additionally, for each suggestion the following information is provided.

Item	Description	Comment
GENERATED_AT	This shows what stage of the design the suggestion was generated at. Typical values are place_design or route_design.	As you progress through the design stages, the decisions that the tool makes are based on the information available at the time. Additionally, information accuracy increases after placement and again after routing.
APPLICABLE_FOR	This stage must be rerun for the suggestion to take effect.	Most suggestions are executed at either <pre>synth_design</pre> or <pre>place_design.</pre>
AUTOMATIC	Indicates if the suggestion is executed automatically or if manual intervention is required.	Automatic suggestions either recommend a switch to the tool or a property to be added to a cell or net.
INCREMENTAL FRIENDLY	Indicates if the suggestion is optimized for the incremental flow.	Non-incremental friendly suggestions must be already present in the reference checkpoint. If you want to add non-incremental friendly suggestions, an updated reference checkpoint must be used.
SCOPE	Indicates the target synthesis run level. GLOBALSCOPE is top level. Otherwise, a sub-module is targeted	Allows a single RQS file to be used on top-level and sub- module out of context (OOC) synthesis runs. Only applicable to synthesis suggestions.

Under the other sections of the report there are usually details about the individual suggestions that have been generated.

4. Click on the **RQS_XDC_1_1** hyperlink. This will take you to the details section for this suggestion.



QoR Suggestions										_ 🗆 🗟 🗙
Q ≚ ♦ "•	Q RQS_XI	DC-1-1								
RQS TIMING-4	No of Paths	Logic Levels	Routes	Slack	Req.	Skew	Datapath Delay	Cell%	Route%	Source Clock
RQS TIMING-1	1	5	6	-0.396	1.667	-0.145	1.902	56.50	43.50	clk_600_clk_wiz
Clocking RQS_CLOCK-9- RQS_CLOCK-21										
	<				1.000	0)

The suggestion description says that the timing constraint is too tight for the given path(s).

The path has a large negative slack which would stand out in a timing report. Timing paths use net delays that are optimal, this gives the tools the correct order to place and route them. Close analysis shows this is a 600 MHz path with high logic levels. This is a path that will need to be fixed.

- 5. Click on the back arrow button * to go back to the GENERATED view.
- 6. In the GENERATED view, click on the **RQS_TIMING-33_1** row. You can see this is an AUTOMATIC suggestion that is applicable for <code>synth_design</code> (see the APPLICABLE_FOR column). This indicates that you must rerun the <code>synth_design</code> command to make use of this suggestion.

QoR Suggestions						_ D 7 X
Q ¥ \$ C	Q 🛨 🌲 🚺 GE	NERA	ATED			
Suggestion Report ^	ID		GENERATED_AT	APPLICABL	A	DESCRIPTION
GENERATED	∼ 🗁 Timing					
~ Timing	RQS_TIMING-33-1		synth_design	synth_desig	Yes	Improve timing on critical path using RETIMING_FORWAR
RQS_TIMING-33-1	RQS_TIMING-44-1		synth_design	synth_desig	Yes	Improve timing on critical path using RETIMING_BACKWAI
RQS_TIMING-44-1	RQS_TIMING-19-1		synth_design	synth_desig	Yes	Retime across high fanout nets to improve timing.
RQS_TIMING-19-1	RQS_TIMING-27-1		synth_design	synth_desig	Yes	Improve module level timing by using BLOCK_SYNTH.RETI
~ XDC	∽ the xdd of the x			-		
RQS_XDC-1-1	RQS_XDC-1-1		synth_design	synth_desig	No	Paths above Max Net/LUT budgeting. Review paths and
 Clocking 				-		
RQS_CLOCK-9-1	RQS CLOCK-9-1		synth_design	place_desig	Yes	Sub optimal Fvco on MMCM/PLL. Update MMCM/PLL setting
RQS_CLOCK-21-1	<					>
	ML Strategies are availab	le oni	ly in default/explor	e at successfu	lly rout	ted design.

When you select the row in the RQS Summary view, the suggestion object is selected and the properties can be seen in the QoR Suggestion Properties window. If you examine the command property, you can confirm that generates a retiming forward property for synth_design.



QoR Sugg	estion Prop	erties ? _ D 2 X
RQS_TIM	ING-33-1	$\leftarrow \Rightarrow \diamond$
Q I	\$ - 4	+ - 0 2+
APPLICA	BLE_FOR	synth_design
APPLIED		
AUTO		~
CATEGO	RY	Timing
CLASS		qor_suggestion
COMMAN	ND	set_property retiming_forward 1 [get_cells {{clk300_to_clk600_ffs_i/
DESCRIP	NOIT	Improve timing on critical path using RETIMING_FORWARD property.
ENABLED	D	
FAILED_1	TO_APPLY	
FLOW_SU	UPPORT	Default
General	Propertie	IS

7. In the GENERATED view, click on the **RQS_TIMING-33_1** ID to go to the details table for that suggestion. Careful examination of the Endpoint column will confirm that this is the same path that was mentioned for RQS_XDC-1.

QoR Suggestions 📃 🗆 🗟 🗙								
Q ≚ ≑ ← → `	* Q	RQS_TIMING-33	-1					
General	^ nout	Datapath Delay	Cell%	Route%	Source Clock	Destination Clock	Startpoint	Endpoint
 RQS Summary GENERATED 		1.902	56.50	43.50	clk_600_clk_wiz_0	clk_600_clk_wiz_0	clk300_to_clk600_ffs_i/expanded_sig3_reg[1932]/C	clk300_to_clk600_ffs_i/bit_reducer_i/tmp_r_reg/D
V Timing								
RQS_TIMING-33-1								
RQS_TIMING-44-1								
~ XDC								
RQS_XDC-1-1								
Clocking								
RQS_CLOCK-9-1	•							
	<			[Reset Default	Export Suggestions	Add Suggestions to Project	

8. In the GENERATED view, you can see the remaining suggestions. The RQS_CLOCK-9 suggestion is applicable for place_design. This is shown in the following figure:

