AN 775: Generating Initial I/O Timing Data and I/O Element Delays for Intel FPGAs

Updated for Intel® Quartus® Prime Design Suite: **21.3**

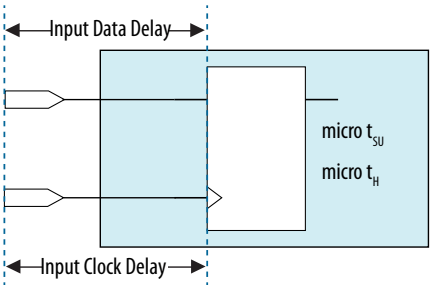
Contents

1. Generating Initial I/O Timing Data for Intel FPGAs

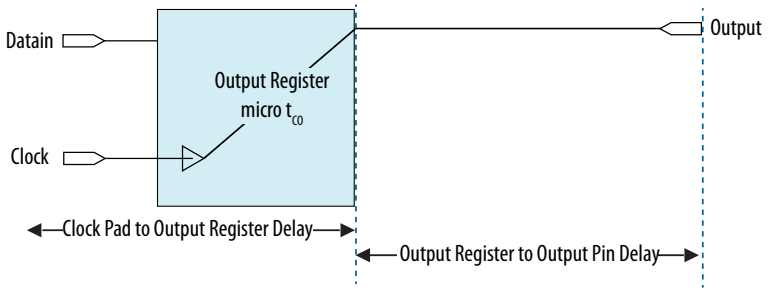
This document describes generating initial I/O timing data and I/O element delay information for Intel FPGA devices using the Intel® Quartus® Prime software GUI or Tcl commands.

Initial I/O timing data is useful for early pin planning and PCB design. You can generate initial timing data for the following relevant timing parameters to adjust the design timing budget when considering I/O standards and pin placement.

Table 1. I/O Timing Parameters

Timing Parameter	Description
Input setup time (t_{SU}) **Input hold time (t_H)**	**Figure 1. t_{SU} and t_H Timing Parameters**  t_{SU} = input pin to input register data delay + input register micro setup time - input pin to input register clock delay t_H = - input pin to input register data delay + input register micro hold time + input pin to input register clock delay

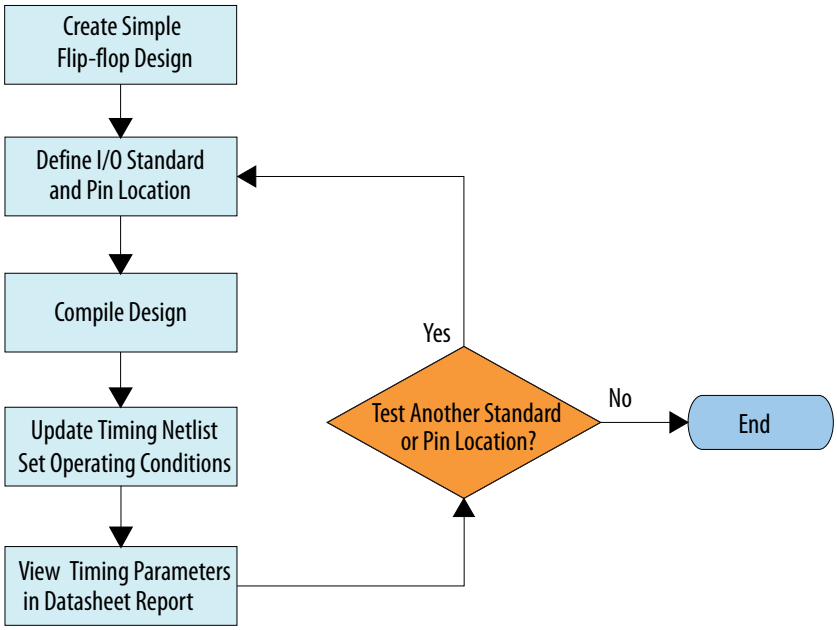
continued...

