Efinity® Trion® Tutorial

UG-EFN-TUTORIAL-v7.1
May 2025
www.efinixinc.com

Contents

Introduction **3**

1.0 Prepare the Tutorial Files **3**

2.0 Create Your Project **4**

3.0 Run the Flow **5**

3.1 RTL Simulation 5
3.2 Synthesize the Design 6
3.3 Perform Post-Map Simulation 7
View Waveforms 7
3.4 Build the Device Interface 8
3.5 Perform Place & Route 10

4.0 Review Results **11**

4.1 Review Place and Route Results in the Floorplan Editor 11
4.2 Use the Timing Browser 13
4.3 Use the Tcl Command Console 14

5.0 Configure the FPGA **14**

Where to Learn More **15**

Revision History **16**

Introduction

This tutorial walks you through the Efinity® software flow from beginning to end using the example project helloworld and the T8 FPGA on the Trion® T8 BGA81 Development Board. In addition to running the flow, you will adjust design constraints and view place-and-route results. You will use both the graphical user interface (GUI) and the command-line interface.

This tutorial assumes that you have already installed the Efinity software, Icarus Verilog (iVerilog) simulator, GTKWave, and USB drivers for the board according to the instructions in the Efinity Software Installation User Guide.

Note: The Efinity software also supports other simulators; if you want to use one of those instead, the simulator must be installed and set up according to the instructions in "Third-Party Simulator Support" in the Efinity Software Installation User Guide.

Icarus Verilog (iVerilog) is a free Verilog simulation tool you can use to compile and simulate Verilog HDL source code. The software is available as source code or as pre-compiled binaries.

GTKWave is an open-source tool that analyzes post-simulation dumpfiles and displays the results in a graphical interface. It includes a waveform viewer and RTL source code navigator. You can use GTKWave with the iVerilog simulator to analyze and debug your simulation model, or to view any VCD waveform.

Tip: *Windows users:* Add the tool paths (`$iverilog_folder$\bin\` and `$GTKWave_folder$\bin\`) to your System Variables path to ensure the software launches correctly.

Learn more: For detailed information on how to use the Efinity software, refer to the Efinity Software User Guide.

1.0 Prepare the Tutorial Files

To prepare the tutorial files and set up your environment, perform these steps:

1. Open a terminal window.
2. Change to the directory in which you installed the Efinity® software, *<installation path>***/efinity/***<version>*.
3. Type the following commands:
   - Linux:

```
> source bin/setup.sh
> cd project
> mkdir tutorial
> cp -r helloworld/ tutorial/
> cd tutorial/helloworld
```

   - Windows:

```
> bin\setup.bat
> cd project
> md tutorial
> xcopy helloworld tutorial\helloworld\
> cd tutorial\helloworld
```

2.0 Create Your Project

In this step, you create a new project based on the helloworld example design using the **Project** and **Design** tabs in the **Project Editor**.


Open the Efinity GUI


Create new project


Choose a directory to store project data


Import design and constraint files


Delete file

1. Open the Efinity GUI.
2. Create a new project (**File > Create project** or click the new project button). The **Project Editor** opens to the **Project** tab.
3. Type `helloworld` in the **Name** box.
4. In the **Location** box, click the button Choose a directory to store project data.
5. Select the **tutorial/helloworld** directory.
6. Click **Select folder**.
7. Type a project description in the **Description** box. For example, My `helloworld example project`.
8. Choose Trion® as the family.
9. Choose **T8F81** as the device. Keep the default timing model selection.
10. Click the **Design** tab.
11. Type `helloworld` in the **Top Module/Entity** box.
12. Next to the **Design** box, click the button Import design files.
13. Browse to the *<install directory>***/project/tutorial/helloworld** directory.
14. Under **File Option**, choose **File to Import > All Files**. This setting imports design and constraint files.
15. Click **Choose**.
16. Select the **helloworld_tb.v** testbench file and delete it. You will use this file for simulation; it is not a design file.
17. Click **OK** to close the **Project Editor**.

The Dashboard's Project pane displays the settings for your new project.

3.0 Run the Flow

In this step you run through the complete software flow, including simulation, using the GUI and command line interfaces.