QoR Suggestions									
General	ID		GENERATED_AT	APPLICABLE_FOR	AUTOMATIC	Incre	DESCRIPTION		
V RQS Summary	∼ 🗁 Timing								
GENERATED	RQS_TIMING-33-1	v	synth_design	synth_design	Yes	No	Improve timing on critical path using RETIMING_FORWARD property.		
✓ Timing	RQS_TIMING-44-1		synth_design	synth_design	Yes	No	Improve timing on critical path using RETIMING_BACKWARD property.		
RQS_TIMING-33-1	RQS_TIMING-27-1		synth_design	synth_design	Yes	No	Improve module level timing by using BLOCK_SYNTH.RETIMING property.		
RQS_TIMING-44-1	✓ □ XDC	1							
~ XDC	RQS_XDC-1-1	v	synth_design	synth_design	No	No	Paths above Max Net/LUT budgeting. Review paths and either reduce logic I		
RQS_XDC-1-1									
Clocking	RQS_CLOCK-9-1		synth_design	place_design	Yes	Yes	Sub optimal Fvco on MMCM/PLL. Update MMCM/PLL settings to improve the		
RQS_CLOCK-9-1	<								
	ML Strategies are available	only in	default/explore at su	ccessfully routed desig	n.				
			Reset Default	Export Suggestions	Add Sugges	stions to P	roject		

9. Click **RQS_TIMING-19-1** and select the timing path. Careful examination of these paths show there is no timing failure on these paths.



- High fanout net driven by a LUT a1_2r_reg expanded sig1 reg[0] FO=6015 FO=6015 expanded_sig1[1999] i 1 FO=2006 FO=2000 FO=1FO=1 FO=2000 Q 10 0 FO=1 FO=1Q FO=2001 FO=1FO=6006 FO=6006 EQN=4'h8 FDRE FDRE
- 10. Press F4 to load the schematic. You can see this is a high fanout net driven by a LUT.

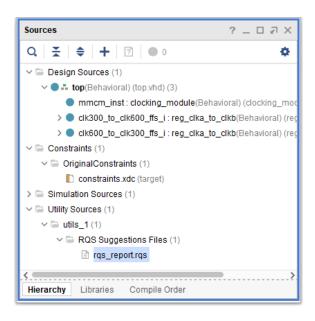
This suggestion is slightly different to the other retiming suggestions. When RQS identifies paths with a difficult profile that it can improve, it attempts to do so regardless of whether there is a timing error.

- 11. Note RQS_TIMING-27. This suggestion uses BLOCK_SYNTH properties at synthesis and applies to the entire module. You can determine the target cell by looking at the properties of the object as described in step 4. *Uncheck* the box for **RQS_TIMING-27**. This suggestion applies to the entire module and has overlap with the other suggestions. For clarity, it will not be applied.
- 12. With the remaining boxes checked, click **Add Suggestions to Project**. When the Export Suggestion dialog box opens, change the file name to **rqs_report.rqs** and select **Copy sources to project** as shown in the following figure.

\lambda Export Suggestion		×
Save selected suggestions to project	1	4
Save suggestions as:	rqs_report.rqs	⊗
	O	Cancel

13. Examine the Sources window. You will see that the RQS file has been added to the utils_1 fileset. This ensures that the file is captured using the get_files command, project archives and recognized in the next step when we add the file to a run.





Step 4: Run with Suggestions

You will now add a suggestion to a run and examine what happens when a suggestion is applied and how it is reported.

1. In the Design Runs window, click the + icon and generate both a new Synthesis and Implementation run. Ensure both runs have the Default run strategy and in the final step, select **Do not launch now**.

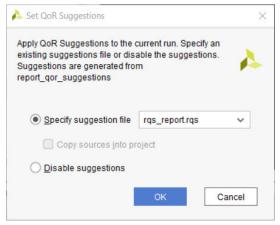
🝌 Create New Runs					×
Launch Options Configure hosts for la	unching runs, and/or set advanced lau	nch options			4
Launch <u>d</u> irectory:	Sefault Launch Directory				~
Options <u>Launch run</u> <u>G</u> enerate so <u>O</u> not laun		~			
?		< <u>B</u> ack	Next >	<u>F</u> inish	Cancel

2. In the Design Runs window, right-click on the new **synth_2** run and select **Set QoR Suggestions**



	Synthesis Run Properties	Ctrl+E
×	Delete	Delete
	Make Active	
	Change Run Settings	
	Set Incremental Synthesis	
	Include Incremental Synthesis Information in DCP	
	Set QoR Suggestions	
	Save As Run Strategy	
	Save As Report Strategy	
	Display Messages	
	Copy Run	
÷	Create Runs	
	Open Run Directory	
	Export to Spreadsheet	

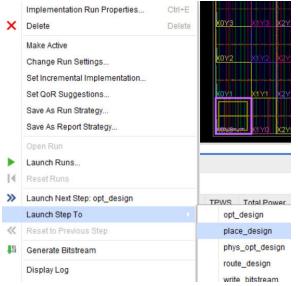
3. Specify the suggestion file as the RQS file added to the project from the previous step and click **OK**.



- 4. Repeat steps 2 and 3 for the implementation run. Specify the same RQS file for each run.
- 5. In the Design Runs window, right-click on synth_2 and select Make Active.
- 6. In the Flow Navigator, click **Run Synthesis**.



7. As this design takes a long time to route, we will only run to place_design and analyze at this stage. When synthesis is complete, in the Design Runs window, right-click on the new implementation run and select Launch Step To → place_design.



- With the implementation running, select the Design Runs window and right click → Open Run on the synth_2 synthesis run with QoR Suggestions. When opened select Reports → QoR Suggestions... and select Ok in the dialog box to run the command.
- 9. Click on the **RQA Summary**. You will see that the score has improved from 3 to 4.