Timing Parameter	Description
Clock to output delay (t_{CO})	**Figure 2. t_{CO} Timing Parameters**  t_{CO} = + clock pad to output register delay + output register clock-to-output delay + output register to output pin delay

Generating initial I/O timing information includes the following steps:

- Step 1: Synthesize a Flip-flop for the Target Intel FPGA Device on page 5
- Step 2: Define I/O Standard and Pin Locations on page 6
- Step 3: Specify Device Operating Conditions on page 6
- Step 4: View I/O Timing in Datasheet Report on page 7

Figure 3. I/O Timing Data Generation Flow

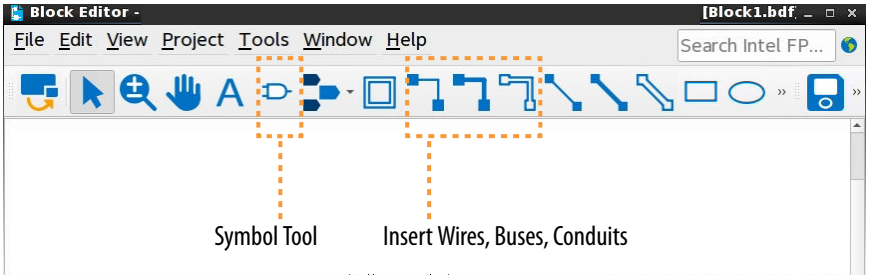


1.1. Step 1: Synthesize a Flip-flop for the Target Intel FPGA Device

Follow these steps to define and synthesize the minimum flip-flop logic to generate initial I/O timing data:

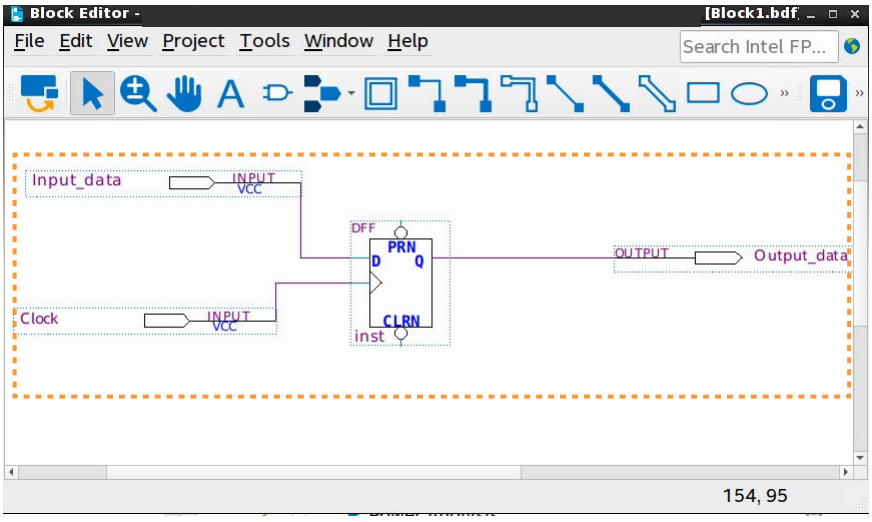
1. Create a new project in Intel Quartus Prime Pro Edition software version 19.3.
2. Click **Assignments ➤ Device**, specify your target device **Family** and a **Target device**. For example, select the **AGFA014R24** Intel Agilex™ FPGA.
3. Click **File ➤ New** and create a **Block Diagram/Schematic File**.
4. To add components to the schematic, click the **Symbol Tool** button.

Figure 4. Insert Pins and Wires in Block Editor



5. Under **Name**, type DFF, and then click **OK**. Click in the Block Editor to insert the DFF symbol.
6. Repeat 4 on page 5 through 5 on page 5 to add an `Input_data` input pin, `Clock` input pin, and `Output_data` output pin.
7. To connect the pins to the DFF, click the **Orthogonal Node Tool** button, and then draw wire lines between the pin and DFF symbol.

Figure 5. DFF with Pin Connections



8. To synthesize the DFF, click **Processing ➤ Start ➤ Start Analysis & Synthesis**. Synthesis generates the minimum design netlist required to obtain I/O timing Data.