Note: After you run different parts of the flow, use the Netlist pane to explore your design by showing the design hierarchy, the elaborated design, and the synthesized netlist. (You can only view the synthesized netlist after you have performed synthesis.)

3.1 RTL Simulation

First, perform RTL simulation on the design's source files. The **helloworld** design includes the **helloworld_tb.v** testbench file for simulation. You should already have installed the iVerilog simulator and GTKWave using the instructions in the Efinity Software Installation User Guide.

1. Open a terminal window.
2. Type the command:
   Linux:
   `efx_run.py helloworld.xml --flow rtlsim`
   Windows:
   `efx_run.bat helloworld.xml --flow rtlsim`

Tip: By default, the Efinity® software calls the iVerilog simulator. If you want to target ModelSim or NCSim, use these options:

`--modelsim` to target the ModelSim simulator.

`--ncsim` to target the NCSim simulator.

The software performs simulation and writes the results to the **helloworld.rtl.simlog** file in the **outflow** directory. Double-click the filename under Simulation in the Dashboard Results tab to open the log file in the Efinity Code Editor.

You can view the results graphically using the GTKWave application. The **helloworld** design includes commands that dump data for use with GTKWave.

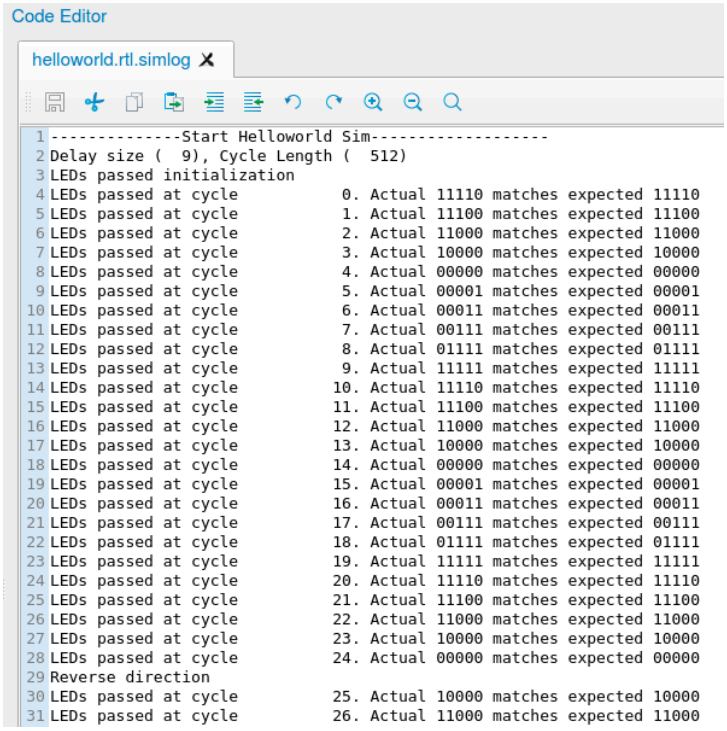
1. Add the following lines to your testbench to generate the dumpfiles:

```
$dumpfile("outflow/<file name>.vcd");
$dumpvars(0, sim);
```

2. Simulate with the iVerilog simulator.
3. Use this command to view the output waveform:

```
gtkwave outflow/<project name>.vcd
```

Figure 1: helloworld Simulation Results



3.2 Synthesize the Design

You perform synthesis from the GUI or from the command line.

Enable/disable automated flow Synthesize the design

1. In the GUI Dashboard, turn off the automated flow.
2. Click the Synthesize button.

OR, at the command line, use the command line:

Linux:

```
> efx_run.py helloworld.xml --flow map
```

Windows:

```
> efx_run.bat helloworld.xml --flow map
```

You view the synthesis report files, **helloworld.map.rpt** and **helloworld.map.out**, in the Results pane in the GUI or in the **outflow** directory.

Note: If you run command line operations while the GUI is open, the GUI state does not automatically sync with the command line state.

3.3 Perform Post-Map Simulation

You perform post-map simulation from the command line.