QoR Suggestions		_ D 7 ×
Q ₹ ♦ C	Q RQA Summary	
General	QoR Assessment Score	Flow Guidance
v QoR Assessment	4 - Design runs have a good chance of success	Run report_qor_suggestions to generate suggestions. They may indirectly impr
RQA Summary		
Assessment Det		
ML Strategy Avai		
 Suggestion Report 		
GENERATED		
APPLIED V		
<>	C	````````````````````````````````
	Reset Default Export Suggestion	Add Suggestions to Project

10. Next click on the **Assessment Details**. You will see that the Net and LUT budget score has been reduced but not eliminated. This is a consequence of the high frequency we are forcing paths to run at in this design:



QoR Suggestions							@ ? >
Q ₹ ♦ C	Q Assessment Details						
General	Name	Threshold	Actual	Used	Available	Status	
✓ QoR Assessment	Timing						
RQA Summary	Utilization					ок	
Assessment Det	Clocking					OK	
ML Strategy Avai	Congestion					OK	
 Suggestion Report 	Number of paths above Max Net/LUT Budgeting	0	1			REVIEW	
APPLIED							
	Reset Default Export Suggestio	ns Add	Suggesti	ons to P	roject		

- 11. Close the synthesized design.
- 12. When place_design is finished, first go and examine the very top of the implementation log file for the new implementation run. You can see a table summary of the suggestions that have been read in. This summary helps confirm that what is read in is what you expect.

```
1. Read QOR Suggestions Summary
    ------
Read QOR Suggestions Summary
Suggestion Summary | Incr Friendly | Total |
    + - - - - - -
| Total Number of Suggestions |
                                 1 |
                                     5 |
                                 1 |
 Automatic
                          4 |
0 |
                                      1 |
 Manual
 APPLICABLE_AT
                                       _____
                                 synth_design
  opt_design
  That overlap with synthesis suggestions |
                                 0 |
                                      0 |
                                      1 |
  place_design
                                 1 |
  postplace_phys_opt_design
                                  0 |
                                      0 |
  route_design
                                  0 |
                                      0 |
                                 postroute_phys_opt_design
----+
```

- 13. Right-click on the implementation run and select **Open Run Directory**. Next open the checkpoint file by double-clicking on **top_placed.dcp**. This step is necessary as we are examining an intermediate run step in the interests of saving time.
- 14. In the new instance of Vivado tools, select **Reports** → **Report QoR Suggestions** ... and click OK.



15. In the new report, you will notice that there are more sections under Suggestion Report.

QoR Suggestions							- [N R C
Q ₹ ♦ C	Q <u>∓</u> ≑ 0 AP	PLIE	D					
Assessment De 🔨	ID		GENERATED_AT	APPLICABLE_FOR	AUTOMATIC	SCOPE	Incremental Friendly	DESCR
ML Strategy Ava	∽	1						
 Suggestion Report 	RQS_TIMING-33-1	1	synth_design	synth_design	Yes	GLOBALSCOPE	No	Improv
GENERATED	RQS_TIMING-19-1	1	synth_design	synth_design	Yes	GLOBALSCOPE	No	Retime
APPLIED	RQS_TIMING-44-1		synth_design	synth_design	Yes	GLOBALSCOPE	No	Improv
~ Timing	~ 🚍 Clocking							
RQS_TIMING-33-	RQS_CLOCK-9-1	1	synth_design	place_design	Yes	GLOBALSCOPE	Yes	Sub op
RQS_TIMING-19-	<			-				>
	ML Strategies are availab	le on	1	-				
			Reset Default	Export Suggesti	ons			

- GENERATED This is where new suggestions are listed
- EXISTING These are suggestions that existed previously but are not applied (Not Shown)
- APPLIED These are where suggestions that have been applied are listed
- FAILED TO APPLY Suggestions where the design objects no longer exist and consequently were not applied (Not shown)

Also note that the option to add to the project is not available when a checkpoint is opened. You can still export a file but it is not added to a project.

16. Click **APPLIED** and select the details table for one of the items. For APPLIED suggestions, the timing path summary is still available but it is not possible to cross probe to other views in Vivado as some items may have changed.

Step 5: Accumulating Suggestions

You can now review the newly generated suggestions and add them to the RQS file.

- 1. Click **GENERATED**. The RQS_CLOCK-15 message reports the high THS paths but does not provide an automatic suggestion.
- 2. Examine RQS_CLOCK-2-1. This suggestion recommends changing the clock buffers to BUFGCE_DIV to improve the timing path uncertainty. It is highly recommended to implement this. Because this suggestion is not automated, however, it requires an RTL edit. If you wish, you can make the recommendation and see the improvement, but this step can be skipped. The next steps focus on the automated suggestions.
- 3. Click on **RQS_CLOCK-1-1** to view the detailed report. This suggestion applies CLOCK_DELAY_GROUP to related clocks. In this report, you can see that there is a high clock skew and failing slack.



QoR Suggestions									_ [2 J X
Q I O C	Q RQS_	LOCK-1	-1							
General	Path Type	Skew	Slack	Req.	Datapath Delay	Cell%	Route%	Source Clock	Destination Clock	Source
 QoR Assessment RQA Summary Assessment Details ML Strategy Availabi Suggestion Report GENERATED APPLIED 	HOLD	1.011	-1.073	0.000	0.213	52.60	47.40	clk_600_clk_wiz_0	clk_300_clk_wiz_0	INBUF
Timing RQS_TIMING-33-1 RQS_TIMING-19-1 RQS_TIMING-44-1	ς		Re	set Defa	ult Export Su	ggestion	S			

Clock skew is difficult to identify before place_design because the skew estimate depends heavily on placement. As a consequence, RQS does not offer this suggestion unless a design is placed. Whenever there is a change in information level, it might be advisable to run report_qor_suggestions. The following summarizes the changes as you progress through the tool flow:

- Clocking estimates are accurate after place_design.
- Congestion is available after placement and improves further after routing.
- Timing estimates improve throughout the flow and are impacted by the number of paths analyzed.
- 4. Click Export Suggestions. When suggestions are exported, the APPLIED status is reset. All the previous suggestions and the new RQS_CLOCK-1-1 are combined into one file. You can overwrite the previous file and reuse the runs, or create a new file and runs.
- 5. Select the file location to overwrite the existing file. You can find out the location of this by selecting it in the sources window. Alternatively, it should be at the following location if you have followed the steps carefully <extract_dir>/Lab2/project_2/project_2.srcs/utils_1/imports/project_2.