1.2. Step 2: Define I/O Standard and Pin Locations

The specific pin locations and I/O standard you assign to the device pins impacts the timing parameter values. Follow these steps to assign the pin I/O standard and location constraints:

1. Click **Assignments ➤ Pin Planner**.
2. Assign pin location and I/O standard constraints according to your design specifications. Enter the **Node Name**, **Direction**, **Location**, and **I/O Standard** values for the pins in the design in the **All Pins** spreadsheet. Alternatively, drag node names into the Pin Planner package view.

Figure 6. Pin Locations and I/O Standards Assignments in Pin Planner

Node Name	Direction	Location	I/O Bank	I/O Standard	Reserved	Slew Rate	Differential Pair	Strict Preservation
Clock	Input	PIN_A6	3D	1.2 V				
Input_data	Input	PIN_A20	3C	1.2 V				
Output_data	Output	PIN_A56	3A	1.2 V				
<<new node>>								

3. To compile the design, click **Processing ➤ Start Compilation**. The Compiler generates I/O timing information during full compilation, and automatically launches the Timing Analyzer with updated netlist at the end of full compilation.

Related Information

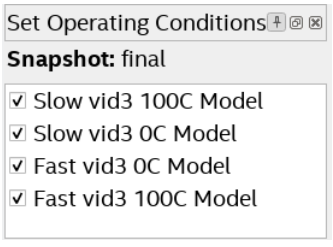
- I/O Standards Definition
- Managing Device I/O Pins

1.3. Step 3: Specify Device Operating Conditions

Follow these steps to set operating conditions for timing analysis following full compilation.

1. The Timing Analyzer launches automatically and updates the timing netlist following full compilation, as Step 2: Define I/O Standard and Pin Locations on page 6 describes.
2. Under **Set Operating Conditions**, select one or more of the available timing models, such as **Slow vid3 100C Model** and **Fast vid3 100C Model**.

Figure 7. Set Operating Conditions Pane

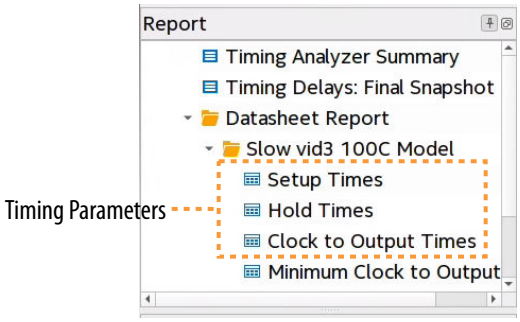


1.4. Step 4: View I/O Timing in Datasheet Report

Generate the Datasheet Report in the Timing Analyzer to view the timing parameter values.

1. In the Timing Analyzer, click **Reports ➤ Datasheet ➤ Report Datasheet**.
2. Click **OK**.

Figure 8. Datasheet Report in Timing Analyzer



The **Setup Times**, **Hold Times**, and **Clock to Output Times** reports appear under the **Datasheet Report** folder in the **Report** pane.

3. Click each report to view the **Rise** and **Fall** parameter values.
4. For a conservative timing approach, specify the maximum absolute value.

Example 1. Determining I/O Timing Parameters from the Datasheet Report

In the following example **Setup Times** report, the fall time is greater than the rise time, therefore $t_{SU}=t_{fall}$.

Figure 9. Setup Times Report

Setup Times

Show: All | Hide | <<Filter>>

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	Input_data	Clock	7.237	7.394	Rise	Clock

In the following example **Hold Times** report, the absolute value of the fall time is greater than the absolute value of the rise time, therefore $t_{H}=t_{fall}$.

Figure 10. Hold Times Report

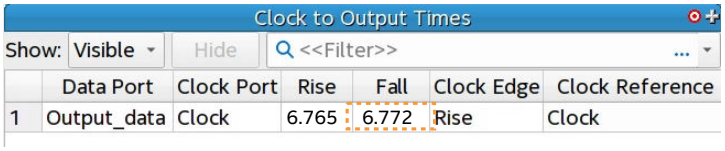
Hold Times

Show: Visible | Hide | <<Filter>>

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	Input_data	Clock	-4.518	-4.618	Rise	Clock

In the following example **Clock to Output Times** report, the absolute value of the fall time is greater than the absolute value of the rise time, therefore $t_{CO}=t_{fall}$.

