Vivado Design Suite Tutorial

Design Analysis and Closure Techniques

UG938 (v2018.3) December 14, 2018

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
12/14/2018 Version 2018.3	
Lab 1: Setting Waivers with the Vivado IDE	Updated IDE screenshots.
Lab 2: Using Report QoR Suggestions	Updated all steps with new instructions.
06/29/2018 Version 2018.2	
Lab 2: Using Report QoR Suggestions	Added Lab 2: Using Report QoR Suggestions.
04/27/2018 Version 2018.1	
General updates	Initial Xilinx release for 2018.1.

Table of Contents

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.

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 2018.3 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>`.

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 `report_cdc`, `report_drc`, or `report_methodology` commands. Waived checks are also filtered out from the mandatory DRCs run at the start of the implementation commands, such as `opt_design`, `place_design`, and `route_design`. 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 (`.dcp` 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*** *>* ***All Programs*** *>* ***Xilinx Design Tools*** *>* ***Vivado <version>*** *>* ***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
   ```

   Note: 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** > **Timing** > **Report CDC**.

2. In the Report CDC dialog box, leave the default settings as-is, and click **OK**.

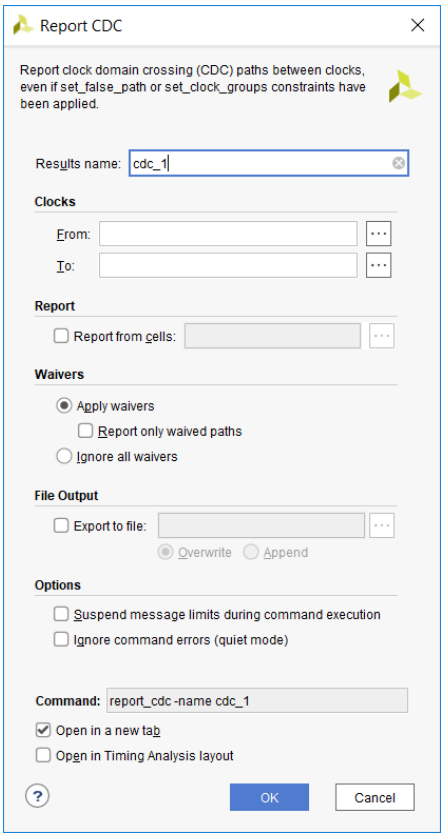


Figure 1: Report CDC Dialog Box

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

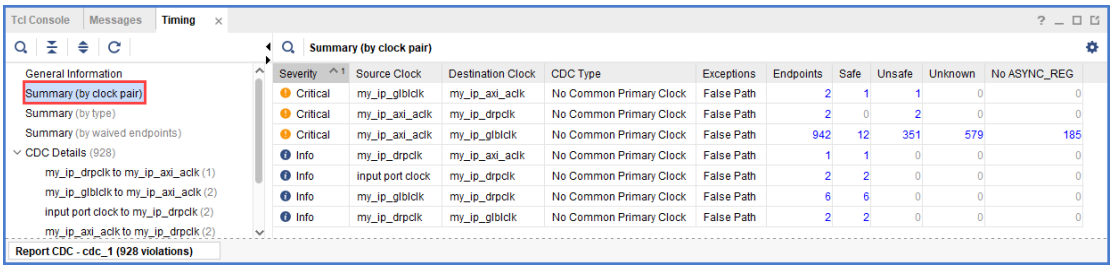


Figure 2: Summary by Clock Pair Section in the CDC Report

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

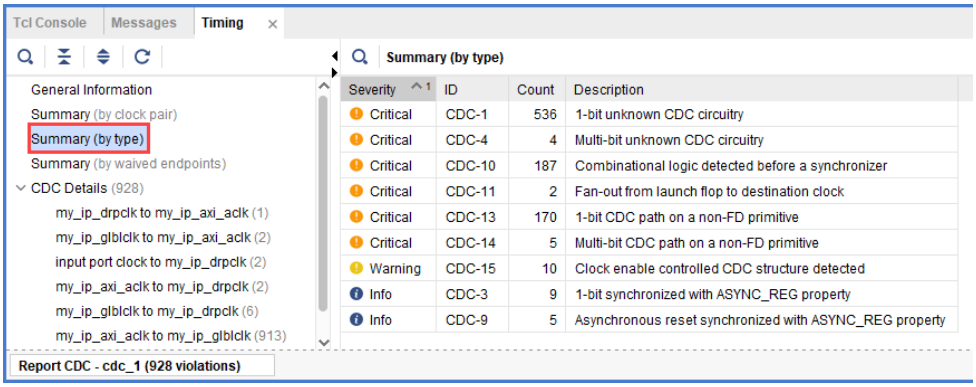


Figure 3: Summary by CDC Type Section in the CDC Report

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.

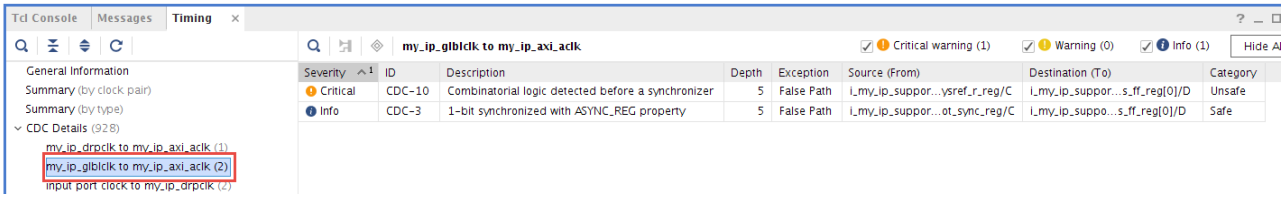


Figure 4: my_ip_glblclk to my_ip_axi_aclk CDC Violations

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

   *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.*

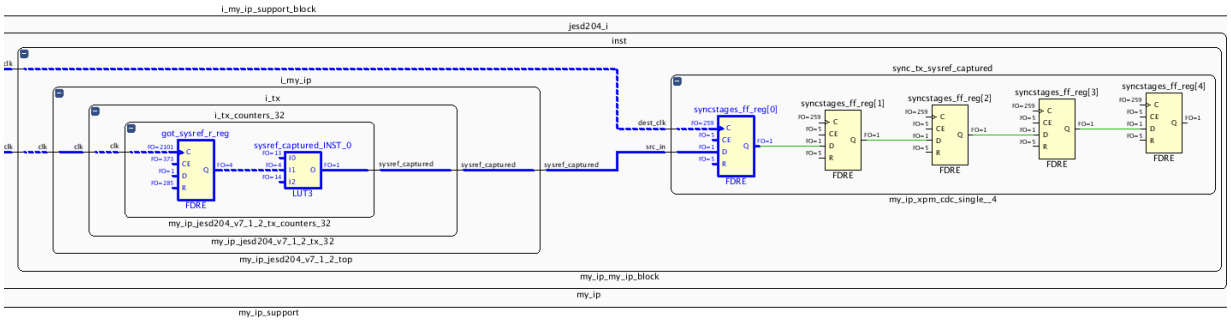


Figure 5: CDC-10 Violation Schematic

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**.

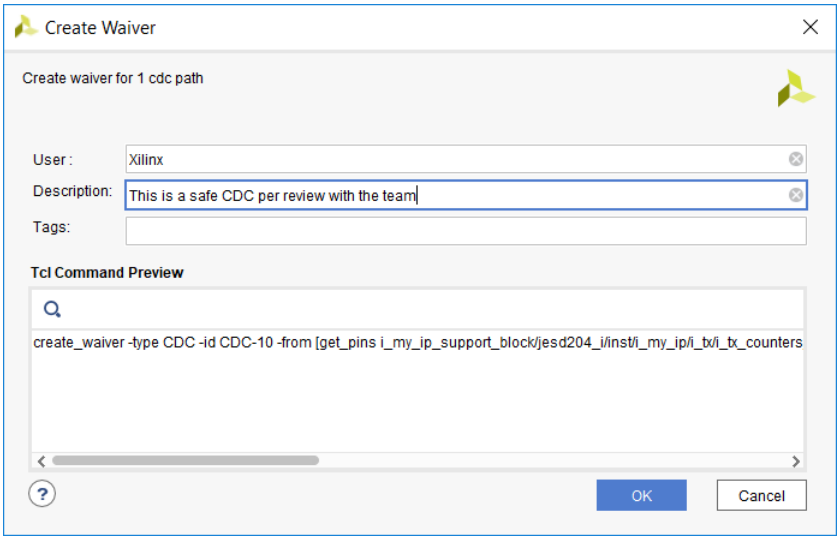


Figure 6: Create Waiver Dialog Box for CDC-10 Violation

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.

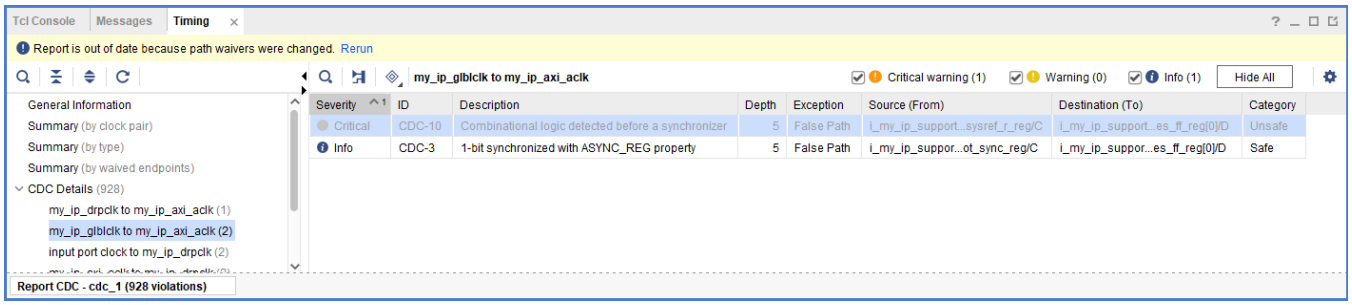


Figure 7: CDC Report with CDC-10 Violation Waived

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.

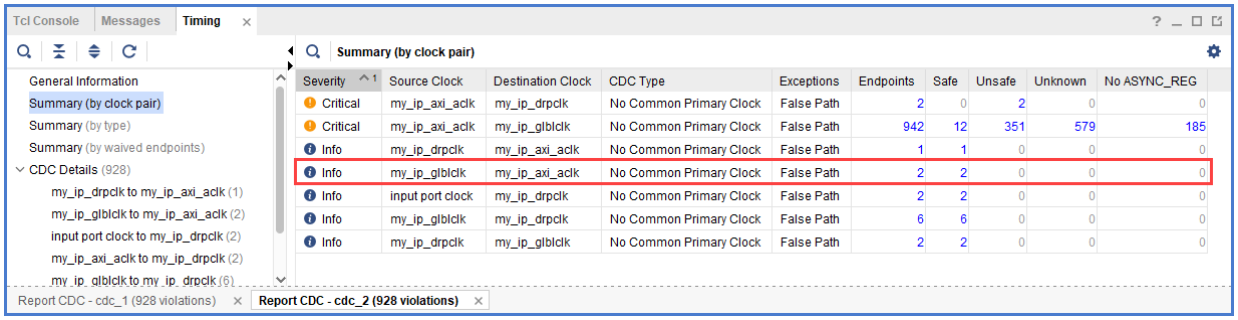


Figure 8: Summary by Clock Pair Differences

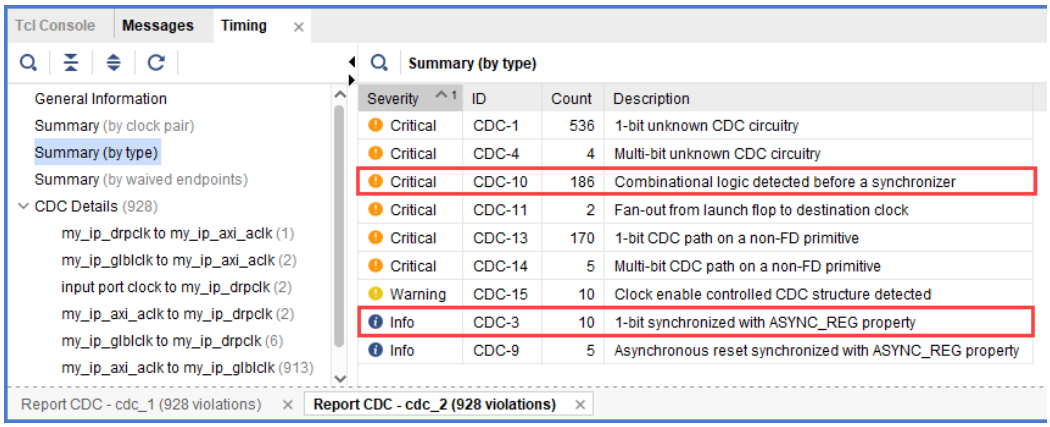


Figure 9: Summary by Type Differences

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

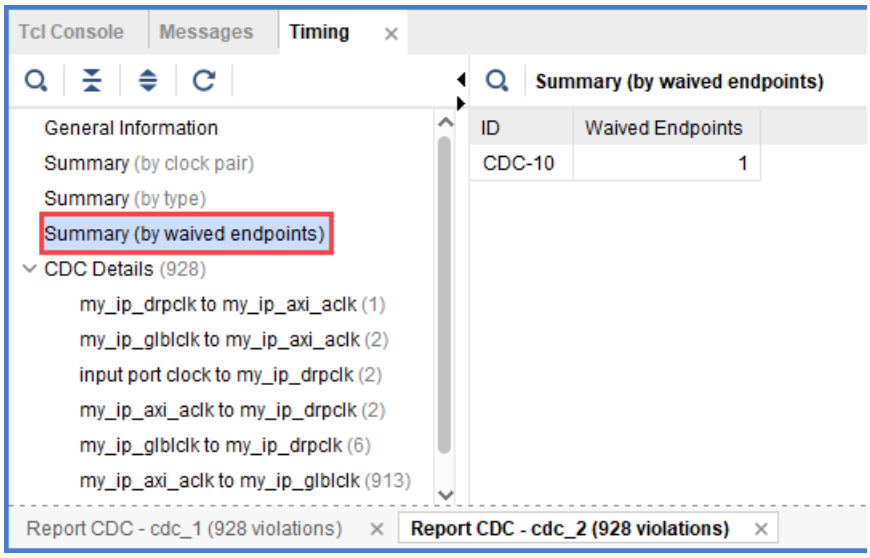


Figure 10: Summary by Waived Endpoints

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.

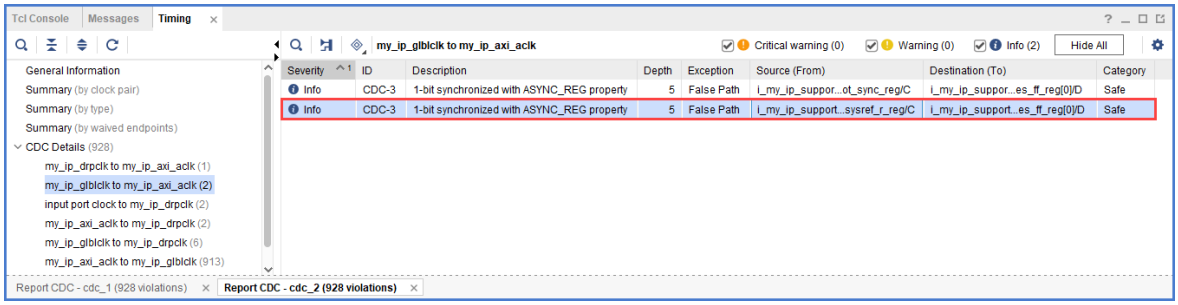


Figure 11: Detailed Report

7. Select the new **CDC-3** row, and click the **Schematic** toolbar button .

   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.

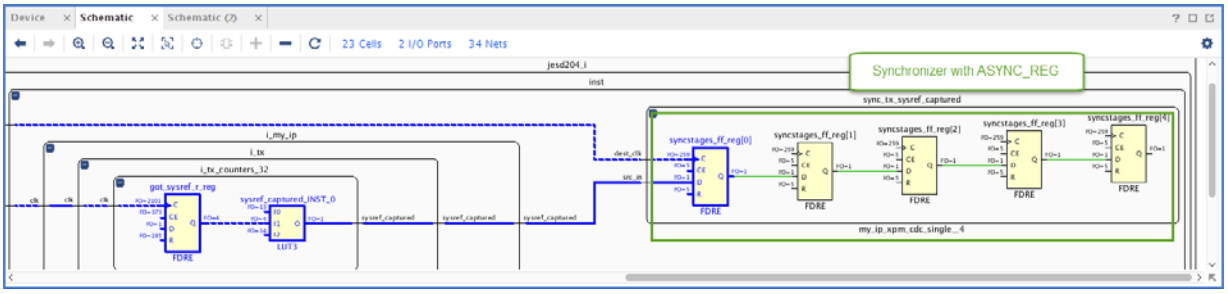


Figure 12: CDC-3 Schematic

IMPORTANT: *In this lab, Report CDC only reports a single violation per endpoint and per clock pair. When multiple violations apply to a specific 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`.

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
   ```