You are now at the point where you know the fundamentals in handling RQS files and accumulating suggestions. If you have time, rerun implementation through to <code>route_design</code> and examine the impact of the latest suggestion. Alternatively generate alternative suggestions by running <code>report_qor_suggestions</code> on your own design.

Summary

In this lab, you achieved the following:

• Using RQA, you conducted an assessment before and after improvements were implemented, examining an improvement in the score.



• Using RQS, you conducted a complex analysis of a demonstration design. You first examined the reports that showed RQS recommendations to solve implementation problems, then you generated an RQS file and added it to a project implementation run. The Vivado implementation tools executed the suggestions automatically for you. You subsequently performed further analysis and generated further suggestions, accumulating them in the RQS file.





Lab 3

Running ML Strategies

Introduction

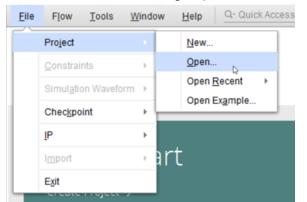
The report_qor_suggestions command can generate an implementation design strategy that is predicted to be optimal for the design using machine learning algorithms. In this tutorial, you will look at:

- How to generate ML strategy suggestions.
- How to setup the implementation run to use ML strategy suggestions.
- Reporting specifics related to ML strategies.

Step 1: Generating an ML Strategy RQS File

This step shows the process of opening a routed design with QoR Suggestions and generating a new RQS file with strategies. For details on the design, refer to Step 1: Understanding the Design.

1. In the Vivado Design Suite, go to File → Project → Open and select the project located in <extract_Dir>/Lab3/project_2.



- 2. In the Flow Navigator, click **Open Implemented Design**.
- 3. From the pull-down menus, select **Reports** \rightarrow **Report QoR Suggestions** ..., and click **OK**.



4. In the RQA Summary table, you will see the QoR Assessment Score and Flow Guidance. This table helps identify good candidate designs on which to use ML strategy suggestions. QoR assessment scores of 3 and above have a chance to meet timing. Designs with an RQA score of less than 3 are not prevented from generating ML strategies.

QoR Suggestions	– אנם –
Q X \$ C	Q RQA Summary
General	QoR Assessment Score Flow Guidance
 QoR Assessment RQA Summary Assessment Details ML Strategy Availability Suggestion Report GENERATED EXISTING APPLIED 	4 - Design runs have a good chance of success Run report_qor_suggestions to generate suggestions. They may i
~ Timing	× <
	Reset Default Export Suggestions Add Suggestions to Project

5. Click **ML Strategy Availability**. This table details the required directives for the reference run to generate strategies.

QoR Suggestions		_ 🗆 🗟 X
Q ¥ ♦ C	Q 🚺 ML Strategy Availability	
General	Conditions for ML Strategy Availability Value Status	
v QoR Assessment	opt_design directive Default OK	
RQA Summary	place_design directive Default OK	
Assessment Details	phys_opt_design directive Default OK	
ML Strategy Availability	route design directive Default OK	
 Suggestion Report GENERATED EXISTING APPLIED 		
~ Timing	ML Strategies are available only when the default/explore directives have been used in the implementation	on flow and 🗘
~ Timing	ML Strategies are available only when the default/explore directives have been used in the implementation Reset Default Export Suggestions Add Suggestions to Project	on flow and

The status for all directives must be listed as OK to generate strategies. The requirements are as follows:

- The opt_design directive value must be either Default or Explore.
- The place_design, phys_opt_design, and route_design conditions must be the same as each other and must be set to either Default or Explore.
- 6. In the Design Runs window, confirm the strategy is Vivado Implementation Defaults. This requirement is met when a design has been run with either the Vivado Implementation Defaults or the Performance_Explore strategy.
- 7. In the QoR suggestion report, select **GENERATED**. Three new strategies have been generated but they are not selectable because strategy generation is currently only available using Tcl.



QoR Suggestions								_ 🗆 🖉 🗙
Q ¥ ♦ C		Q 🔮 🌲 GENERA	TED					
General	^	ID		GENERATED_AT	APPLICABLE_FOR	AUTOMATIC	SCOPE ^1	Incremental Friendly
v QoR Assessment		~ 🚍 Clocking	\checkmark					
RQA Summary		P RQS_CLOCK-2-1		route_design	place_design	No	GLOBALSCOPE	No
Assessment Details		🗢 🗇 Strategy	0					
ML Strategy Availability		RQS_STRAT-17-1		route_design	none	No	GLOBALSCOPE	No
 Suggestion Report 		RQS_STRAT-66-1		route_design	none	No	GLOBALSCOPE	No
GENERATED		RQS_STRAT-71-1		route_design	none	No	GLOBALSCOPE	No
EXISTING APPLIED								
~ Timing	~	<)
		Reset Default	Expor	t Suggestions	Add Suggestions	to Project		

8. Select **RQS_STRAT-17-1**. Here, you can see the details of the strategy being suggested. It is possible to set these up manually, but to automate the process more easily, the recommended flow is to read an RQS file containing strategies and set the directive to RQS on the implementation commands.