Figure 11. Clock to Output Times Report



Related Information

- Timing Analyzer Quick-Start Tutorial
- Intel Quartus Prime Pro Edition User Guide: Timing Analyzer
- How To Video: Introduction to Timing Analyzer

1.5. Scripted I/O Timing Data Generation

You can use a Tcl script to generate I/O timing information with or without using the Intel Quartus Prime software user interface. The scripted approach generates text-based I/O timing parameter data for supported I/O standards.

Note: The scripted method is available only for Linux* platforms.

Follow these steps to generate I/O timing information reflecting multiple I/O standards for Intel Agilex, Intel Stratix® 10, and Intel Arria® 10 devices:

1. Download the appropriate Intel Quartus Prime project archive file for your target device family:
   - Intel Agilex devices—https://www.intel.com/content/dam/www/programmable/us/en/others/literature/an/io_timing_agilex_latest.qar
   - Intel Stratix 10 devices—https://www.intel.com/content/dam/www/programmable/us/en/others/literature/an/io_timing_stratix10.qar
   - Intel Arria 10 devices—https://www.intel.com/content/dam/www/programmable/us/en/others/literature/an/io_timing_arria10.qar
2. To restore the `.qar` project archive, launch the Intel Quartus Prime Pro Edition software and click **Project ➤ Restore Archived Project**. Alternatively, run the following command line equivalent without launching the GUI:

   ```
   quartus_sh --restore <archive file>
   ```

   The `io_timing_<device>_restored` directory now contains the `qdb` subfolder and various files.
3. To run the script with the Intel Quartus Prime Timing Analyzer, run the following command:

   ```
   quartus_sta –t <device>.tcl
   ```

   Wait for completion. The script execution may require 8 hours or more because each change on I/O standard or pin location requires design recompilation.
4. To view the timing parameter values, open the generated text files in `timing_files`, with names such as `timing_tsuthtco_<device>_<speed>_<IO_standard>.txt`.

Related Information

- Command Line Scripting
- ::QUARTUS::STA Tcl Package

2. Generating I/O Element Delay Information for Intel FPGAs

You can generate I/O element (IOE) delay information for Intel FPGA devices using the current version of Intel Quartus Prime Pro Edition software GUI or Tcl commands.

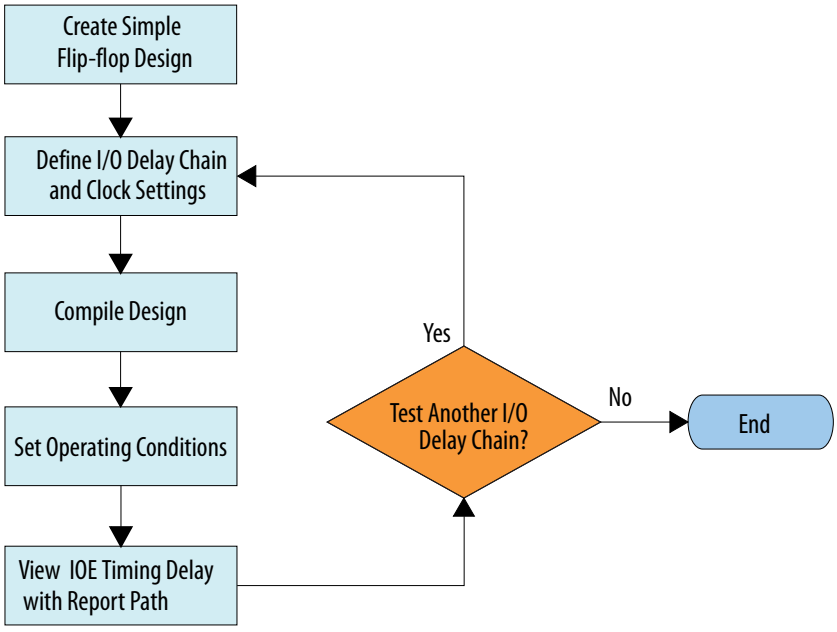
Note: The Tcl script-based method is available only for Linux platforms.