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

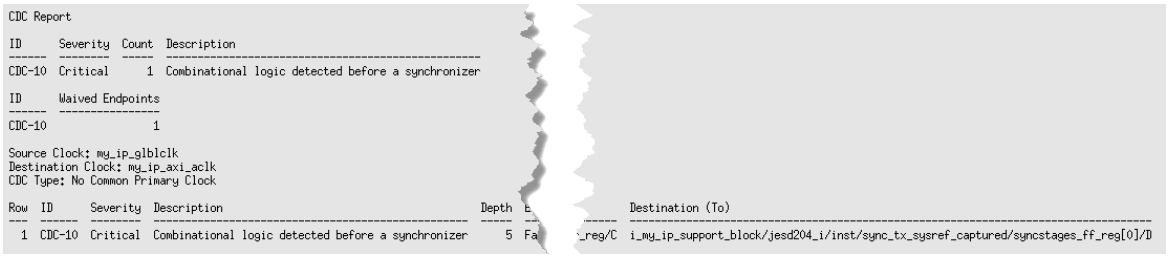


Figure 13: CDC Report

Generating a Vivado IDE CDC Report for Waived Violations

1. Select **Reports > Timing > Report CDC**.
2. In the Report CDC dialog box (Figure 1), 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.

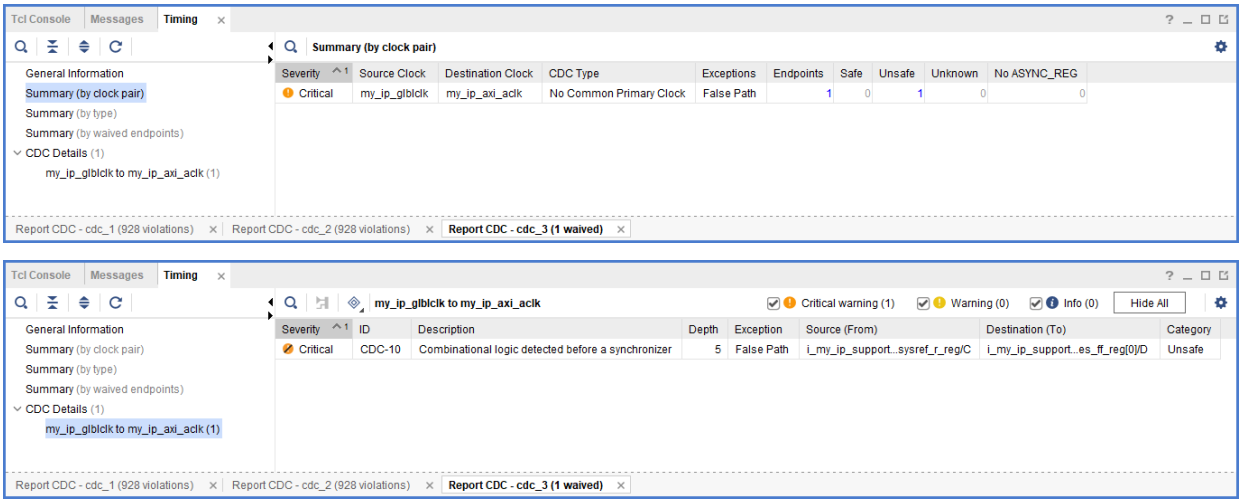


Figure 14: Summary by Clock Pair and CDC Details Reports

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.

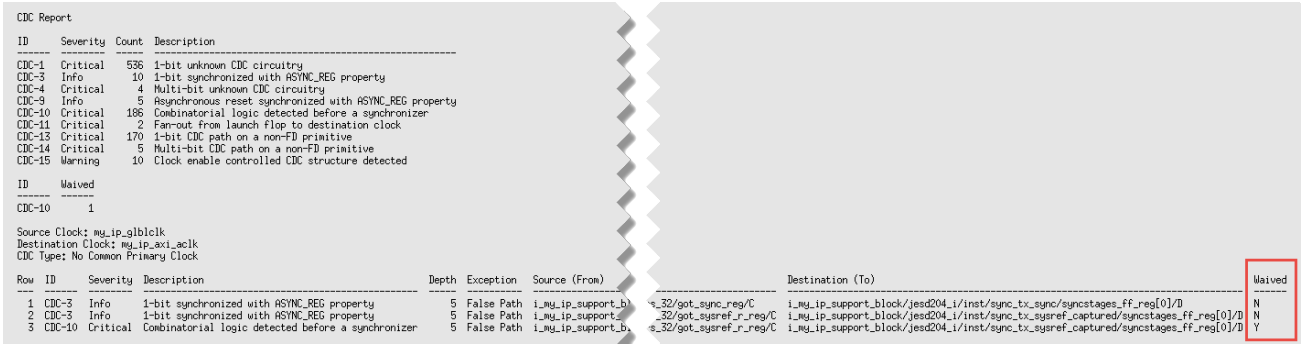


Figure 15: Text Report with Waivers

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 Report

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

Figure 16: Text Report Before Waiver

TIP: *You can also generate a list of all violations regardless of the waivers from the Vivado IDE. Select* ***Reports > Timing > 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.

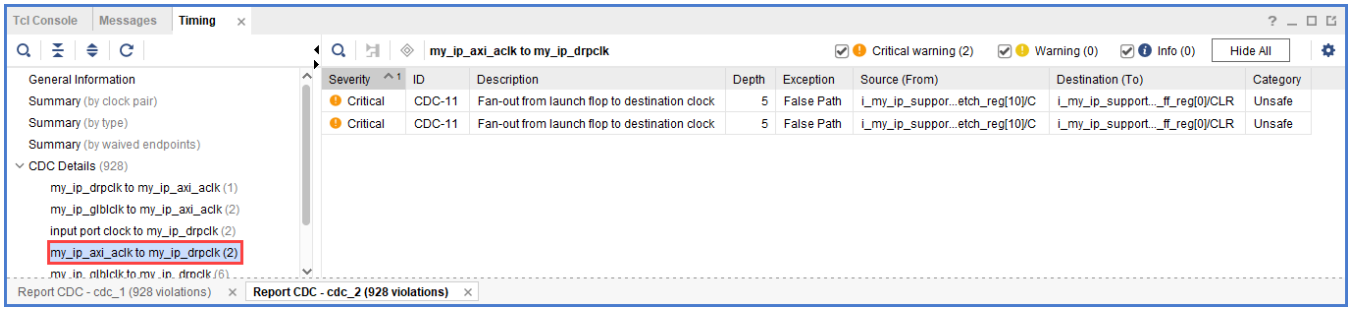


Figure 17: CDC Report with Multiple CDC Violations

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

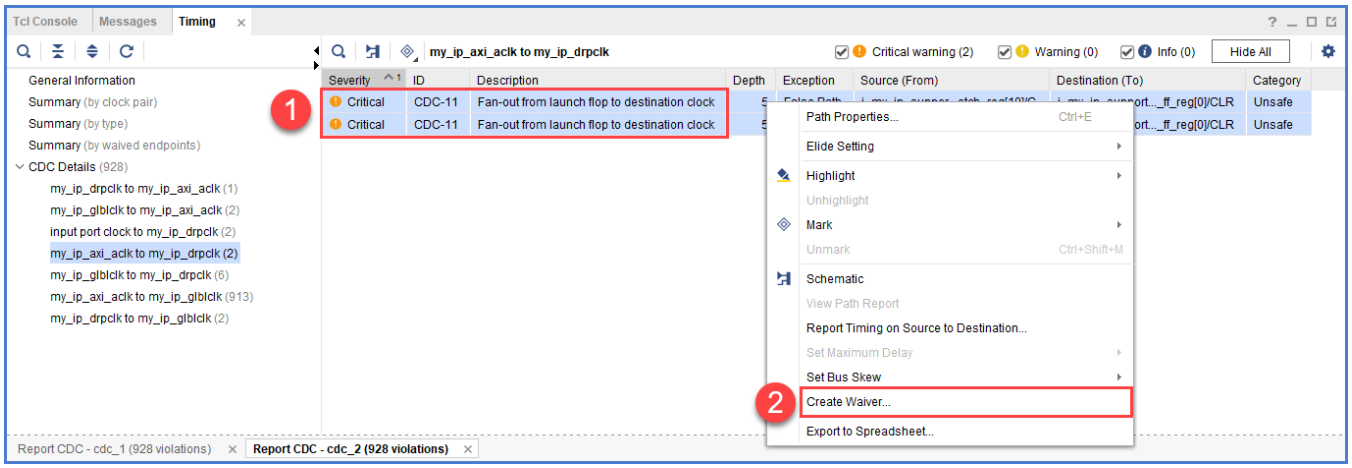


Figure 18: CDC Report with Create Waiver Command

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

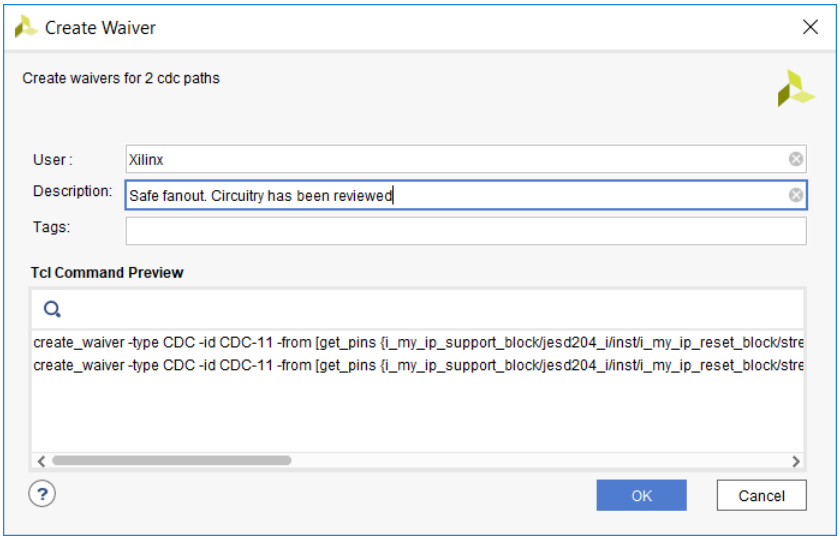
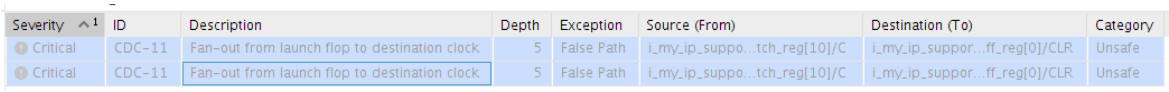


Figure 19: Create Waiver Dialog Box for CDC-11 Violations

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	ID	Description	Depth	Exception	Source (From)	Destination (To)	Category
Critical	CDC-11	Fan-out from launch flop to destination clock	5	False Path	i_my_ip_suppo...tch_reg[10]/C	i_my_ip_suppor...ff_reg[0]/CLR	Unsafe
Critical	CDC-11	Fan-out from launch flop to destination clock	5	False Path	i_my_ip_suppo...tch_reg[10]/C	i_my_ip_suppor...ff_reg[0]/CLR	Unsafe

Figure 20: Disabled CDC-11 Waivers

3. Select **Reports > Timing > Report CDC** to rerun Report CDC. In the Report CDC dialog box (Figure 1), 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.