QoR Suggestions				_ 🗆 🕫 🗙
Q X \$ C		Q RQS_STRAT-17	7-1	
General ~ QoR Assessment RQA Summary Assessment Details ML Strategy Availability ~ Suggestion Report GENERATED EXISTING APPLIED ~ Timing	Î	Command opt_design place_design phys_opt_design route_design	Options -directive ExploreArea -directive ExtraTimingOpt -directive AggressiveExplore -directive NoTimingRelaxation	
		Reset Default	Export Suggestions Add Suggestions to Project	

9. At the Tcl console, ensure you are in a suitable writable directory. This can be at the same level as the project. Issue the write_qor_suggestions command to write out the suggestions as shown in the following example.

```
file mkdir <extract_Dir>/Lab3/project_2/ML_STRAT
cd <extract_Dir>/Lab3/project_2/ML_STRAT
write_qor_suggestions -strategy_dir ./
```

This writes one RQS file per strategy. Each RQS file also contains all of the other suggestions that are not strategy suggestions. This ensures that you can use all the other QoR suggestions and that there is no confusion as to which strategy suggestion the tools should select.

Step 2: Creating ML Strategy Runs

In this step, you will use the files generated to create ML strategy project-based runs.

1. Examine the contents of the ML_STRAT directory.



Name	Date modified	Туре	Size
impl_3Project_MLStrategyCreateRun1.tcl	21/06/2021 14:19	TCL File	2 KB
impl_3Project_MLStrategyCreateRun2.tcl	21/06/2021 14:19	TCL File	2 KB
impl_3Project_MLStrategyCreateRun3.tcl	21/06/2021 14:19	TCL File	2 KB
impl_3SuggestionFile1.rqs	21/06/2021 14:19	RQS File	8 KB
impl_3SuggestionFile2.rqs	21/06/2021 14:19	RQS File	8 KB
impl_3SuggestionFile3.rqs	21/06/2021 14:19	RQS File	8 KB
NonProject_MLStrategyCreateRun1.tcl	21/06/2021 14:19	TCL File	1 KB
NonProject_MLStrategyCreateRun2.tcl	21/06/2021 14:19	TCL File	1 KB
NonProject_MLStrategyCreateRun3.tcl	21/06/2021 14:19	TCL File	1 KB

You can see the following contents:

- 3 x RQS files
- 3 x project-based Tcl scripts
- 3 x non-project-based Tcl scripts

The RQS files are common for both project and non-project flows. The non-project scripts are examples of how to use the RQS file. The project-based scripts can be sourced. Each of the three scripts references one of the RQS files. All three should be sourced.

2. Source each of the project-based Tcl files. Each one creates a run in the Design Runs window, sets up the RQS file, and sets the directives to RQS. The run options are copied from the reference run.

```
source ./impl_3Project_MLStrategyCreateRun1.tcl
source ./impl_3Project_MLStrategyCreateRun2.tcl
source ./impl_3Project_MLStrategyCreateRun3.tcl
```

3. In the Design Runs window, select **impl_2_ML_Strategy_1**.

Q ≚ ♦ H ≪ ▶	» + %									
Name	Constraints	Status	Incremental	WNS	TNS	WHS	THS	WBSS	TPWS	Total
✓ ✓ synth_1_rqs	OriginalConstraints	synth_design Complete!	Off							
✓ impl_3 (active)	OriginalConstraints	route_design Complete, Failed Timing!	Off	-0.168	-78.372	0.003	0.000		0.000	
impl_3_ML_Strategy_1	OriginalConstraints	Not started	Off							
impl_3_ML_Strategy_2	OriginalConstraints	Not started	Off							
impl_3_ML_Strategy_3	OriginalConstraints	Not started	Off							

- 4. In the Implementation Run Properties window, select the **Properties** tab and confirm that RQS_FILES is set.
- 5. In the Implementation Run Properties window, select the **Options** and confirm the directive is set to RQS for the <code>opt_design</code>, <code>place_design</code>, <code>phys_opt_design</code>, and <code>route_design</code> commands.



Implementation Run Properties				? _ 🗆	N N
impl_1_ML_Strategy_1			+	⇒ 0	\$
Incremental Implementation:	Not	set			î
<u>S</u> trategy:	<u>1</u> 0 '	Vivado In	nplementati	on 👻	
Description:	Defa	ault settii	ngs for Imple	ementation.	
✓Design Initialization (init_des	sign)				
tcl.pre					$\overline{\ }$
tcl.post					•••
✓Opt Design (opt_design)					
is_enabled			\checkmark		
tcl.pre					
tcl.post					
-verbose			\cap		
-directive*		RQS			~
More Options					-
✓Power Opt Design (power_o	pt_de	esign)			~
General Properties Optic	ons	Log	Reports	Messages	

You are now set up to run with ML strategies. By the time you have an ML strategy file, you cannot generate new strategies after design changes, but you can add other suggestions.

6. You are now ready to launch the runs. Select all the ML strategy runs, right-click, and select **Launch Runs...**. The runs will now proceed in parallel, and complete like a standard run.

Summary

In this lab, you achieved the following:

- Used report_qor_suggestions to generate ML strategies.
- Created the RQS and Tcl files using write_qor_suggestions.
- Sourced the Tcl to set up the ML strategy runs and confirmed the key aspects of setting up an ML strategy run.





Lab 4

Intelligent Design Runs

Introduction

In this tutorial, you will work with Intelligent Design Runs (IDRs). An IDR is a simple-to-use implementation run that extends the performance of your design by automating QoR suggestions and ML strategies. These features are applied in a way that maximizes the improvement in performance.

This lab covers how to:

- Create an IDR.
- Add Tcl hooks to the IDR.
- Examine the log file for an IDR.
- Examine reports for an IDR.
- Create a single pass run from an IDR.

Step 1: Creating Intelligent Design Runs

In this section, you will create an example design and an Intelligent Design Run (IDR).

1. First, you need a project. Select Open Example Project.



2. Click $Next \rightarrow BFT$ and then select Next again.



Templates	Description
Q ¥ ♦ ▲ ⊕ •€ ▲ AXI DMA ■ ■ ■ ■ ④ CFPS DDR PL Debug ○ CFU (HDL) ② CPU (HDL) ● ● ③ CPU (HDL) ● ● ● CPU (Synthesized) ● ● ● MicroBlaze Design Presets ● Multi-Rate GTY ● Vavegen (HDL) ■ 2ynq UtraScale+. MPSoC Design Pre ■ Zynq UtraScale+. MPSoC Design Pre ■ ■ aidma ■	BFT Small RTL project

- 3. In the next screen, ensure you are working on in a suitable writable directory, and select **Next**.
- 4. For part selection, you *must* select xcku035-fbva900-2-e and then select $Next \rightarrow Finish$.