You can specify a different input delay for pin in your design from the pin-to-input register, or a delay from the output register-to-output pin values. This capability allows you to ensure that the signals within a bus have the same delay going into or out of the device. For detailed descriptions of the various IOE structures in different FPGA devices, refer to the FPGA device documentation in related links.

Generating IOE delay information includes the following steps in the flow:

- Step 1: Create Simple Flip-Flop Design on page 11
- Step 2: Define I/O Delay Chain and Clock Settings on page 12
- Step 3: Specify Device Operating Conditions on page 12
- Step 4: View IOE Timing Delay with Report Path on page 13

Figure 12. IOE Delay Information Generation Flow

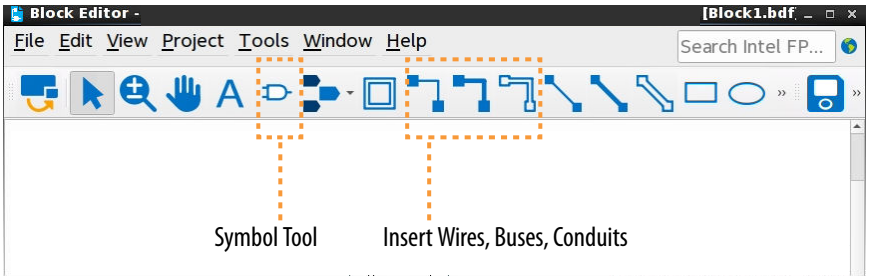


2.1. Step 1: Create Simple Flip-Flop Design

Follow these steps to define and synthesize the flip-flop logic to generate the IOE:

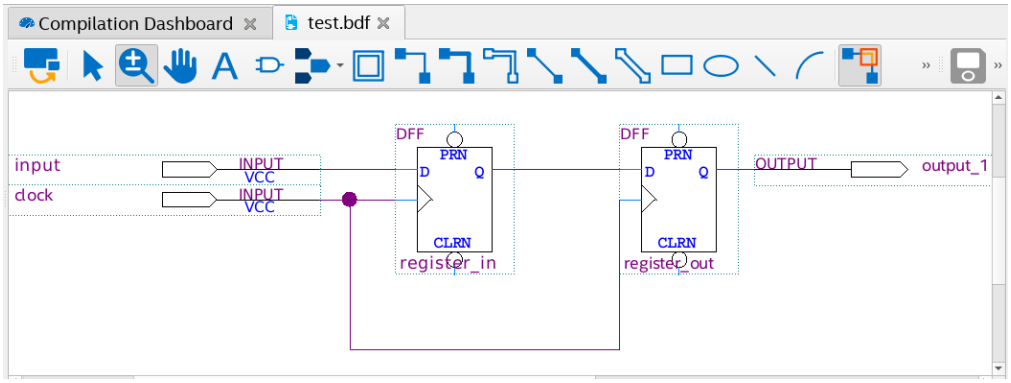
1. Create a new project in Intel Quartus Prime Pro Edition software version 21.3.
2. Click **Assignments ➤ Device**, specify your target device **Family** and a **Target device**. For example, select the **AGFA014R24A** Intel Agilex FPGA.
3. Click **File ➤ New** and create a **Block Diagram/Schematic File**.
4. To add components to the schematic, click the **Symbol Tool** button.

Figure 13. Symbol Tool Button in Block Editor



5. Under **Name**, type DFF, and then click **OK**. Click twice in the Block Editor to insert two instances of the DFF symbol.
6. Repeat 4 on page 11 through 5 on page 11 to add an `input` input pin, `clock` input pin, and `output_1` output pin.
7. Connect the pins to the DFFs using the **Orthogonal Node Tool** button, and draw wire lines between the pins and DFFs symbols, as Figure 14 on page 11 shows.