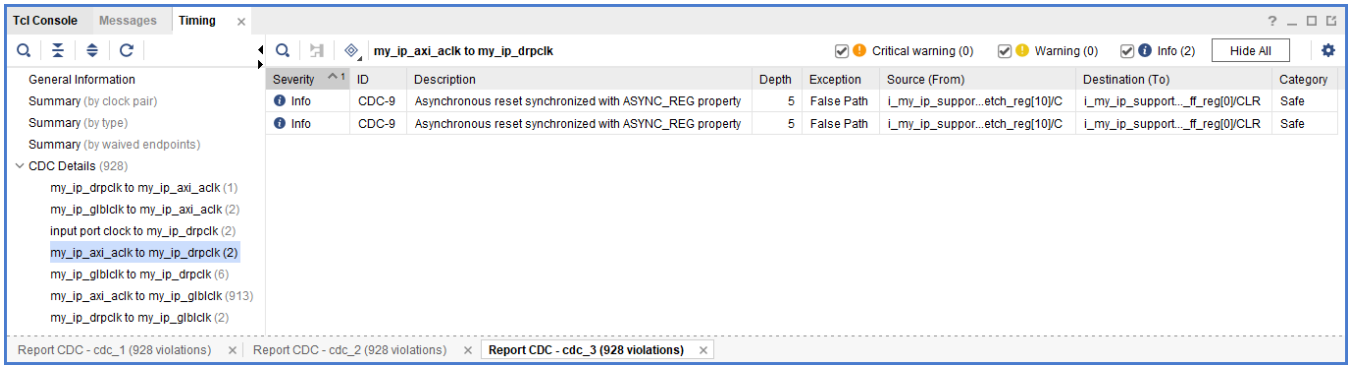


Figure 21: Detailed Report with CDC-9 Violations

5. To view a schematic of the violation, select the **CDC-9** row in the CDC Report, and click the **Schematic** toolbar button .

6. Verify that there is a 5-level synchronizer on the destination clock domain.

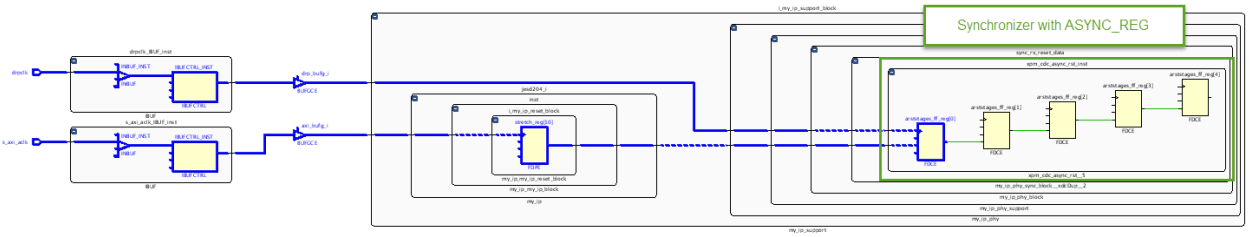


Figure 22: Schematic with 5-Level Synchronizer

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.

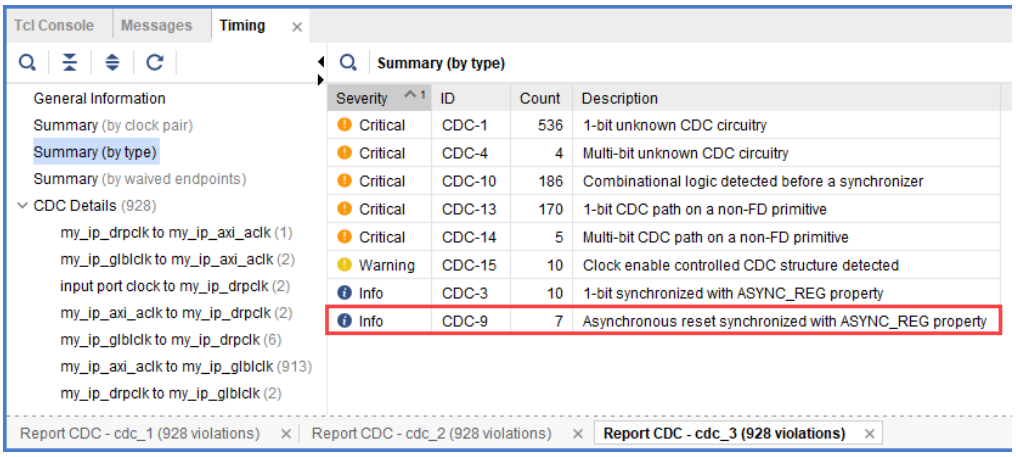


Figure 23: CDC Report with CDC-9 Violations

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.

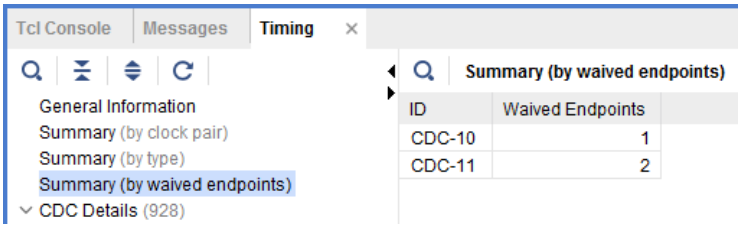


Figure 24: Summary by Waived Endpoints

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.

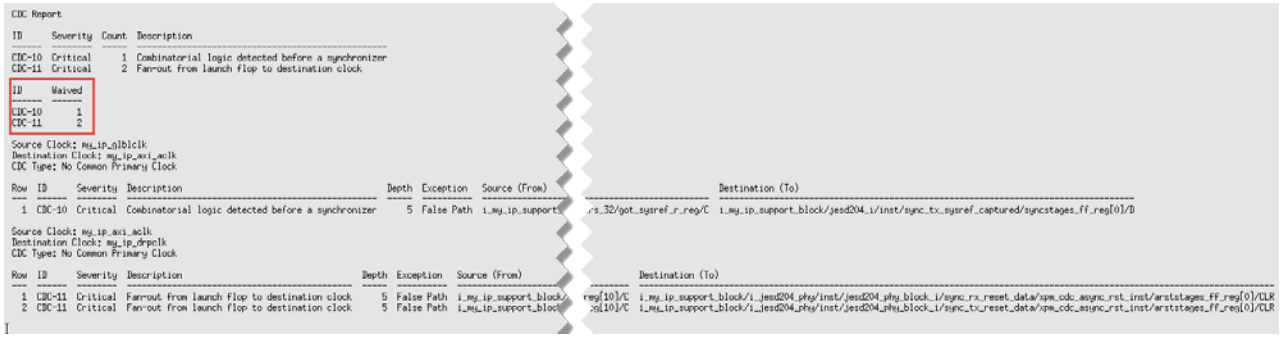


Figure 25: Text Report with Waived Violations

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` *or* `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_in
st/arststages_ff_reg[0]/CLR}] \
  -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_tx_reset_data/xpm_cdc_async_rst_in
st/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 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.*

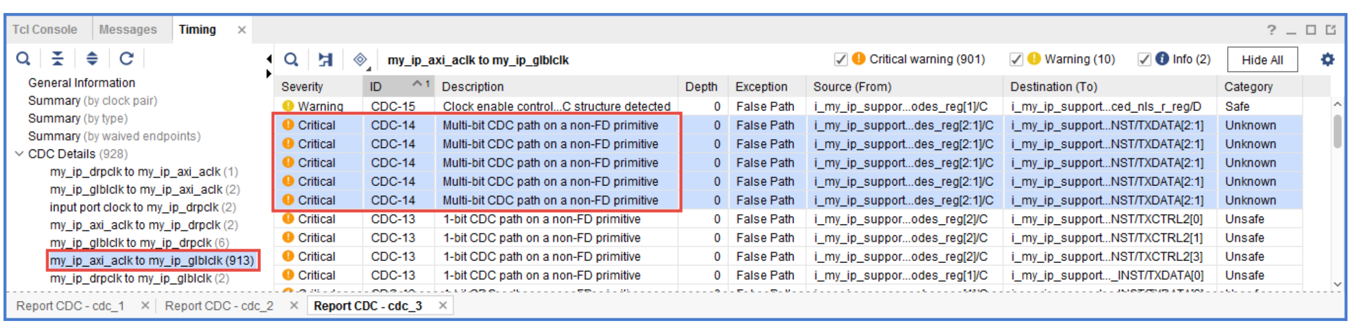


Figure 26: CDC Report with CDC-14 Violations

2. 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 > Timing > 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.

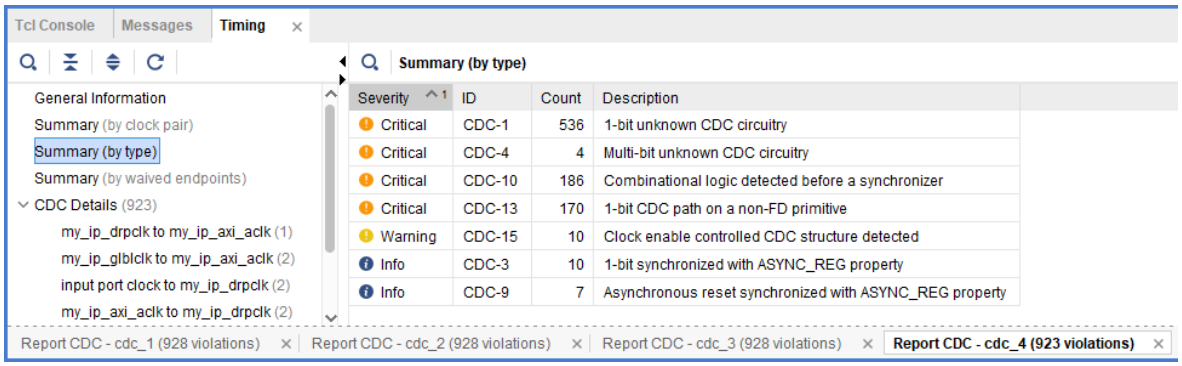


Figure 27: CDC Report Summary without CDC-14 Violations

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.

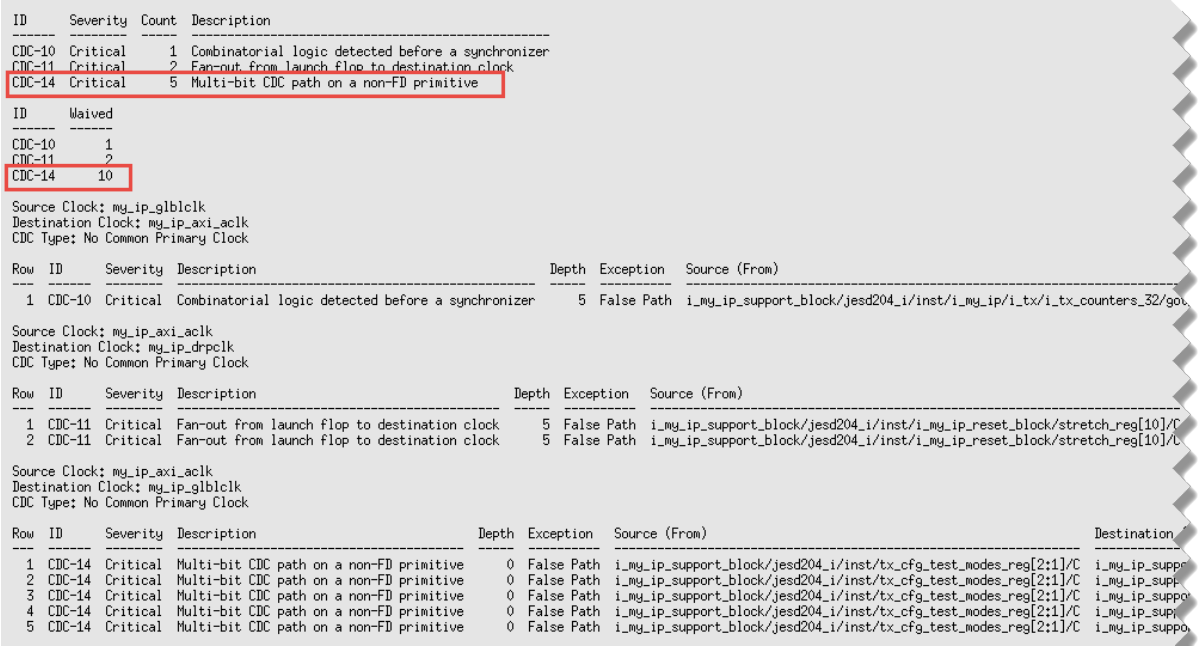


Figure 28: Text Report with Waived CDC-14 Violations

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.