		Ope	en Example P	roject (or	xcofisapps	:001)			
fault Part ose a default Xilinx par	t for your projec	ct.							1
Search: Q-		~							
Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs	Gb Transceivers	GTPE2 Transcei
xc7k70tfbg484-2	484	285	41000	82000	135	0	240	4	0
xc7k70tfbg676-2	676	300	41000	82000	135	0	240	8	0
xc7v585tffg1157-2	1157	600	364200	728400	795	0	1260	20	0
xcku035-fbva900-2-e	900	468	203128	406256	540	0	1700	16	0

You should now have an open project that is waiting to be synthesized.

Note: IDRs are not supported for 7 series devices. Selecting the incorrect family requires you to regenerate the project.

5. This lab aims to demonstrate how to add a Tcl file to an IDR. To do this, add a file to impl_1. In the Design Runs window, select **impl_1**.



- 6. In the Implementation Run Properties window, select the **Options** tab.
- 7. Scroll to the <code>opt_design</code> tcl.pre option and add the Tcl script <extract_dir>/Lab4/ pre_opt.tcl to the option.

Implementation Run Prope	rties ? _ 🗆 🛙	3
> impl_1	← ⇒ 0	1
Incremental Implementation:	Not set]
<u>S</u> trategy:	🔓 Vivado Implementation Defaults* (Vivado Implementation 2021) 👻 💾	
Description:	Default settings for Implementation.	ľ
~ Design Initialization (init_de	esign)	
tcl.pre	E	3
tcl.post	E	-
More Options		
∨Opt Design (opt_design)		
is_enabled	$\mathbf{\nabla}$	
tcl.pre*	/Lab4/post_opt.tcl	-
tcl.post	E	•
-verbose		
-directive	Default 🗸	
More Options		
General Properties O	ptions Log Reports Messages	

Alternatively, use the following Tcl syntax:

```
add_files -fileset utils_1 -norecurse <extract_dir>/Lab4/post_opt.tcl
set_property STEPS.OPT_DESIGN.TCL.PRE [ get_files <extract_dir>/Lab4/
post_opt.tcl -of [get_fileset utils_1] ] [get_runs impl_1]
```

- 8. In the Flow Navigator, click **Run Synthesis**.
- The next step is to create the Intelligent Design Run. In the Design Runs window, right-click on impl_1→ Close Timing using Intelligent Design Runs.

						20.1
<u>S</u> trategy:		🔓 Vivado Implem	entation	n Defaults* (V		Implementation Run Properties
Description:		Default settings fo	or Impler	mentation.	×	Delete
∼Design Initia	alization (init_de	sign)				Make Active
tcl.pre						Change Run Settings
tcl.post						Set Incremental Implementation
More Optic	ons					Close Timing using Intelligent Design Runs
∨ Opt Design	(opt_design)					Set QoR Suggestions
is_enabled	L					Save As Run Strategy
tcl.pre*				, i		Save As Report Strategy
General Pr Cl Console	Messages	Log Reports	Intelli	Messages igent Desig	•	Open Run Launch Runs
Q <u>∓</u> ≑	≪ ▶	» + %			>	Reset Runs Launch Next Step: opt design
lame	Constraints	Status		Increment	1.1	Launch Step To
🗸 synth_1	constrs_1	synth_design Com	plete!	Off	<<	Reset to Previous Step
⊳ impl_1	constrs_1	Not started				Generate Bitstream

If you try this step again to create an additional run, you will find that the option is disabled. This is to prevent the creation of runs that would produce the same results. Runs created from the same netlist with different directives produce the same results because the IDR controls the directive, meaning that they are effectively the same.



If you have a different floorplan or speed grade, create different implementation runs. You can create one IDR from each implementation run.

Step 2: Navigating Intelligent Design Runs

In this step you will learn how to control Intelligent Design Runs and navigate options and reports.

1. Select the Intelligent Design Runs tab at the bottom of the Vivado[®] IDE.

Tcl Console Messages Log Re	ports Inte	lligent Design Runs 🛛 🗙 Desi	gn Runs					1	- 0	3.0
Q ≚ ♦ 14 ►										
Name	Constraints	Status	Incremental	WNS	TNS	WHS	THS	V	TRAM	10
✓ ▷ i_impl_1_1	constrs_1	Not started	Off					1		
		Child Runs Not Started						0		
▷ i_impl_1_1_rqs	constrs_1	Not started	Off					1		
> > Stage 2: Tool Option Exploration		Child Runs Not Started						1		
✓ ▷ Stage 3: Last Mile Timing Closure		Child Runs Not Started								
i_impl_1_1_incr_rqs	constrs_1	Not started	Off					1.		

This window is opened when an IDR is created. If closed, it can be reopened by selecting **Window** \rightarrow **Intelligent Design Runs**. From this window you can see that there is a hierarchical nature to Intelligent Design Runs. The top-level run is split into three stages:

- Stage 1: Design Optimization
- Stage 2: Tool Option Exploration
- Stage 3: Last Mile Timing Closure

In this lab, only Stage 1 is demonstrated. This is the most complex stage, but can be completed quickly because you are working with a simple design.

2. Expand Stage 2: Tool Option Exploration.

Tcl Console Messages Log Re	ports Intel	lligent Design Runs 🛛 🗙 Des	ign Runs							
Q 꽃 ♦ 14 ►								<i>C</i>		
Name	Constraints	Status	Incremental	WNS	TNS	WHS	THS	Wb	"AM	10
✓ ▷ i_impl_1_1	constrs_1	Not started	Off							
		Child Runs Not Started						C		
i_impl_1_1_rqs	constrs_1	Not started	Off					1		
		Child Runs Not Started								
i_impl_1_1_ml_strat_1	constrs_1	Not started	Off							
i_impl_1_1_ml_strat_2	constrs_1	Not started	Off							
i_impl_1_1_ml_strat_3	constrs 1	Not started	Off							

There are three runs underneath this stage. These runs can be run in parallel if your compute resources allow.