Figure 14. DFFs with Pin Connections



8. To synthesize the DFFs, click **Processing ➤ Start ➤ Start Analysis & Synthesis**. Synthesis generates the minimum design netlist required to obtain I/O timing Data.

2.2. Step 2: Define I/O Delay Chain and Clock Settings

The I/O delay chain setting that you specify determines the minimum and maximum delay path from input pin to output pin through each register. specifies the range of I/O delay chain settings for this example.

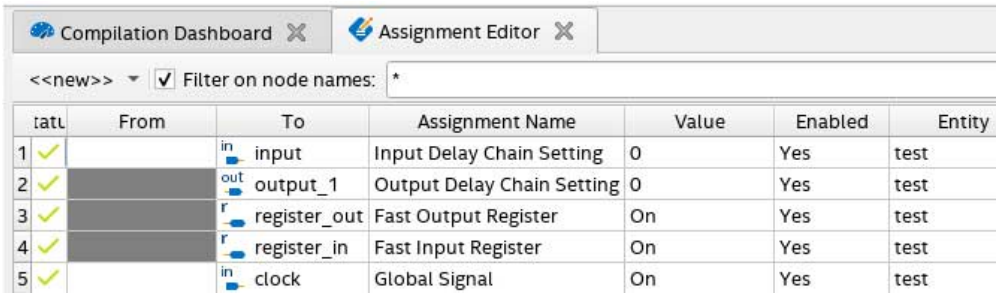
Table 2. I/O Delay Chain Settings for Example

Setting	Maximum Value	Minimum Value
Input Delay Chain	63	0
Output Delay Chain	15	0

To assign the I/O delay chain settings, follow these steps:

1. Click **Assignments ➤ Assignment Editor**.
2. In the Assignment Editor, assign settings to registers and pins, according to your design specifications, as the following example assignments show:

Figure 15. Registers and Pins in Assignment Editor



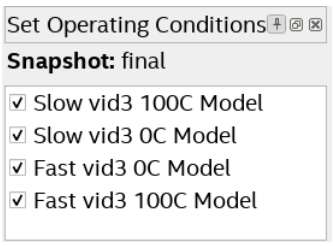
3. To compile the design, click **Processing ➤ Start Compilation**. The Compiler implements the assignments during compilation, and then launches the Timing Analyzer automatically.

2.3. Step 3: Specify Device Operating Conditions

Follow these steps to set operating conditions in the Timing Analyzer after compilation:

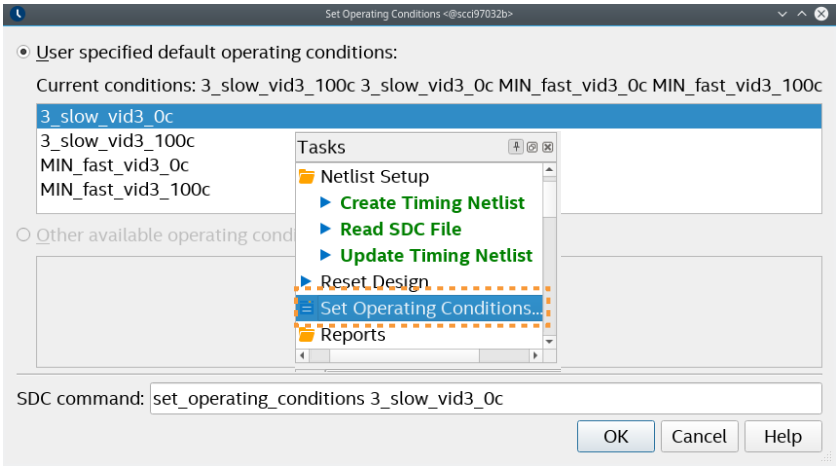
- Under **Set Operating Conditions**, enable or disable one or more operating condition timing models:

Figure 16. Set Operating Conditions Pane



- In the **Tasks** pane, click **Set Operating Conditions**. Specify the desired operating conditions in the **Set Operating Conditions** dialog box.