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

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

Figure 29: write_waivers.xdc File

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 > Report DRC**.
2. In the Report DRC dialog box, leave all settings at their default, and click **OK**.

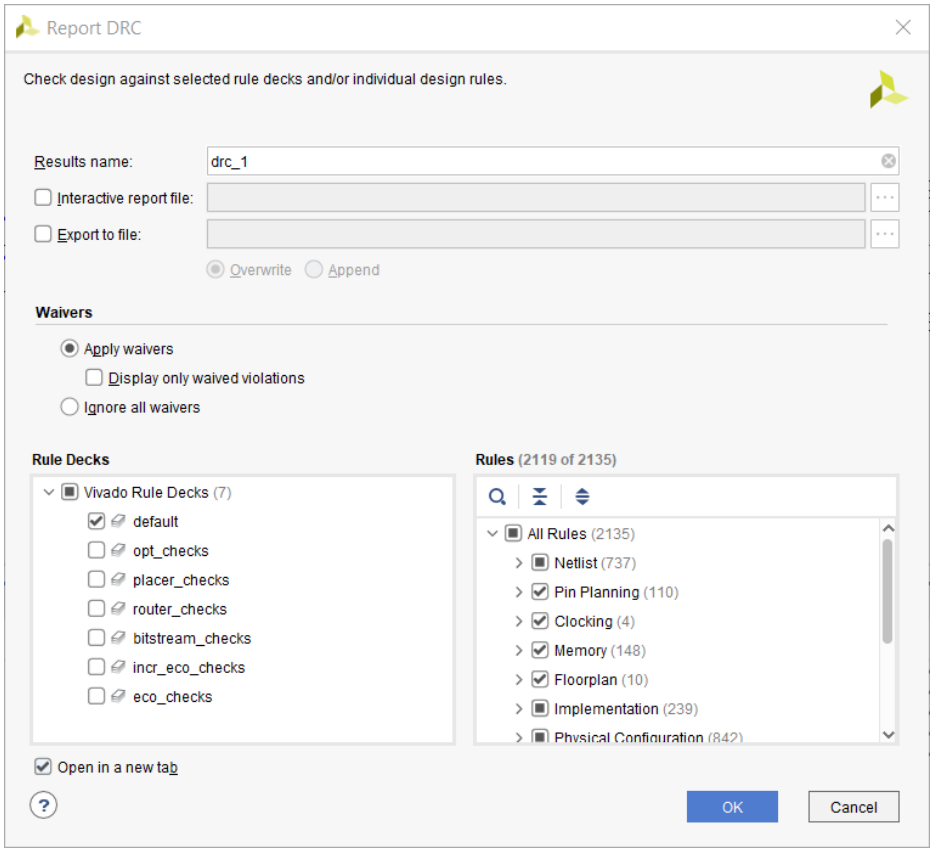


Figure 30: Report DRC Dialog Box

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.

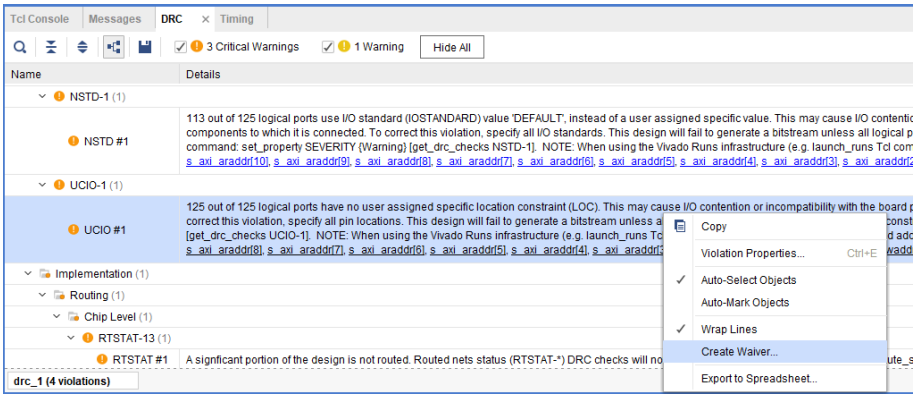


Figure 31: DRC Report with Create Waiver Command

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

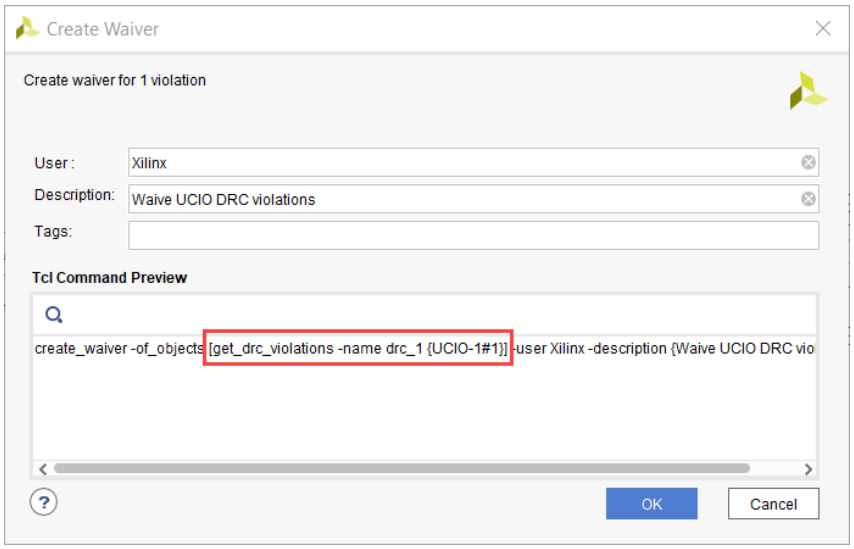


Figure 32: Create DRC Waiver Dialog Box

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 { refclk0p glblclkp refclk0n 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_bvalid 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] s_axi_aclk s_axi_aresetn s_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
```

Figure 33: Output of DRC Waivers

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 > Report DRC** to rerun Report DRC.
9. In the Report DRC dialog box, select **Display only waived violation** to report only waived violations, and click **OK**.

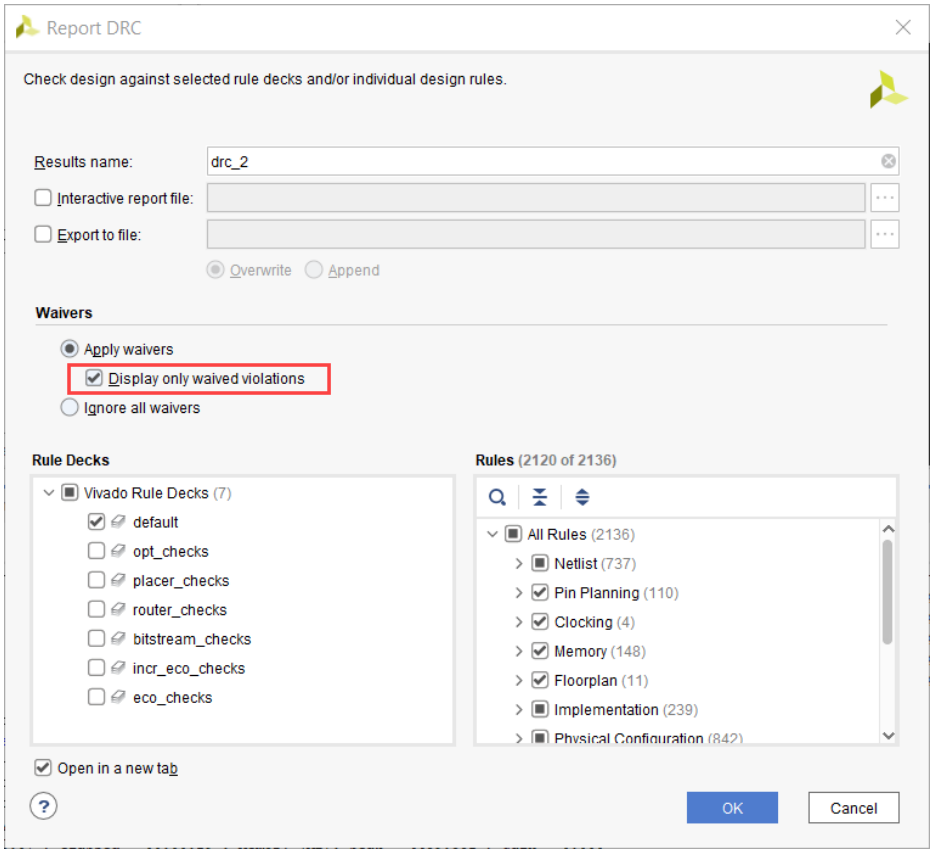


Figure 34: Report DRC Dialog Box with Display Only Waived Violations

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

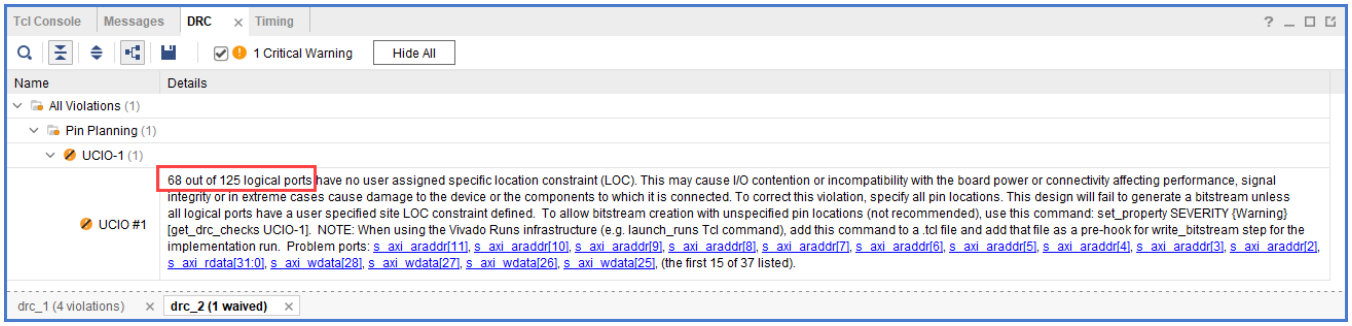


Figure 35: DRC Report with Waived Violations

IMPORTANT: *You cannot waive READONLY or NODISABLE violations. For example, if you enter* `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.

IMPORTANT: *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
METHODOLOGY  0           0               0            0             0
CDC          936         923             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 Vios  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 synchronizer   187         186             1            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

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.
```

Figure 36: Report Waivers Summary

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 plus 1 RTDAT-13 and 1 LOCE-1, 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.

```
------------------
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
METHODOLOGY  0           0               0            0             0
CDC          936         923             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 Vios  Remaining Vios  Waived Vios  Used Waivers  Set Waivers
----------  ---------------  ---------------------------------------------------------------------------  ----------  --------------  -----------  ------------  -----------
UCIO-1*     Critical Warning  Unconstrained Logical Port                                                  125         57              68           1             1
LOCE-1*     Warning          Pblock ranges contradict LOC constraints on logic assigned to the Pblock    1           1               0            0             0
NSTD-1*     Critical Warning  Unspecified I/O Standard                                                     113         113             0            0             0
RTSTAT-13*  Critical Warning  Insufficient Routing                                                         1           1               0            0             0


--------------------------------
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 synchronizer   187         186             1            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-1   Critical  1-bit unknown CDC circuitry                           536         536             0            0             0
CDC-3   Info      1-bit synchronized with ASYNC_REG property           10          10              0            0             0
CDC-4   Critical  Multi-bit unknown CDC circuitry                      4           4               0            0             0
CDC-9   Info      Asynchronous reset synchronized with ASYNC_REG property  7           7               0            0             0
CDC-13  Critical  1-bit CDC path on a non-FD primitive                 170         170             0            0             0
CDC-15  Warning   Clock enable controlled CDC structure detected       10          10              0            0             0

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.
```

Figure 37: Waiver Report Including Non-Waived Violations

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 SUMMARY
----------------
Waiver Type  Total Vios  Remaining Vios  Waived Vios  Used Waivers  Set Waivers
-----------  ----------  --------------  -----------  ------------  -----------
DRC          240         172             68           1             1
METHODOLOGY  125         125             0            0             0
CDC          936         923             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 Vios  Remaining Vios  Waived Vios  Used Waivers  Set Waivers
----------  ---------------  ---------------------------------------------------------------------------  ----------  --------------  -----------  ------------  -----------
UCIO-1#     Critical Warning Unconstrained Logical Port                                                   125         57              68           1             1
LOCE-1#     Warning          Pblock ranges contradict LOC constraints on logic assigned to the Pblock     1           1               0            0             0
NSTD-1#     Critical Warning Unspecified I/O Standard                                                     113         113             0            0             0
RTSTAT-13#  Critical Warning Insufficient Routing                                                         1           1               0            0             0