In a terminal, type the command:

Linux:

```
> efx_run.py helloworld.xml --flow mapsim
```

Windows:

```
> efx_run.bat helloworld.xml --flow mapsim
```

(Use the `--modelsim` or `--ncsim` option to target those simulators.)

Note: Simulation may take several minutes to run in the iVerilog simulator.

View the report file, **helloworld.map.simlog**, in the Results pane in the GUI or in the **outflow** directory. Use GTKWave to view the waveform.

View Waveforms

To use GTKWave to view a waveform:

1. Add the following lines to your testbench to generate the dumpfiles:

```
$dumpfile("outflow/<file name>.vcd");
$dumpvars(0, sim);
```

2. Simulate with the iVerilog simulator.
3. Use this command to view the output waveform:

```
gtkwave outflow/<project name>.vcd
```

3.4 Build the Device Interface

Efinix Trion® FPGAs wrap a Quantum®-accelerated core (logic, memory, and multipliers) with a periphery that sends signals out to the device pins. The device periphery includes blocks such as GPIO pins, PLLs, and oscillators. You use the Efinity® Interface Designer to build the peripheral portion of your Trion® design.

In this section you use the Interface Designer to view the helloworld interface and add a missing GPIO resource.


Open Interface Designer


Show/Hide Editor


Show/Hide Resource Assigner


Create Block


Generate Interface Output Files


Save


Enable/disable automated flow


Place


Package Planner

1. Open the Interface Designer. The tool opens to a summary of your design and displays a list of interface blocks organized into categories.

   Note: The first time you launch the Interface Designer, it builds a cache and may be a little slow to load.

2. Review the assignments for `led[3]`.

   a) Expand **GPIO (6)**.

Tip: The number in parentheses indicates how many blocks your design implements in each category. For example the `helloworld` design has 6 GPIO pins.

   b) Select **led[3]**. The tool displays the resources led[3] uses in the Block Summary to the right.
   c) Click Show/Hide Editor to open the Block Editor if it is not already open.
   d) Click Show/Hide Resource Assigner to open the Resource Assigner table and review the settngs.

3. Review the assignment in the Package Planner.

   a) Select **GPIO (6)** > **led[3]**.
   b) Click the Package Planner icon in the toolbar.
   c) Find pin E3 and click it. The **Pin Information** tab opens and displays the information about the pin such as the I/O bank and voltage.
   d) Close the Package Planner.

4. Generate constraints by clicking the Generate Interface Output Files button.
   The software saves the interface design and creates the following output files in the **outflow** directory:
   - **helloworld.interface.csv**—Constrains the FPGA design pins used in the interface between the core and the periphery.
   - **helloworld.pt.rpt**—Provides information about the interface.
   - **helloworld.pinout.csv**—Contains the board design pinout in CSV format.
   - **helloworld.pinout.rpt**—Has the board design pinout in a nicely formatted text file format.
   - **helloworld.pt_timing.rpt**—Timing report for the Trion® interface logic.
   - **helloworld.pt.sdc**—Template SDC file to constrain the FPGA design pins based on the interface configuration.

- **helloworld_template.v**—Template Verilog HDL file defining the FPGA design pins based on the interface configuration.

5. Close the Interface Designer and return to the Efinity main window.
6. Turn off the automated flow if it is not already off.
7. Click the Place dashboard button to run placement only. When placement completes, the Result pane displays under the dashboard.
8. Review the Results table, and notice that it reports your design has a missing interface pin.
9. Double-click **Result > Placement > helloworld.place.rpt**. The placement report opens.
10. Review the report. Under the IO Placement Summary, notice that `led[4]` is unassigned. It should be assigned to a GPIO output resource.
11. Open the Interface Designer.
12. Create a new block for `led[4]`.
    a) Select **Design : T8F81 > GPIO**.
    b) Click Create Block to create a new block instance.
    c) Enter `led[4]` as the instance name and press return.
    d) Choose output as the mode.
    e) Save.
    f) Click the Generate Interface Output Files button. Before generating constraints and other output files, the software performs a design check to look for problems in the design. In this example, you did not assign a resource to the new instance, so the software issues an error.
    g) Click **OK** in the message window. The Message Viewer opens and displays more detail about the design problem.
    h) Select **led[4]** in the Design Explorer.
    i) Open the Resource Assigner.
    j) In the Resource cell for led[4], type `GPIOL_21`. Alternatively, you can double-click in the Resource cell for led[4] and choose **GPIOL_21** from the resource list.
    k) Press Enter. The software assigns `led[4]` to GPIOL_21 and indicates the assignment in the resource list with a checkmark.
    l) Generate Efinity constraint files again.
13. Close the Interface Designer and return to the Efinity software.