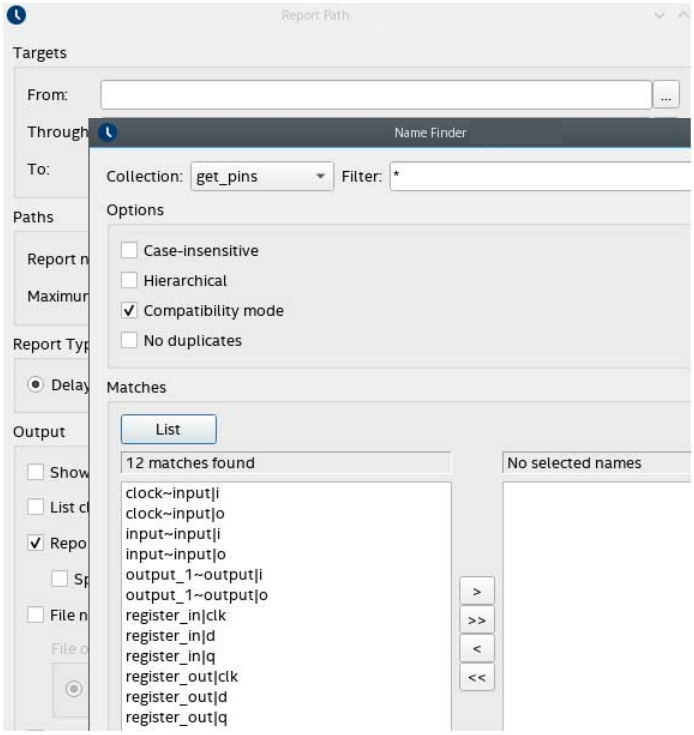
Figure 17. Set Operating Conditions Dialog Box



2.4. Step 4: View IOE Timing Delay with Report Path

Use the **Report Path** command to generate reports showing IOE timing delay.

Figure 18. Report Path Dialog Box and Node Finder



1. In the Timing Analyzer, click **Reports ➤ Path Diagnostics ➤ Report Path**.
2. In the **Report Path** dialog box, define the target path by selecting the path **Targets** in the **From (...)** list using the Node Finder.
3. Click **OK** to generate the report.

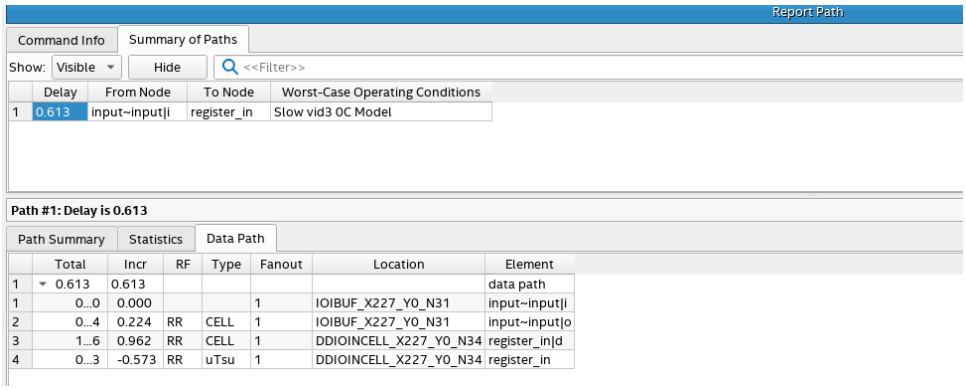
The following show examples show IOE delays after running **Report Path**.

- Example 1: Viewing the Input Element Delay in Reports on page 14
- Example 2: Viewing the Output Element Delay in Reports on page 14

2.4.1. Example 1: Viewing the Input Element Delay in Reports

In the following example, **Report Path** shows the maximum delay of input pin to register for the Slow mode at 0C Model operating condition.