3. Right-click $i_impl_1_1 \rightarrow Launch Runs$ and select how you normally run jobs. Click OK.



Launch R	luns	
Launch the selected synthesi	s or implementation runs.	4
Launch <u>d</u> irectory: 두 <defa< th=""><th>ault Launch Directory></th><th>~</th></defa<>	ault Launch Directory>	~
Launch runs on local I	host: Number of jobs: 6	~
Launch runs on remot		
Launch runs on Cluster	er Isf_synth	~ Q
◯ <u>G</u> enerate scripts only		
?	ОК	Cancel

- 4. Select **impl_1_1** and view the **Properties** in the Implementation Run Properties window. A PARENT property is set to synth_1. As is the case with a normal implementation run, there is a dependency on the synthesis run to complete when this is set. If the RTL code is modified, the run will go out of date.
- 5. Click the **Design Runs** tab and confirm that the synth_1 run is running or finished.
- 6. Click back to **Intelligent Design Runs** and right-click on the **impl_1_1** top-level run.

	Implementation Run Properties	Ctrl+E
×	Delete	Delete
	Launch Runs	
M	Reset Runs	
	View Reports	
	Open Run Directory	
	Open Run	
	Generate Single Pass Implementation Run	
	Export to Spreadsheet	

The right-click menu from the top-level run is the main way to control the Intelligent Design Runs. Some options only become available at the right time:

- Reset Runs is only available after a run is launched.
- View Reports is only available after reports are generated.
- Open Run is only available after a run is successfully completed.
- Generate Single Pass Run is only available after a run is successfully completed.



7. Select the lower-level run, i_impl_1_1_rqs, and right-click on it.

Open Run Directory
Open Run
Export to Spreadsheet

When the lower-level run is selected, you can see that there is a reduced set of options. The Open Run Directory and Open Run options are scoped to the lower-level run. If Open Run is selected, the best completed routed run is opened.

Step 3: Analyzing the Reports and Log File

In this task, you will see where to find reports and analyse runs and intermediate checkpoints from an Intelligent Design Run.

If the window is not already open, select Reports → Intelligent Design Run Reports. This
window captures all the metrics that are captured throughout the IDR and provides links to
any generated reports.

Project Summary × Intel	ligent Desig	n Runs Report	s ×									
i_impl_1_1												
Flow Progress												
Q 🗄 🖨												
Stage/Run/Steps	opt_design	place_design	phys_opt_design	route_design	pos	stroute	_phys_c	opt_des	sign			
~ Design Optimization												
~i_impl_1_1_rqs												
CLEANUP_XDC	1		¥.				-			1		
CLEANUP_UTILIZATION				· · · ·			20					
FIRST_PASS\$*	-	~	~	~			÷.			\		
CLEANUP_CLOCKING		•5								1		
CLEANUP_CONGESTION	× .		•	•			-			- 1		
CLEANUP_TIMING		53	<u>*</u> 2				-					
Tool Option Exploration												
i_impl_1_1_ml_strat_1		53										
i_impl_1_1_ml_strat_2												
i_impl_1_1_ml_strat_3		12	5				- 24			1		
✓ Last Mile Timing Closure												
i_impl_1_1_incr_rqs	- ÷											
Flow Statistics												
Q ₹ ≑												
Stage/Run/Steps	Repo	orts			WNS	TNS	WHS	THS	RQA	Global Con	Level (N-E-S-W)	Global Cong Tile% N-E
v Design Optimization												
<pre>~ i_impl_1_1_rqs</pre>												
~ CLEANUP_XDC												
opt_design		postopt_xi postopt_xi postopt_xi postopt_xi	dc_rqa.rpt	rpt			2	-	4		24	20
~ CLEANUP_UTILIZATION												
opt_design												
place_design					-		~	•				-
phys_opt_design							-					
route_design							-				-	-
/	decian											>
	only. Use rep age.	ort timing summ	ary to get accurate	timing number	_					-(5



The report is broken into two windows. The Flow Progress section is in the top half. This details which steps have been run and which steps are running currently. This is more important for an IDR as opposed to a standard implementation run for the following reasons:

- The IDR is a dynamic run, and not all stages are run for every design.
- The steps might be run repeatedly as the run is reset to apply QoR suggestions.

The Flow Statistics section is in the bottom half, and provides the following:

- Hyperlinks to any available reports
- Timing metrics such as WNS/TNS/WHS/THS
- RQA score
- Congestion level and tile % metrics for post-place and post-route designs

2. Focus in on the Flow Progress section of the report.

low Progress					
Q ₹ ♦					
Stage/Run/Steps	opt_design	place_design	phys_opt_design	route_design	postroute_phys_opt_design
 Design Optimization 					
∨i_impl_1_1_rqs					
CLEANUP_XDC	~	24	543	5 m	1.41
CLEANUP_UTILIZATION			(*)		100
FIRST_PASS\$*	-	~	~	~	1943
CLEANUP_CLOCKING				1.20	0.00
CLEANUP_CONGESTION	•			1.00	0.55
CLEANUP TIMING	•			20-22	

- Green ticks indicate a completed stage.
- Hyphens indicate the stage is not run.
- Circles indicate the stage is running.

Because there are no circle icons, you can infer that the run has completed successfully. Note also the FIRST_PASS step, which is generated for reporting purposes. This special step occurs when there are no utilization or clock suggestions. Only designs without these suggestions have this step.

At the bottom, there is a table footer that describes the meaning of the the \$ and * notations. These notations help you identify which runs you are opening when you select **Open Run**.