-------------------------------
3. REPORT DETAILS (METHODOLOGY)
-------------------------------
Rule      Severity  Description                    Total Vios  Remaining Vios  Waived Vios  Used Waivers  Set Waivers
--------  --------  -----------------------------  ----------  --------------  -----------  ------------  -----------
LUTAR-1   Warning   LUT drives async reset alert   8           8               0            0             0
TIMING-9  Warning   Unknown CDC Logic              1           1               0            0             0
TIMING-10 Warning   Missing property on synchronizer 1         1               0            0             0
TIMING-18 Warning   Missing input or output delay  115         115             0            0             0
```

Figure 38: Detailed Waivers Report with Methodology Checks

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        Type    Read-only  Value
   CLASS           string  true       cdc_waiver
   DESCRIPTION     string  false      No more CDC-14!
   ID              string  true       CDC-14
   INDEX           string  true       1
   NAME            string  true       CDC-14#1
   OBJECT_COUNTS   string  true       elems:2
   SCOPE           string  true
   TAGS            string  false
   TIME            string  true       <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

Introduction

The `report_qor_suggestions` (RQS) command enables the Vivado® Design Suite tools to analyze a design and provide automated solutions for enhancing QoR. The command can be run on any open design at any stage after synthesis in the implementation flow. RQS 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:

- Pre-synthesis constraint files
- Pre-implementation Tcl or constraint files
- Text recommendations requiring action

For pre-synthesis constraint files to take effect, the flow must be restarted from synthesis. For pre-implementation files to take effect, the flow should be restarted from `opt_design` and the Tcl must be applied before that command. Text recommendations usually suggest a constraint or RTL edit that requires the full flow to be rerun. This lab is a step-by-step tutorial for using RQS.

Step 1: Understanding the Design

This lab uses a pre-built 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 pre-built design contains the following modules:

- **Clocking Module:** The main clocking circuit for the design resides in `clocking_module.vhd`. See Figure 39. For simplicity, there are no controls over `RST` (resets) or `LOCKED`; `RST` is tied to `GND` and `LOCKED` is left disconnected.

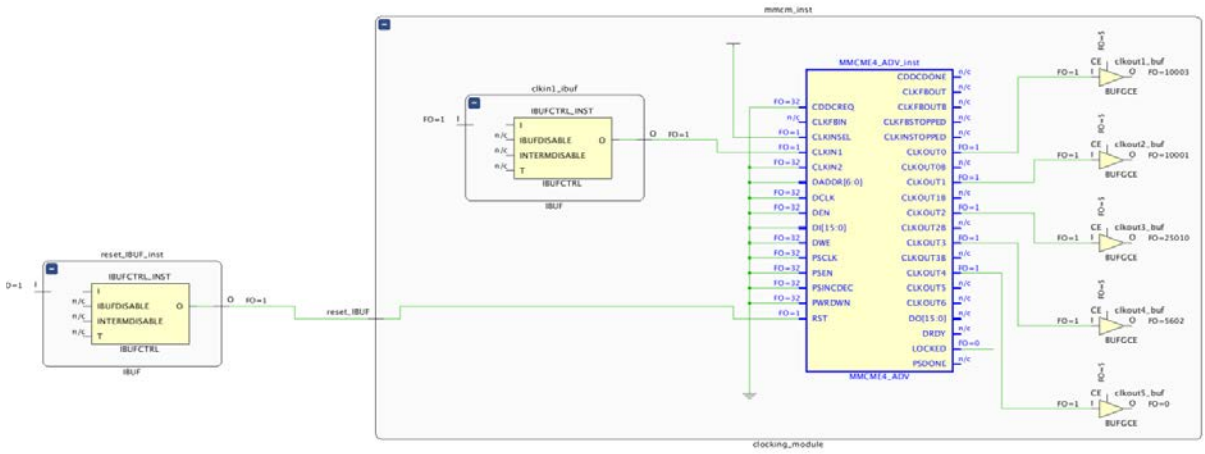


Figure 39: Clocking Module

- **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 reduce Total Hold Slack (THS).
- **Bus Double Register:** This module instantiates two lower-level modules that are connected by a wide data bus. The modules can be placed apart from each other using floorplanning. This has the effect of adding routing to the area between the two modules. The desired outcome is to create enough routing congestion to trigger suggestions.
- **LUT Combiner Module:** This module registers the input data three times. A circuit then takes the XOR of these bits and registers the output to create LUT combining, which can cause congestion.
- **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 expansion and contraction logic creates many logic levels and should be untimed. Being untimed, they are ignored by `report_qor_suggestions`.

The following steps cover opening the project and examining the placement of the floorplanned modules.

1. In the Vivado® Design Suite, go to **File -> Project -> Open** and select the project located in `<extract_Dir>/lab2/1_InitialRun`.

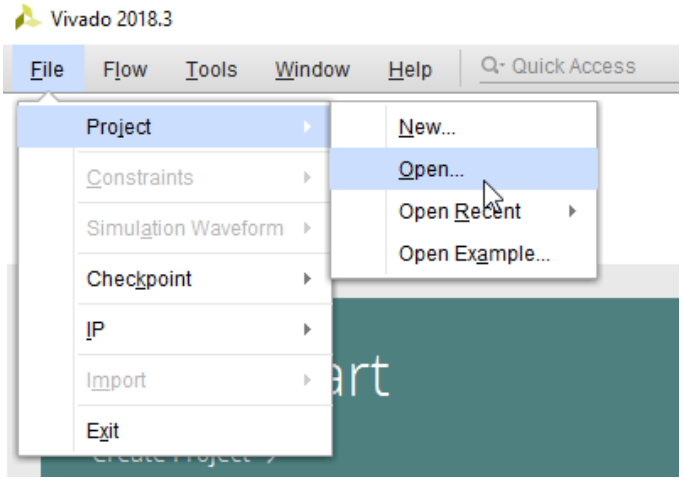


Figure 40: Opening the Project

2. In the Flow Navigator, click **Open Synthesized Design**.
3. In the **Netlist** view, look at the hierarchy.

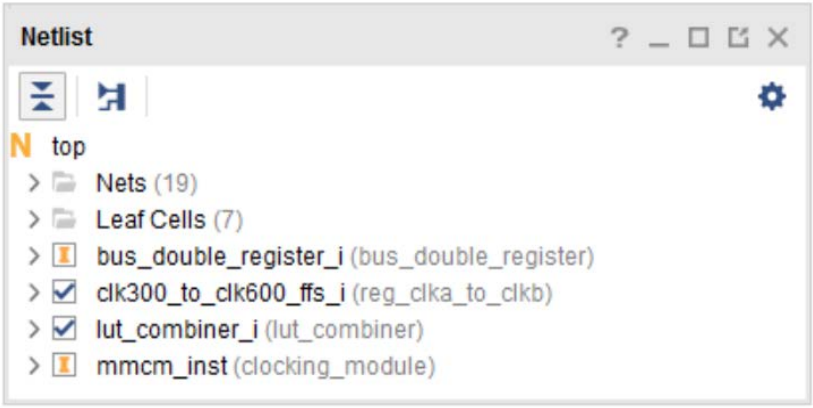


Figure 41: Design Hierarchy

All modules at the top level are independent of each other. There are no timing paths between them. There are also no timing paths between these modules and the I/Os.

4. In **Device** view, look at the four pblocks. These have been added to help trigger suggestions on the design without requiring a highly utilized design.

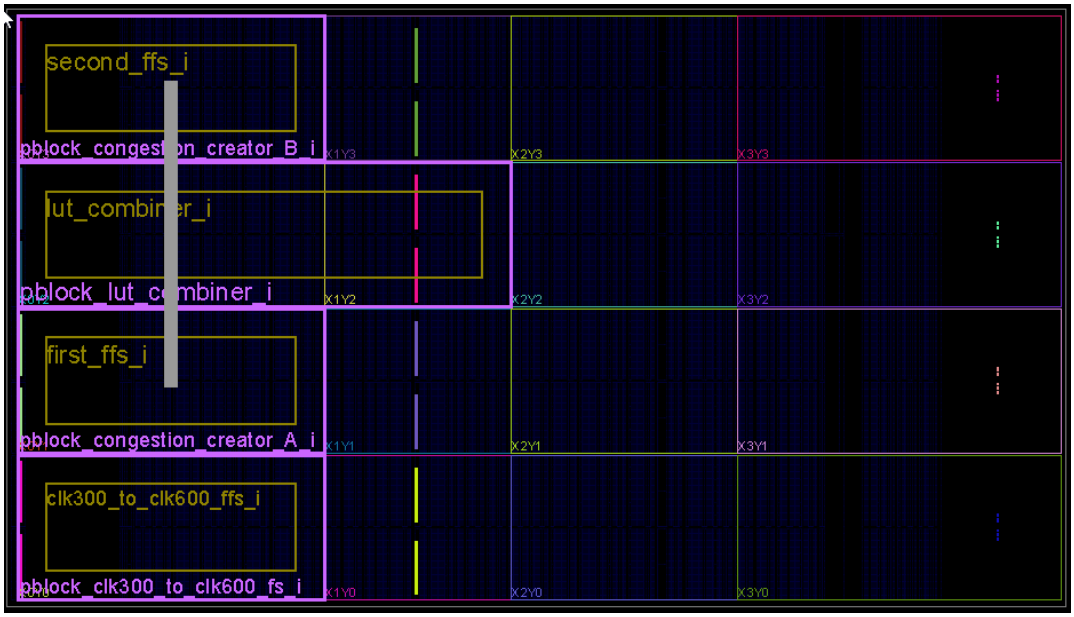


Figure 42: Initial Run Device View

TIP: When a block or blocks are selected, you can investigate the design further by pressing F4 to open the schematic tools.

As shown in Figure 42, the blocks have been placed so that there is a LUT-combined module in between the two FF modules. This placement forces nets across the LUT-combined module to create routing congestion; the effect mimics a real-world design, but is rigid. RQS is tuned to real-world designs.

Step 2: Running Report QoR Suggestions

This step covers running the `report_qor_suggestions` 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 `synth_design`, `link_design`, `opt_design`, `place_design`, `phys_opt_design`, or `route_design`.

1. In the Vivado® IDE, under the **Reports** pull-down menu, select **Report QoR Suggestions...** to bring up the dialog box shown in Figure 43.

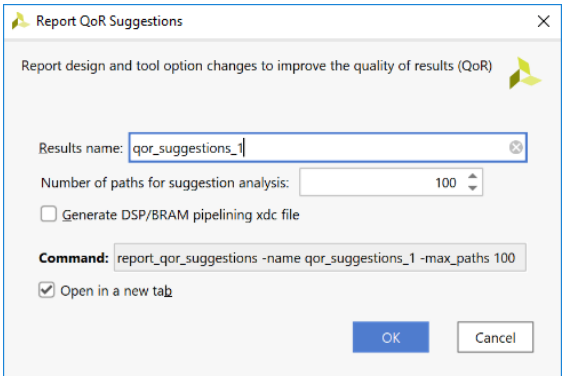


Figure 43: Report QoR Suggestions Dialog Box

2. Do not change any options. Click **OK**. The equivalent Tcl command is `report_qor_suggestions -name qor_suggestions_1`. The report opens automatically in the integrated design environment (IDE).

   Note: By default, the RQS command reports on the 100 worst paths. You can change the number of paths that RQS uses for the analysis of timing-critical paths. Increasing this number generates more suggestions, but on paths that are reducing in criticality. You can override this default setting and expand the number of paths examined by RQS using the `-max_paths` *switch.)*

   Note: It is also possible to generate suggestions that enable what-if analysis for DSP and block RAMs. This option adds pipelines to these blocks and allows you to test the impact of this; however, the circuit will not be logically equivalent.

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 details are provided.

1. In the generated report, click on **RQS Summary**. This brings up the report section show in Figure 44.

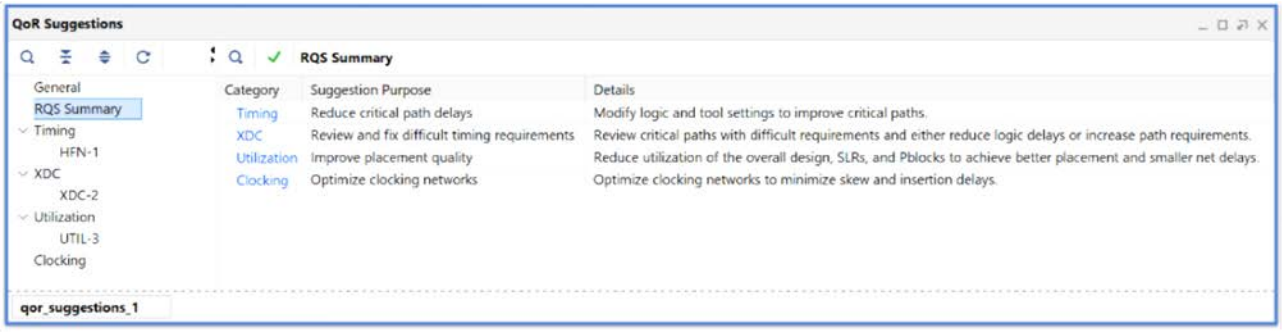


Figure 44: RQS Summary

The summary section gives an overview of which categories are impacted. The order of categories returned is dictated by the criticality of this issues found, with the most important items at the top. You can navigate to detailed suggestions for each of the categories listed.

2. Click on the **Timing** section. The report is shown in Figure 45:

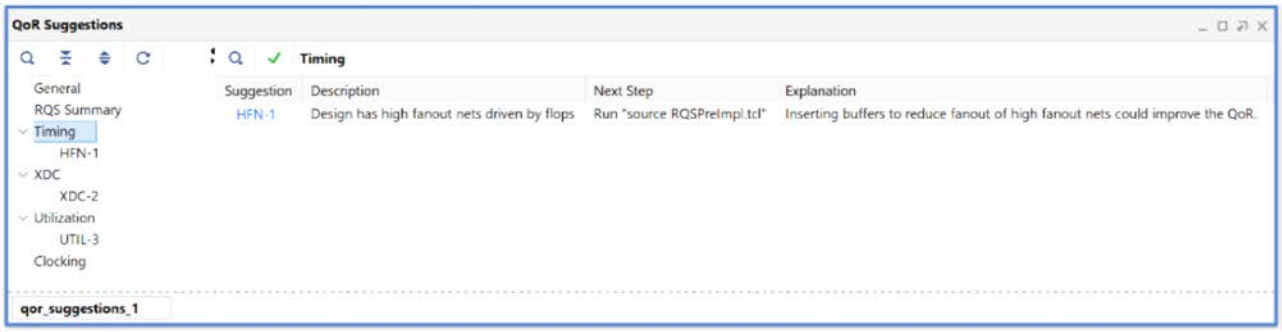


Figure 45: Timing Section

The timing section has two messages associated with it. At this level, you can see a description of the report, the next step to take, and an explanation of why the suggestion has been triggered. To get more details on the specific path that has triggered this suggestion, click on **HFN-1**. This is shown in Figure 46.

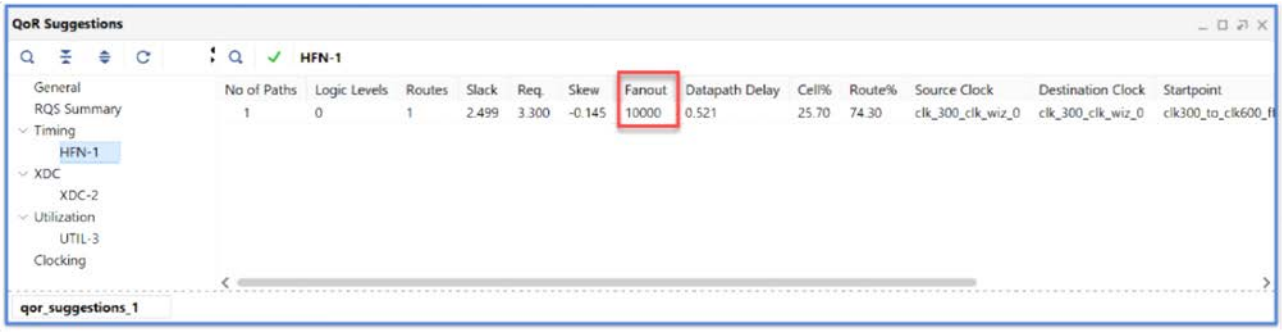


Figure 46: HFN-1 Report Section

It can be seen from the details section that the fanout = 10000. This is the primary trigger, however, and there are other criteria. This suggestion only triggers at the right target frequency, and if there are BUFG resources available.

3. Next, click on **XDC**. This part of the report is shown in Figure 47:

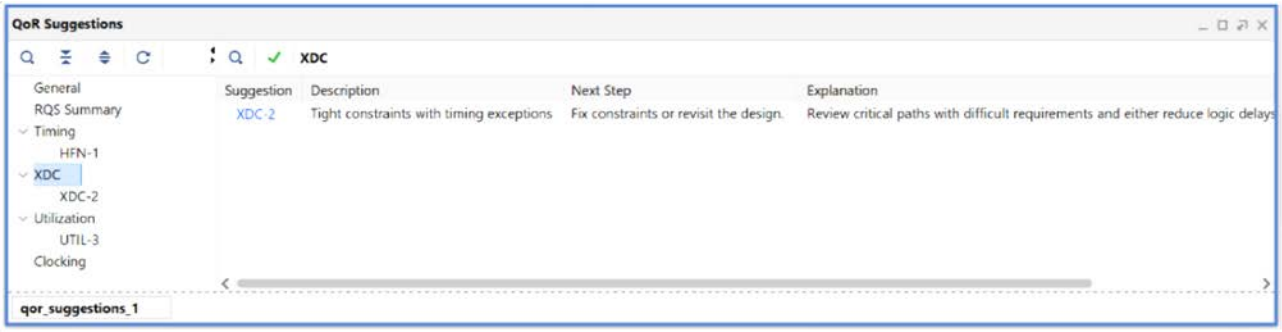


Figure 47: XDC Summary

You can see that RQS has identified a path with tight constraints. Next, navigate to **XDC-2** to find more details on this path, as shown in Figure 48. Here, a 600 MHz path with seven levels of logic is listed.

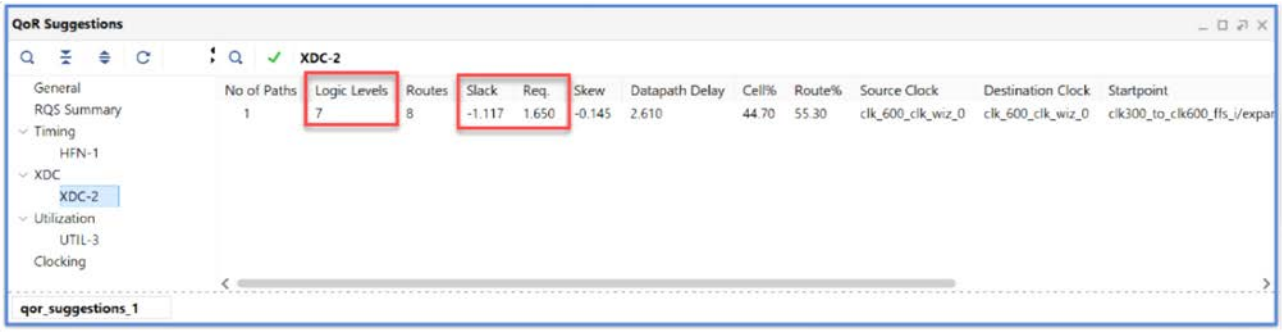


Figure 48: XDC-2 Details

Intervention is required to fix this path; it is not possible to fix every path automatically. For this tutorial, assume that a false path constraint has been missed. Enter the following in the Tcl console to add this:

```
set_false_path -to [get_cells clk300_to_clk600_ffs_i/bit_reducer_i/tmp_reg]
```

4. Click on **Utilization**. Here, there is a single message that reports high utilization of certain cell types within certain pblocks.

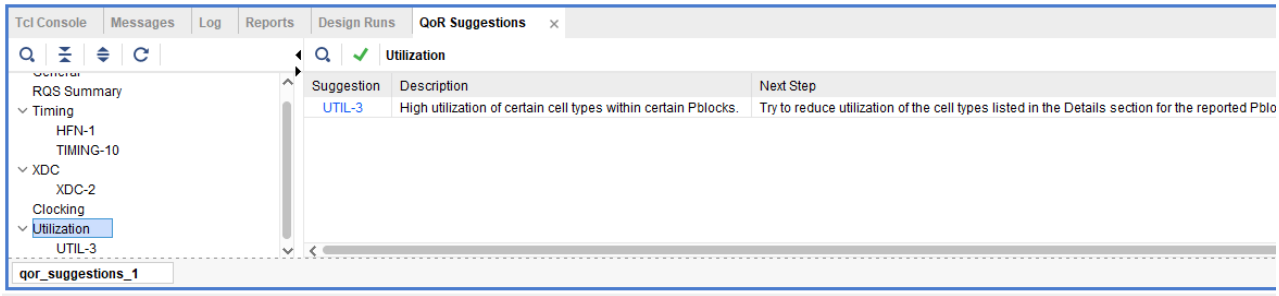


Figure 49: Utilization Summary

Click on **UTIL-3** to view the details section for this message.

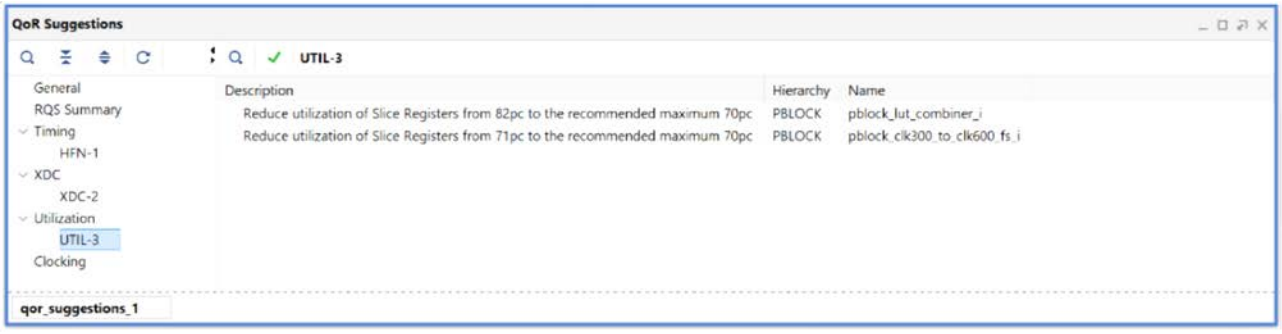


Figure 50: Detailed Utilization Report

You can see that there are two overutilized pblocks. RQS provides a general recommendation to reduce the register utilization of these pblocks to 70%. Using `opt_design` might also reduce the

utilization, but this is not certain. Increasing the size of the pblocks is most likely to improve the utilization statistic.

5. Click **Window -> Physical Constraints**. In the Physical Constraints window, select the `clk300_to_clk600_ffs_i` pblock and view the Statistics tab in the Pblock Properties window to view the utilization of the pblock. You can see that the CLB register utilization is 71.85%.

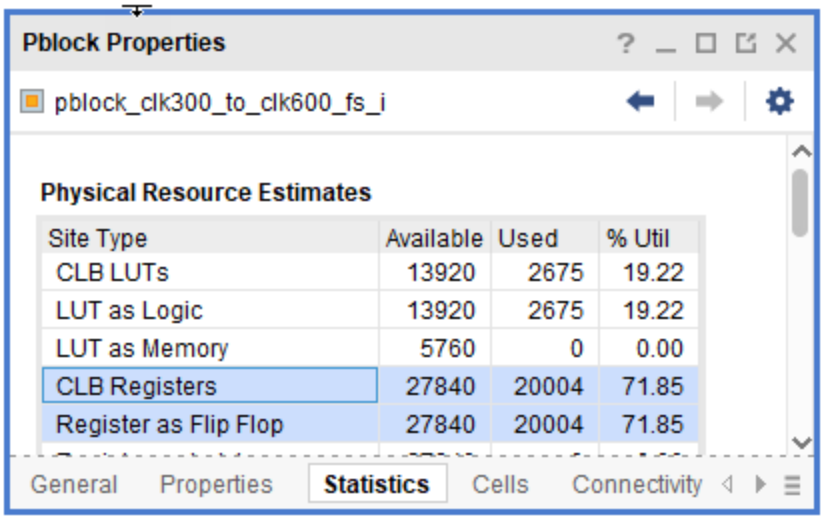


Figure 51: Pblock Utilization Statistics

6. To resize the pblock, add the following command to the Tcl console:

```
resize_pblock pblock_clk300_to_clk600_fs_i -add {SLICE_X0Y0:SLICE_X34Y59} -
remove {CLOCKREGION_X0Y0:CLOCKREGION_X0Y0} -locs keep_all
```

   The updated pblock utilization is now 59.54%. This is lower than the threshold. Leave the other pblock as-is. Here, constraints on pblocks are being used to generate routing congestion. In typical designs, pblock utilization at 82% would cause a failure.

7. Click on **Clocking**. In Figure 52, you can see two messages that are primarily triggered due to high total hold slack (THS).

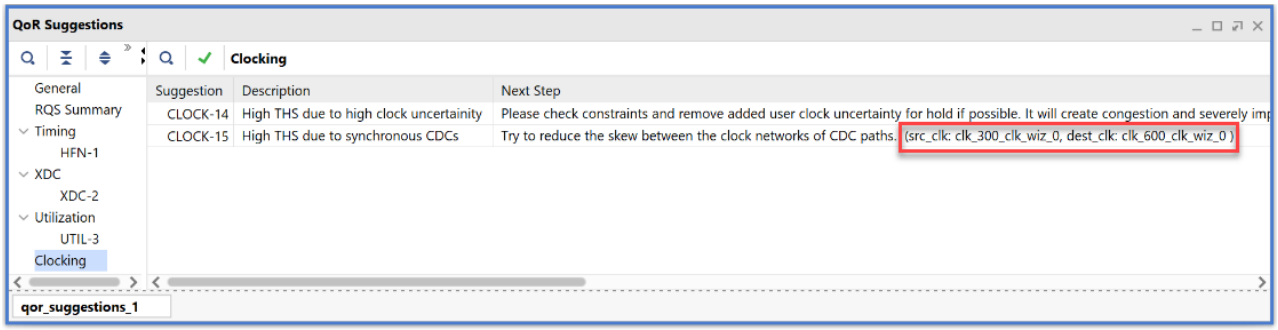


Figure 52: Clocking Summary

Both messages are triggered due to high hold time. There is no detailed table for these messages. It is possible to identify the source and destination clocks from the message. Cross-probing is only available from the detailed view. To generate a timing report, enter the following in the Tcl console:

```
report_timing -from [get_clocks clk_300_clk_wiz_0] -to [get_clocks
clk_600_clk_wiz_0] -name timing_paths -delay_type min
```

The CLOCK-14 message is a general message about what is causing the hold time error. In this case, it is primarily due to clock uncertainty. Clock uncertainty can be caused by user constraints or the netlist make up.

8. In the previously generated timing report**,** click on the **Clock Uncertainty** link to see the makeup of the clock uncertainty. As shown in Figure 53, phase error makes up a significant portion of the issue:

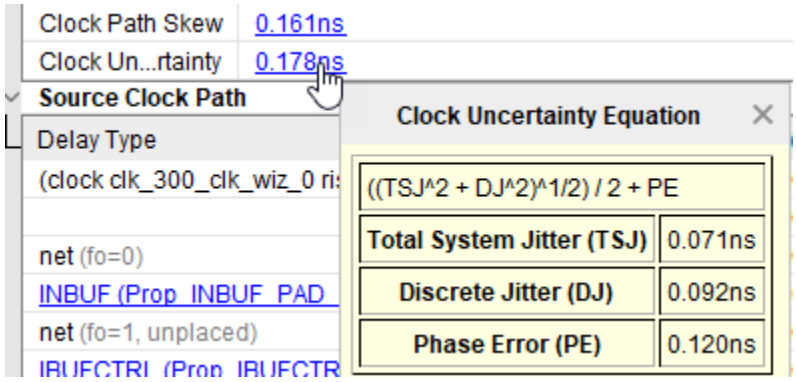


Figure 53: Clock Uncertainty

The CLOCK-15 message is concerned with skew between related clocks. One way to fix this is to use `BUFGCE_DIV` buffers and a common MMCM output. In the next run, the clocking structure is changed to reflect this, thereby reducing the THS.

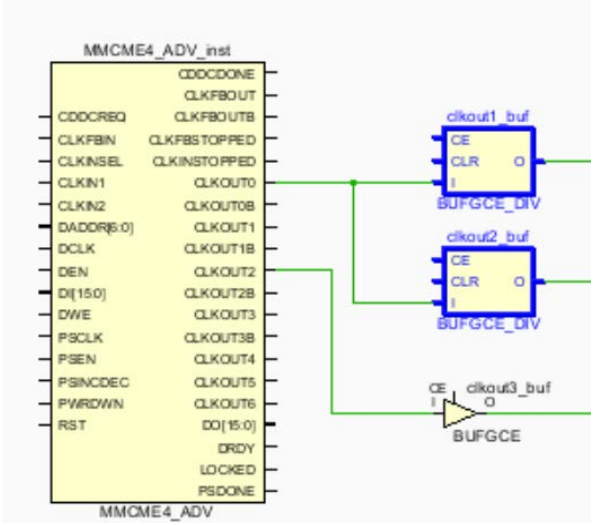


Figure 54: Updated Clocking Using BUFG_DIV Buffers

9. Click on the icon. This opens the Apply Suggestions dialog box shown in Figure 55. Apply Suggestions is used generate the automatic constraints that can be applied to the design.

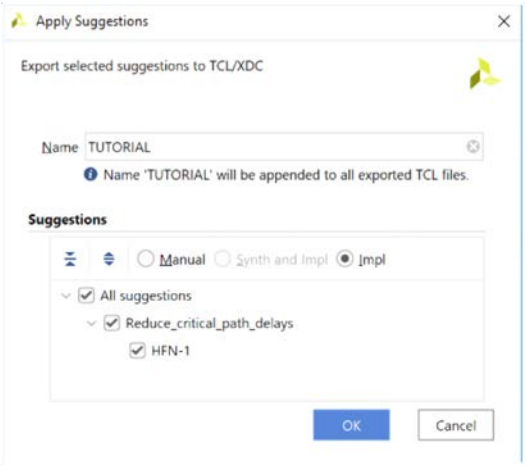


Figure 55: Apply Suggestions Dialog Box

From this dialog box, you can select names to uniquify the output files. You can also generate the suggestions based on the flow you plan to use. The radio button allows you to select solutions for the synthesis and implementation flow, or only the implementation flow. In this case, there are no suggestions to run at synthesis, so you can proceed with only the implementation flow.

Note: Suggestions in RQS can overlap; do not run suggestions at the implementation stage that will be modified by a new synthesis run. If you are early in the design stage, run with synthesis and implementation suggestions. Later in the design flow, it can be beneficial to target implementation-only suggestions.

Change the Name to `TUTORIAL`. Select **Impl** and ensure that the **All suggestions** box is checked. Click **OK**. The files will be written to the present working directory identified using the `pwd` command in the Tcl console.

10. Navigate to the output directory to view the generated file.

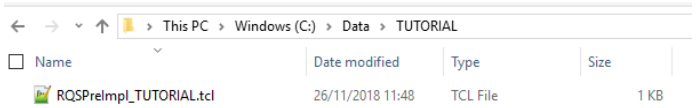


Figure 56: Output File from RQS

Open the `RQSPreImpl_TUTORIAL.tcl` file and examine its contents. This file inserts a `BUFG` buffer on the high-fanout net `a_r` using the `CLOCK_BUFFER_TYPE` property, as shown in the following example:

```
catch { set_property clock_buffer_type BUFGCE [get_nets
{clk300_to_clk600_ffs_i/a_r}] }
```

For this change to take effect, it must be included in the run before the `opt_design` command is run. This helps with meeting timing on this net and improves congestion issues— although it is too early to report on congestion at the post-synthesis stage.

11. In the **Flow Navigator**, click **Settings** -> **Implementation**. Under Opt Design, select the `RQSPreImpl_TUTORIAL.tcl` file for the **tcl.pre** option, and click **OK**.

TIP: *Only one Tcl file can be selected under Opt Design. If RQS outputs two files, you should also make use of the tcl.post switch under Design Initialization.*

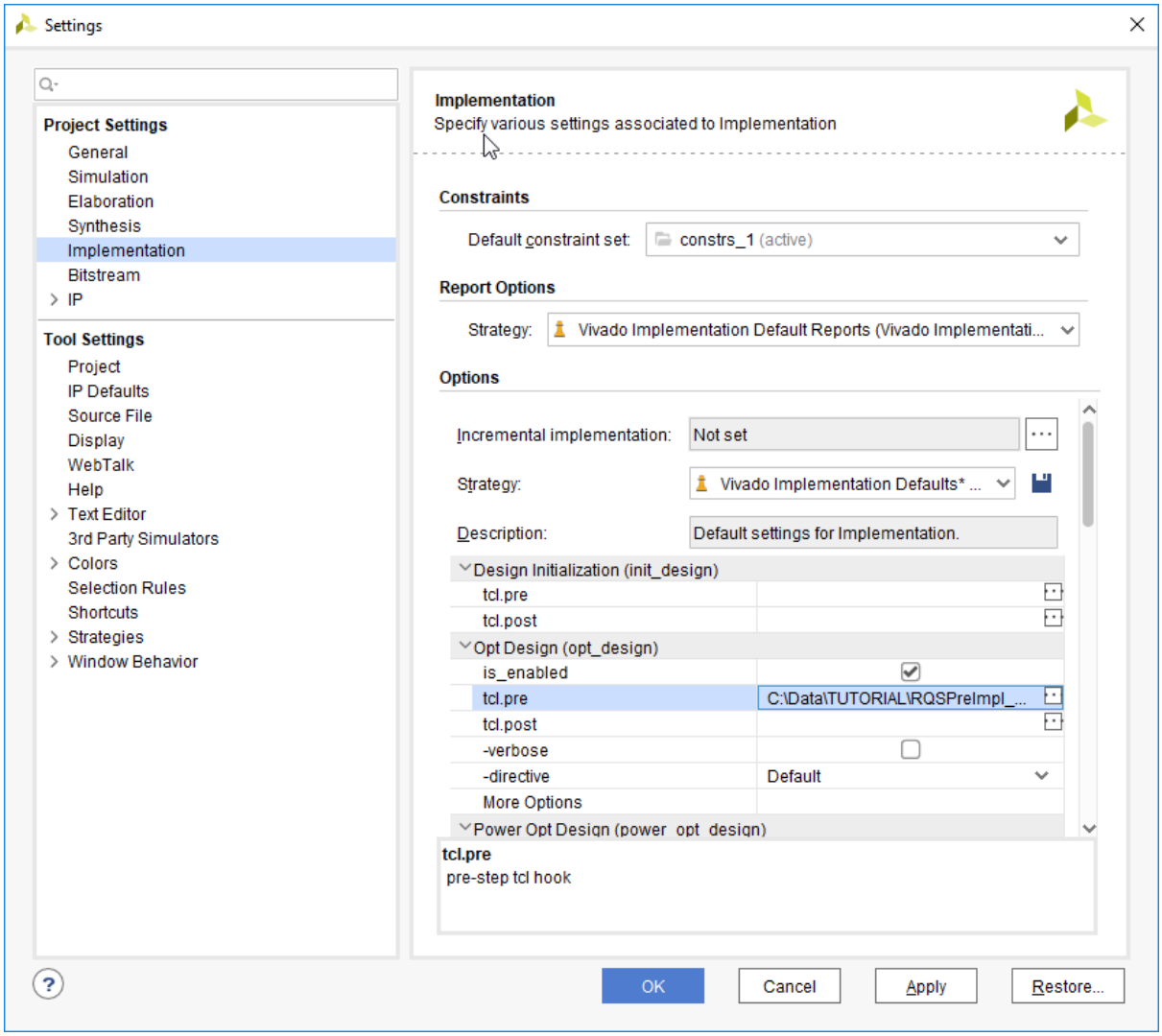


Figure 57: Pre-Step Tcl Hook Pointing at Generated Tcl Scripts

In summary, at the post-synthesis stage, you have accomplished the following:

- Added a false path on critical timing paths
- Identified a clocking change to `BUFGCE_DIV` buffers
- Increased the pblock size to more realistic utilization levels
- Added a constraint that automatically inserts a `BUFG` at `opt_design` to improve timing

By completing these key changes early in the timing closure process, you have cleaned the design of several issues and saved some design cycles. However, there is no congestion analysis at synthesis; implementation must be run to see this.

Step 4: Run with Suggestions

This step covers the run with the suggestions from the previous run now applied.

1. In the Vivado Design Suite, go to **File -> Project -> Open** and select the project located in `<extract_dir>/lab2/2_synth_fixes`. It is OK to close the existing project.
2. In the Flow Navigator, select **Open Implemented Design**.
3. To generate suggestions, you will now artificially create timing failures by tightening the clock constraint. In the Tcl console, enter the following:

   ```
   create_clock -name force_timing_issue_clk -period 5.0 [get_pins
   mmcm_inst/MMCME4_ADV_inst/CLKOUT2]
   ```

   Note: This is not recommended practice.
4. From the Report menu, run **Report QoR Suggestions...**.The summary now includes a congestion report. This section appears when timing failures and congestion have been identified after placement or routing.

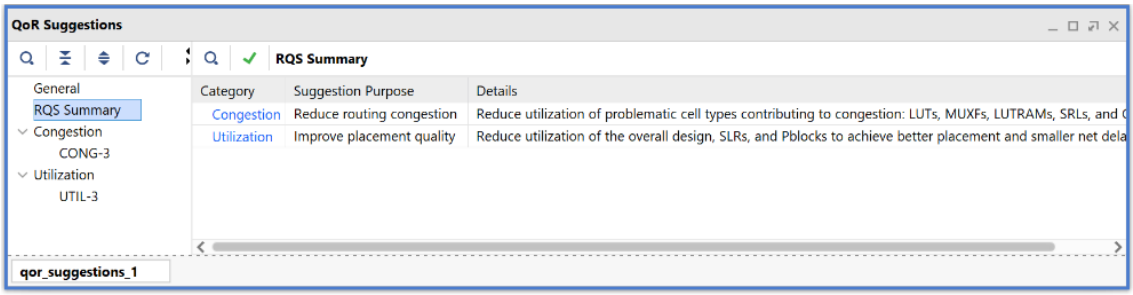


Figure 58: Post-Route Report QoR Suggestions Summary

The other section available in the summary is Utilization. A quick investigation of this shows that it is the same suggestion as previously identified; the utilization of `pblock_lut_combiner_i` is overutilized intentionally to generate congestion.

5. Click on **Congestion**. There are two suggestions; see Figure 59. The first, CONG-31, is a suggestion that typically helps congestion: running `opt_design –merge_equivalent_drivers` helps merge equivalent signals that might overlap after placement, hence saving routing.

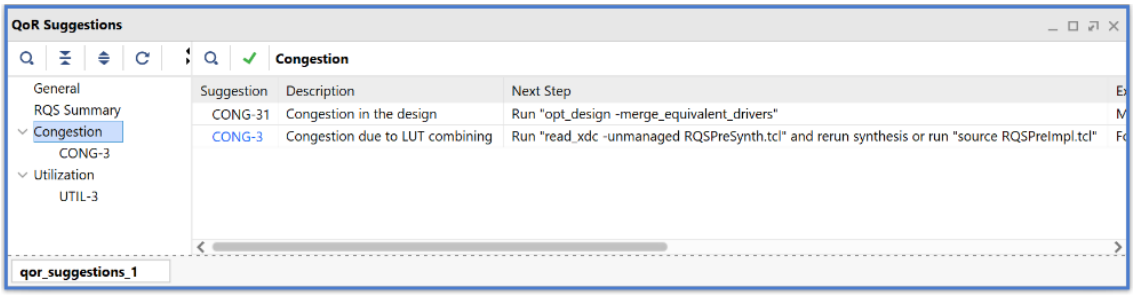


Figure 59: Post-Route Congestion Summary

To add this option to the next run, open **Implementation Options** and add this to the More Options section under Opt Design, as shown in the following figure.

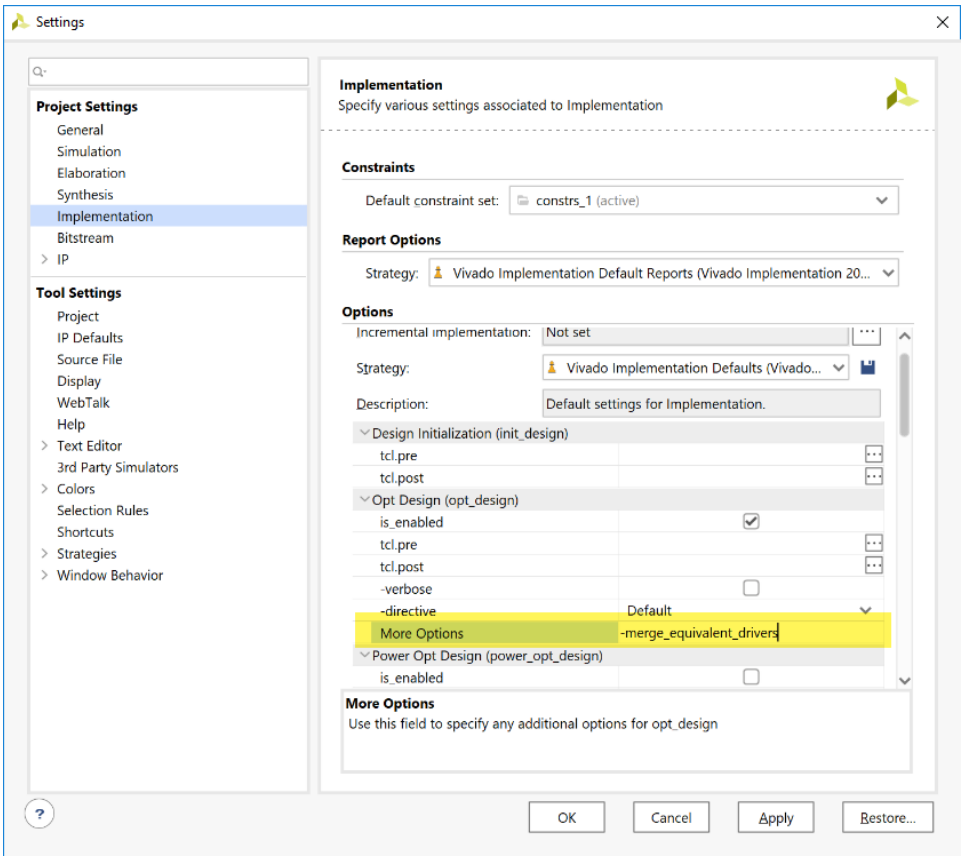


Figure 60: Implementation Options

6. In the congestion summary, click on **CONG-3**, which is a message about excessive LUT combining in the congested area.

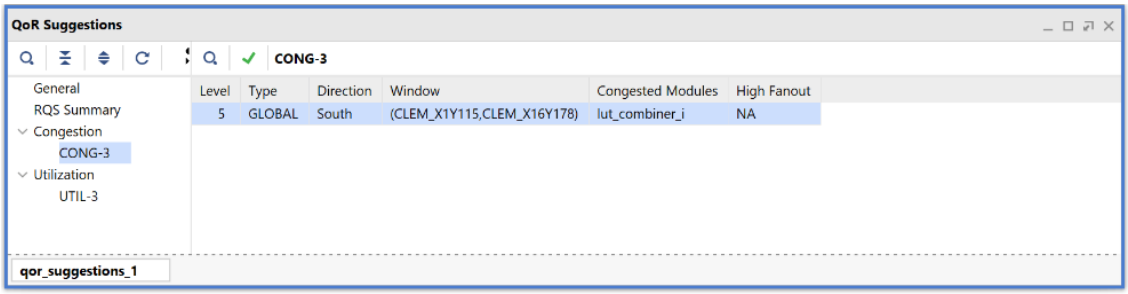


Figure 61: CONG-3 Details

Select **CONG-3** in the table so that the cells are highlighted in the device view. Right-click and select **Mark** to mark the cells that are to be targeted for undoing LUT combining.

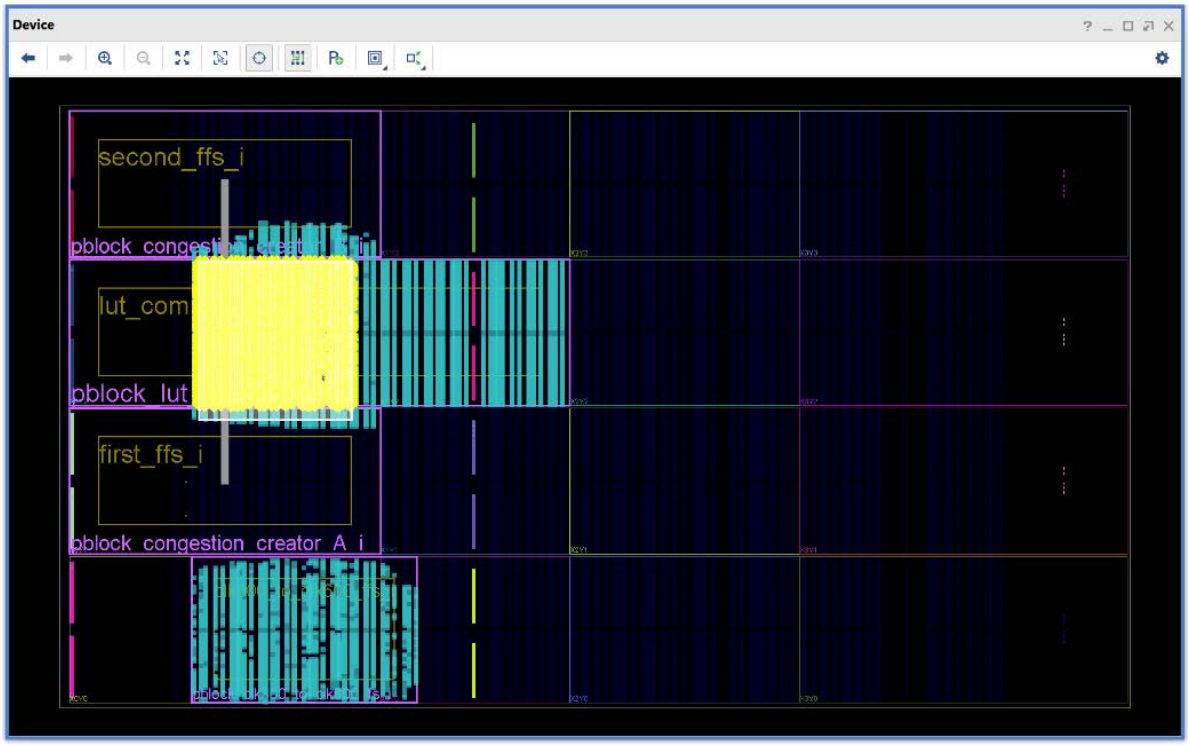


Figure 62: Marked Cells Targeted to Uncombine

7. Click on the **Apply Suggestions** button. This gives you two options to generate automatic constraints:

   - **Synth and Impl**: This generates congestion solutions based on the significant module within the congested area. The overall impact might be wider than the congested area.
   - **Impl**: This generates constraints for the specific cells highlighted.

   Enter `TUTORIAL_POST_ROUTE` in the Name field. Select **Impl** and ensure that all the suggestions are selected, as shown in Figure 63, and click **OK**.

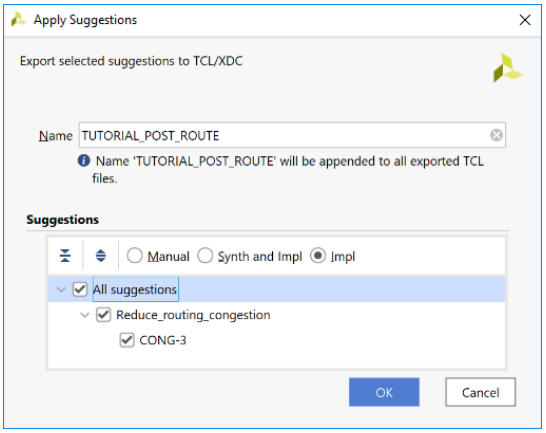


Figure 63: Post-Route Suggestions

8. You can now examine the files that are generated. The `PreImpl_TUTORIAL_POST_ROUTE.tcl` file contains the resetting of the `SOFT_LUTNM` properties on impacted individual cells. To add this file to the implementation run, add it to the post-`link_design` step as shown in Figure 64 (remember that a pre-`opt_design` Tcl file is already selected).

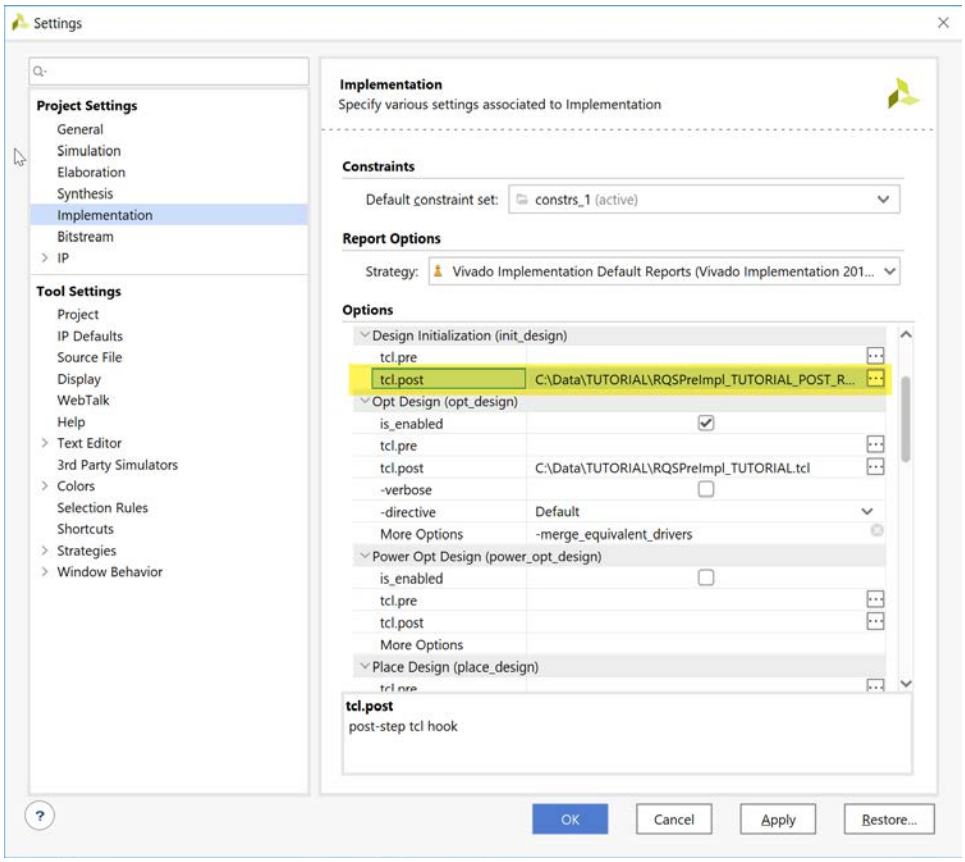


Figure 64: RQS Post-Route Script Added to the Implementation Flow

Note: If there are more than two files to be added to the flow, you need to merge the files into one. RQS does not offer suggestions that have previously been implemented again.

It is also possible to run the synthesis flow to remove LUT combining on this module using the `RQSPreSynth_TUTORIAL POST_ROUTE.tcl` file. Use this file instead of *`RQSPreImpl`*`_TUTORIAL POST_ROUTE.tcl`, because the suggestions overlap. It should be applied to the synthesis run as a pre-`synth_design` Tcl option.

9. Open the `RQSPreSynth_TUTORIAL POST_ROUTE.tcl` file. This file uses `BLOCK_SYNTH` properties to stop LUT combining on affected hierarchies so that the scope is not limited only to the congested area but it is much easier to maintain. It is recommended to use this option if the design is still undergoing many changes. Close the file.

   At this point, the implementation would normally be rerun. However, because this is a constructed design, it will fail implementation. This is because of the high level of LUT combining and pblock utilization used to generate the congestion. Typical designs have lower thresholds and are not impacted like this constructed design.

Summary

In this lab, you used RQS to conduct a complex analysis of a demonstration design. You examined the report and output files, seeing RQS offer both Tcl and text suggestions. RQS provided recommendations to solve implementation problems in the following areas:

- Utilization
- Timing
- Constraints
- Congestion

You saw how to apply the suggestions and at what stage in the flow to apply them, and how to make an informed choice between synthesis and implementation recommendations.

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.