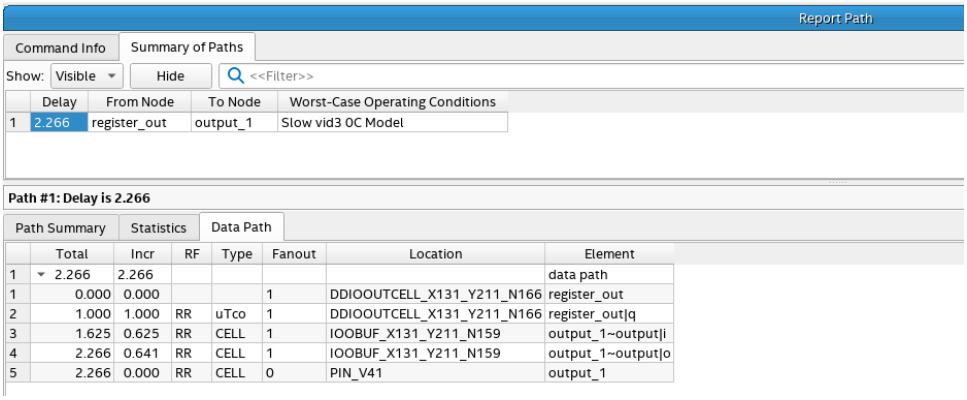
Figure 19. Input Element Delay (Input Delay Chain Setting = 0)



2.4.2. Example 2: Viewing the Output Element Delay in Reports

In the following example, **Report Path** shows the maximum delay of register to output pin for the Slow mode at 0C Model operating condition.

Figure 20. Output Element Delay (Input Delay Chain Setting = 0)



2.5. Scripted IOE Information Generation

As an alternative to the using the GUI, you can create and execute a Tcl script to generate IOE timing information for your target device reflecting different delay values. You can optionally download example scripts that are available for Intel FPGA devices from the Intel FPGA Design Store.

Note: The scripted method is available only for the Intel Quartus Prime software on Linux platforms.

Follow these steps to use scripting to generate IOE information for FPGA devices reflecting these different delay values:

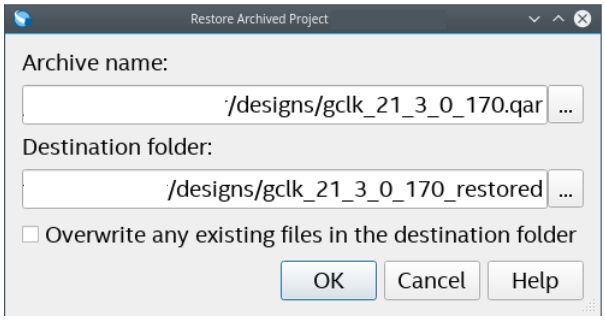
- Delay from input pin to input register
- Delay from output register to output pin.

1. Download the following example example design Intel Quartus Prime project archive:

   gclk_21_3_0_170.qar

2. In the Intel Quartus Prime software, click **Project ➤ Restore Archived Project** and restore the `gclk_21_3_0_170.qar` project.

Figure 21. Restore Archived Project



3. To run the `ioe_<device>_d1_d5.tcl` script with the Timing Analyzer, run the following command:

```
quartus_sta -t ioe_<device>_d1_d5.tcl
```

4. Once the script completes, open the `/timing_files/.txt` file in a text editor. This file contains the IOE timing information.

2.6. Document Revision History for AN 775: Generating Initial I/O Timing Data and I/O Element Delays for Intel FPGAs

Document Version	Intel Quartus Prime Version	Changes
2021.12.09	21.3	• Retitled document to encompass new chapter on IOE timing information. • Revised *Generating Initial I/O Timing Data for Intel FPGAs* topic to include IOE timing information. • Updated *Step 2: Define I/O Standard and Pin Locations* topic for automatic launch of Timing Analyzer following full compilation. • Updated *Step 3: Specify Device Operating Conditions* topic for automatic launch of Timing Analyzer following full compilation. • Added new chapter: *Generating I/O Element Delay Information for Intel FPGAs*.
2019.12.08	19.3	• Revised title to reflect content. • Added support for Intel Stratix 10 and Intel Agilex FPGAs. • Added step numbers to flow.

continued...

Document Version	Intel Quartus Prime Version	Changes
		• Added timing parameter diagrams. • Updated screenshots to reflect latest version. • Updated links to related documents. • Applied latest product naming and style conventions.
2016.10.31	16.1	• First public release.