3.5 Perform Place & Route

You perform place and route from the GUI or from the command line.

 Enable/disable automated flow  Place the design

1. In the GUI, turn the automated flow back on.
2. Click the Place dashboard button.

   If automated flow is off, the software goes through placement only; if it is on, the software goes through placement and routing and bitstream generation.

To perform place and route and generate at the command line, use the following command:

Linux:

```
> efx_run.py helloworld.xml --flow pnr
```

Windows:

```
> efx_run.bat helloworld.xml --flow pnr
```

View report files in the Results pane in the GUI or in the **outflow** directory.

4.0 Review Results

The software includes several tools with which you review and cross-probe results:

- Timing Browser
- Floorplan Editor
- Tcl Command Console

The software writes timing information to the **helloworld.timing.rpt** file in the **outflow** directory. You can also view this file in the GUI Results pane.

The helloworld SDC file defines the 33 MHz clock, `clk`, and a 5 ns input and output delay for all outputs relative to `clk`.

Note: To apply SDC changes to your design, you must recompile.

4.1 Review Place and Route Results in the Floorplan Editor

The Floorplan Editor provides a visual representation of the device. After you compile, the viewer shows the tiles used for logic, routing, memory, etc.

View Floorplan	World view	Show fanin
Show all nets	Net tracer	Show fanout
Clear net trace	Toggle Floorplan Legend/Filter	

Tip: Close the Console or detach the Floorplan Editor for easier viewing.

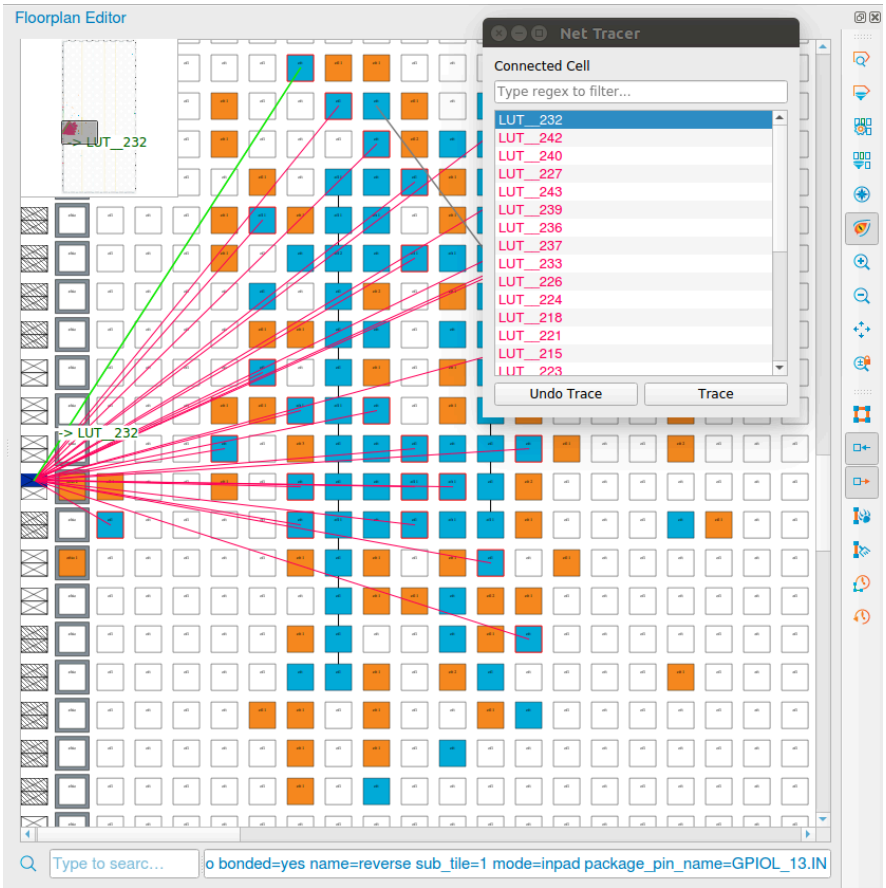
Note: The Floorplan Editor may be disabled if the software has not loaded the place-and-route data. If you have disabled auto-loading, you cannot view place-and-route results in the Floorplan Editor or Timing Browser, or use the Tcl console. To enable these tools, click the **Load Place and Route Data** button in the main window. You enable and disable auto-loading in the Efinity® Preferences dialog box (**File** > **Preferences**).

