AN 775: Generating Initial I/O Timing Data

for Intel FPGAs

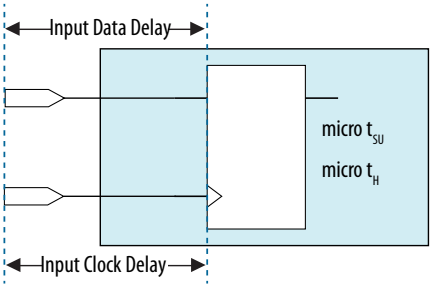
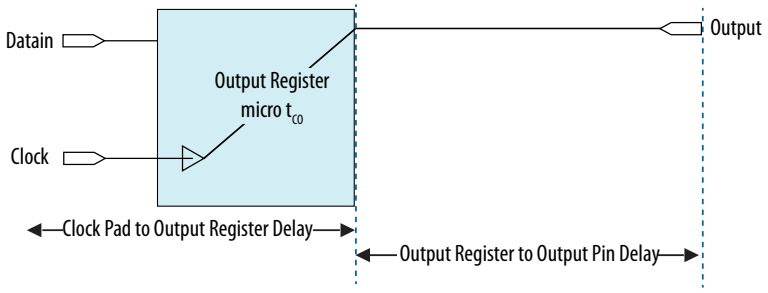
Updated for Intel® Quartus® Prime Design Suite: **19.3**

Contents

1. AN 775: Generating Initial I/O Timing Data for Intel FPGAs

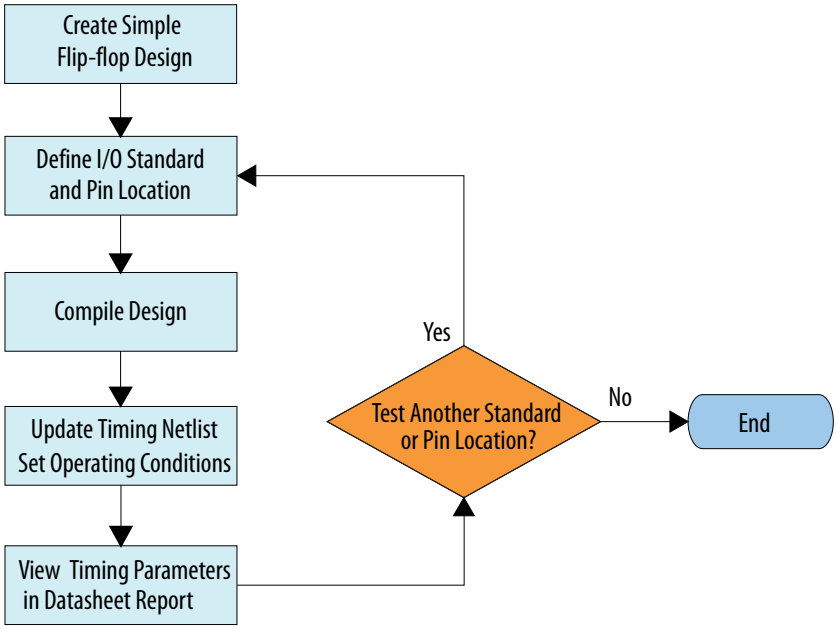
You can generate initial I/O timing data for Intel FPGA devices using the Intel® Quartus® Prime software GUI or Tcl commands. Initial I/O timing data is useful for early pin planning and PCB design. You can generate initial timing data for the following relevant timing parameters to adjust the design 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
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:

Figure 3. I/O Timing Data Generation Flow

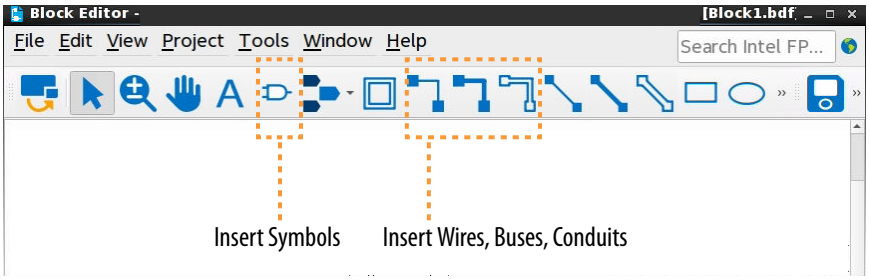


1.1. Step 1: Synthesize a Flip-flop for the Target Intel FPGA Device

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

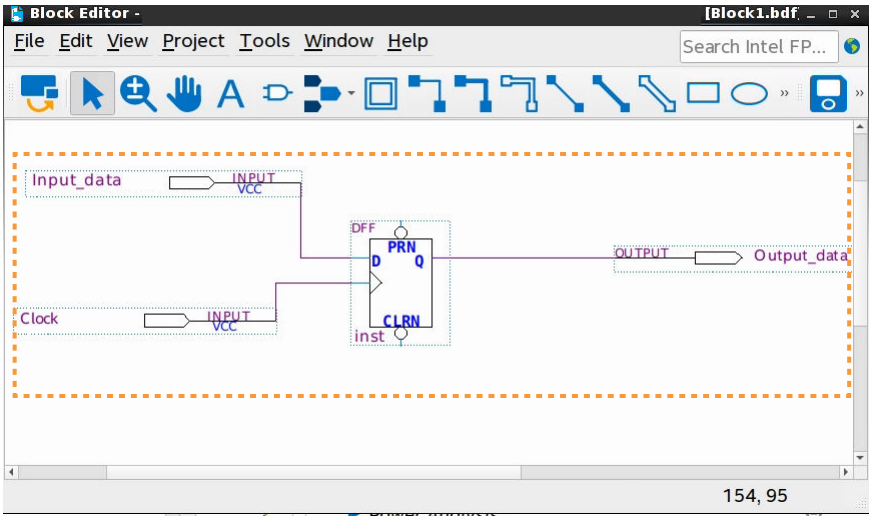
1. Create a new project in Intel Quartus Prime Pro Edition software version 19.3.
2. Click **Assignments ➤ Device**, specify your target device **Family** and a **Target device**. For example, select the **AGFA014R24** Intel Agilex™ FPGA.
3. Click **File ➤ New** and create a **Block Diagram/Schematic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

Related Information

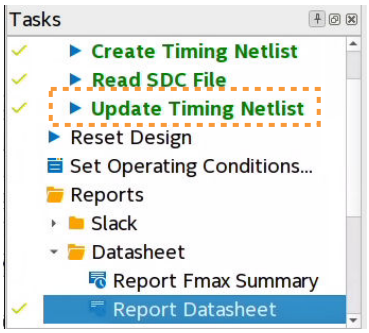
- I/O Standards Definition
- Managing Device I/O Pins

1.3. Step 3: Specify Device Operating Conditions

Follow these steps to update the timing netlist and set operating conditions for timing analysis following full compilation:

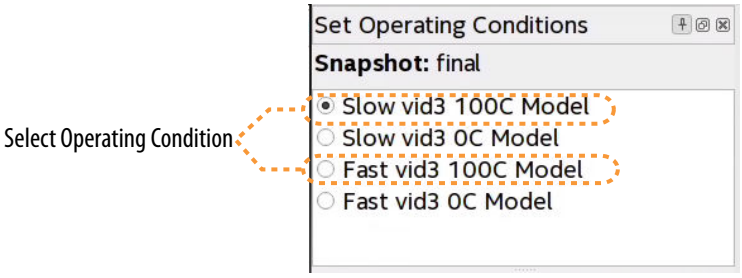
1. Click **Tools ➤ Timing Analyzer**.
2. In the **Task** pane, double-click **Update Timing Netlist**. The timing netlist updates with full compilation timing information that accounts for the pin constraints you make.

Figure 7. Task Pane in the Timing Analyzer



3. Under **Set Operating Conditions**, select one of the available timing models, such as **Slow vid3 100C Model** or **Fast vid3 100C Model**.

Figure 8. Set Operating Conditions in the Timing Analyzer

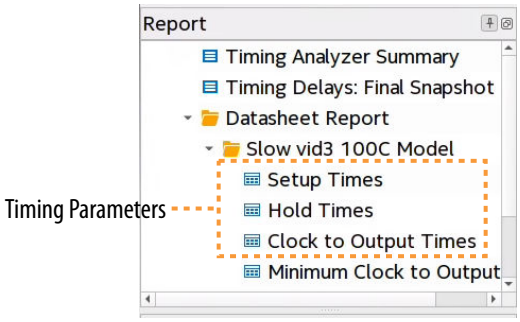


1.4. Step 4: View I/O Timing in Datasheet Report

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 9. 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 10. 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 11. 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 12. Clock to Output Times Report

Clock to Output Times

Show: Visible | Hide | <<Filter>>

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	Output_data	Clock	6.765	6.772	Rise	Clock

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

1.6. AN 775: Generating Initial I/O Timing Data Document Revision History

Document Version	Intel Quartus Prime Version	Changes
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. • 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.