3. Look at the **Flow Statistics** section of the report. This report can be larger, and you can make more space by reducing the divider between the section above and clicking on the arrows to reduce sections of the report.



i_impl_1_1 Flow Progress										
Q <u>¥</u> ¢										
Stage/Run/Steps opt_	design	place_design	phys_opt_design	route_desi	gn pos	stroute	_phys_o	pt_desi	gn	
V Design Optimization										
~i_impl_1_1_rqs		Ho	ver betw	een						
low Statistics		110						-	× .	
Q ₹ ♦			s and pu							
Stage/Run/Steps	Repo	botton	n section	of th	ens	TNS	WHS	THS	RQA	Global Cong Level (N-
<pre>vi_impl_1_1_rqs</pre>			the Party of the P	~~~~	1					
> CLEANUP_XDC			report							
> CLEANUP_UTILIZATION			-							
V FIRST_PASS\$*										/
opt_design										
place_design		postplace_first postplace_first postplace_first postplace_first	2.051	0.000	22	3	3	. \		
phys_opt_design	pos pos	stplace_physopt stplace_physopt stplace_physopt stplace_physopt	2.051	0.000	n	2	3			
route_design		postroute_first postroute_first postroute_first postroute_first	_pass_rqa.rpt	ary.rpt	1.407	0.000	0.000	0.000	5	. (
postroute_phys_opt_desig	n									
> LEANUP_CLOCKING										
> LEANUP CONGESTION > LEANUP COllapse > lool Shifteren > Last Mile Timing Closure	the	ese sec	tions							

On the left of this window are the reports, in the middle are the timing and RQA statistics, and on the right are the congestion metrics. Statistics are only available if the flow stage has been run, or if the flow stage generates the statistics. For example, during place, hold statistics are not generated.

4. Click **postplace_physopt_first_pass_util.rpt**. Pay attention to the name of this report: postplace_phys_opt indicates the implementation step and first_pass indicates the flow step. When you are ready, close the report.



- 5. In the Intelligent Design Runs window, right-click **impl_1_1** and select **Open Run Directory**. In this directory, open the **idrTextReport.txt** file. This is the text equivalent to the IDE report. It contains the same information except the links to the report. When you are ready, close the file.
- 6. In the directory explorer, go up one level. You will be in the project_1.runs folder. There is a directory for the main IDR and each of the sub-runs. Click into **impl_1_1_rqs**. Here, you will see all the reports and intermediate checkpoints. When you are ready, close the directory explorer.
- 7. In the Intelligent Design Runs window, select the top-level run impl_1_1.
- 8. In the Implementation Run Properties window, select the **Log** tab and maximize the window.
- 9. Select the search icon and enter "IDR 1-60". Click **Next** multiple times to navigate through the log file. This ID highlights the start and end banners that have been added when you enter and exit a step within Stage 1: Design Optimization.

Implementation Run Properties	? _ 🗆 🖾 ×
✓ i_impl_1_1	♦ ⇒ Q
Q. E	
IDR 1-60 V Next Previous Highlight	□ <u>M</u> atch Case □
INFO: [IDR 1-601] XDC cleanup phase started Command: opt_design -directive Explore INFO: [Vivado_Tcl 4-136] Directive used for opt_design is Attempting to get a license for feature 'Implementation' INFO: [Common 17-349] Got license for feature 'Implementa Running DRC as a precondition to command opt_design	and/or device '
General Properties Log Messages	·····>``

As the implementation process goes back and forth, it can be difficult to identify which step you are in. These messages will help clarify this.

10. Search for "INFO-TUTORIAL4". This has been inserted by the Tcl script you added in Step 1. This Tcl script is executed every time <code>opt_design</code> is called. In this case, <code>opt_design</code> is called only once, so it is similar to a normal flow.

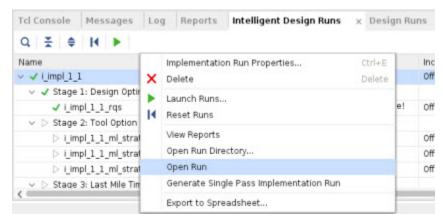
Step 4: Final Analysis

In this task, you will see how to analyze the design and create a single pass run.





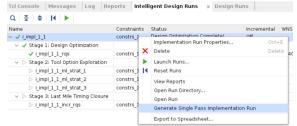
1. In the Intelligent Design Runs window, right-click on the top-level run, **impl_1_1**, and then click **Open Run**. Doing this opens the best overall run from the three stages. This is also the best run from stage 1: stage 2 and stage 3 were not run with this design, and we have seen the best run and best stage run indicated by the \$ and * notations.



A device view opens where you can generate reports and analyze the design as you would with a normal implementation run. When you are ready, close the design.

Note: It is not possible to make an IDR the active run.

- 2. In the Intelligent Design Runs window, right-click on the top-level impl_1_1_rqs and click Open Run Directory. This directory has all the intermediate checkpoints from the stage. You can open a new instance of Vivado and open the checkpoints for further analysis. If the DCP is associated with Vivado, you can open the checkpoints by double-clicking on them.
- 3. When you are finished with analyzing the design, because an IDR can take the equivalent of many implementation runs to complete, it is recommended to run the required commands and suggestions in a single implementation run. In the Intelligent Design Runs window, right-click on the top-level impl_1_1 and select Generate Single Pass Implementation Run.



4. In the Create Single Pass Implementation dialog box, enter a run name, impl1_a, and click OK.



entation I ×
run 🝌
impl1_a
Cancel

5. Switch to the **Design Runs** window. You can see that the new run has been made. Right-click on the run and select **Launch Runs**. Confirm that the results are the same as the IDR.

Note: For more difficult runs, you can examine the RQS_FILES property and directives, but because this is an easy-to-meet-timing design, these are not set.

Summary

In this tutorial, you learned how to:

- Create and launch Intelligent Design Runs.
- Analyze reports and checkpoints generated by an IDR.
- Create a single pass flow.



Appendix A

Additional Resources and Legal Notices

Please Read: Important Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at https:// www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING



OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

Copyright

© Copyright 2012-2021 Xilinx, Inc. Xilinx, the Xilinx logo, Alveo, Artix, Kintex, Spartan, Versal, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