1. Open the Floorplan Editor.
2. Turn on the World View to orient yourself in the floorplan.
3. Zoom in. The tiles with a solid color in the center are used blocks (blue for logic and orange for routing; toggle the Floorplan Legend/Filter to view the colors for other blocks). When you click a tile, the block coordinates and name are displayed.
4. Turn off Show All Nets.
5. With a logic tile selected, turn on Show Fanin and Show Fanout. The software shows fanout lines in red and fanin lines in blue.
6. Open the Net Tracer. The Net Tracer shows the cells that are connected to the selected cell. Double click a connected cell name to jump to that cell in the floorplan. The Net Tracer shows your path as you jump from cell to cell.
7. Close the Net Tracer. The traces persist when the Net Tracer window is closed.
8. Click Clear Net Trace to remove the tracing lines.

9. Click used logic tiles (blue) to observe the flow of logic through the tiles.

Note: Leave the Floorplan Editor open to explore timing in the next section.

Figure 2: Floorplan Editor



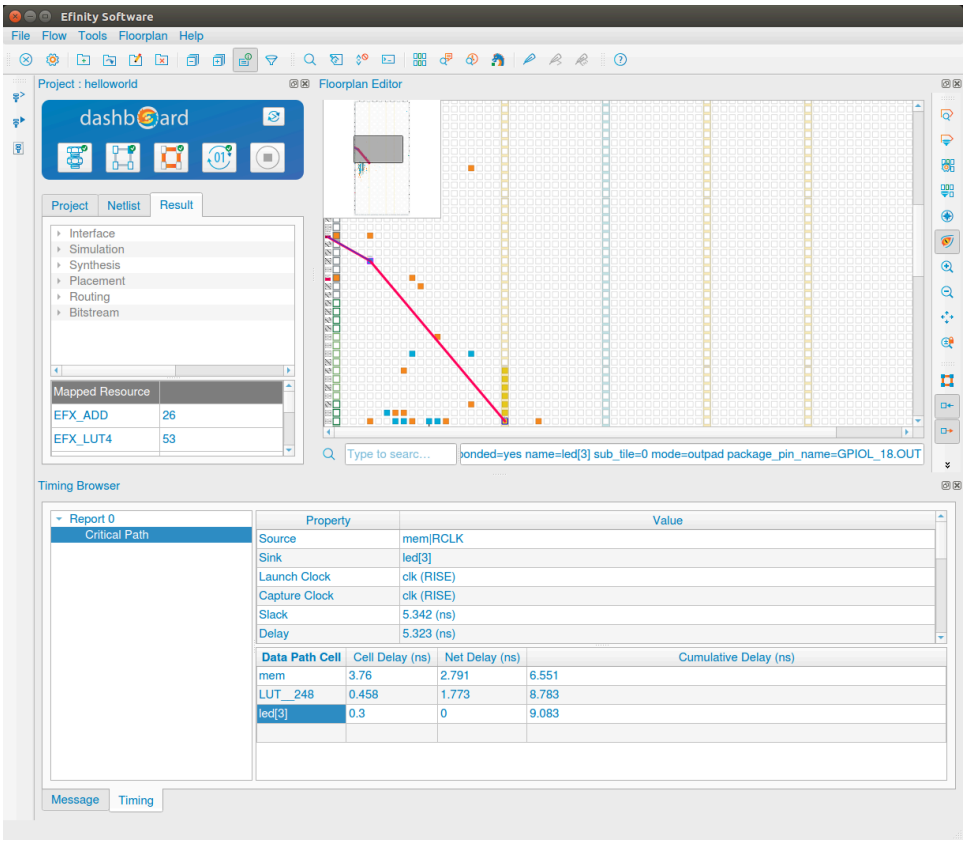
4.2 Use the Timing Browser

Use the Timing Browser with the Floorplan Editor to explore the your design's critical paths and the cells on those paths. When you first open the Timing Browser, it displays the most critical path (the path with the least slack) and the cells on that path.

 Timing Browser  Show timing paths  Show timing delay

1. Open the Floorplan Editor if it is not already open.
2. Open the Timing Browser.
3. Turn on Show Timing Paths.
4. In the Timing Browser, click `led[0]` under Data Path Cell. The Floorplan Editor shows the I/O pad that is the end of the critical path.
5. Turn on Show Timing Delay. The Floorplan Editor shows the delay for the cells and nets.

Figure 3: Exploring the Design's Timing



4.3 Use the Tcl Command Console

You use the Tcl Command Console to analyze and explore timing.

Tcl Command Console

1. Open the Tcl Command Console. If you had closed the Console previously, clicking the Show/Hide Tcl Command Console button opens it.
2. Type `source example_report.tcl` to view the helloworld example timing reports in the console.

The Timing Browser displays the new reports. Click on paths in the browser to view them in the Floorplan.

Learn more: Refer to "Appendix B: Tcl Timing Report & Flow Commands" in the Efinity Software User Guide for a listing of available Tcl commands. For help on available Tcl commands, type `help -category <sdc or timing>` in the Tcl Command Console.

5.0 Configure the FPGA

You are now ready to configure the FPGA on the Trion® T8 BGA81 Development Board.

Programmer Select image file

1. Connect the USB cable to the board and to your computer.
2. To configure on the command line, use the command:

   Linux:

   ```
   > efx_run.py helloworld.xml --flow program
   ```

   Windows:

   ```
   > efx_run.bat helloworld.xml --flow program
   ```
3. To configure using the Programmer GUI, launch the GUI by clicking the Programmer icon in the Efinity software or use the command line:
   - *Windows*—<*installation directory*>**\bin\efinity_pgm.bat**
   - *Linux*—Use the command `efinity_pgm.py`
4. Choose the Trion® T8 BGA81 Development Board as the **USB Target**. The board name appears as **AVR USB HID DEMO**.
5. Click the Select Image File icon.
6. Browse to the **outflow** directory and choose <*helloworld*>**.hex**.
7. Choose **SPI Active** or **SPI Passive** configuration mode.
8. Click **Start Program**. The console displays programming messages.

When the software has finished programming:

1. Observe the 5 green LEDs sweep across the LED display.
2. Press SW3 to change the sweep direction.
3. Press SW2 (reset) to stop the LED movement. The LEDs resume sweeping when you release SW2.
4. When you are finished, disconnect the USB cable from the board and your computer.

Where to Learn More

The Efinity® software includes documentation as PDF user guides and on-line HTML help. This documentation is provided with the software. You can also access the latest versions of PDF documentation in the Support Center:

- **Efinity Software User Guide**
- **Efinity Synthesis User Guide**
- **Efinity Timing Closure User Guide**
- **Efinity Software Installation User Guide**
- **Efinity Trion Tutorial**
- **Efinity Debugger Tutorial**
- **Topaz Interfaces User Guide**
- **Titanium Interfaces User Guide**
- **Trion Interfaces User Guide**
- **Efinity Interface Designer Python API**
- **Quantum® Trion Primitives User Guide**
- **Quantum® Titanium Primitives User Guide**
- **Quantum® Topaz Primitives User Guide**

In addition to documentation, Efinix field application engineers have created a series of videos to help you learn about aspects of the software. You can view these videos in the Support Center.

Revision History

Table 1: Revision History

Date	Version	Description
May 2024	7.1	Generate Efinity Constraints Files button renamed as Generate Interface Output Files. (DOC-2296)
August 2022	7.0	Removed the Debugger tutorials; they now appear in a separate Efinity Debugger Tutorial. Removed hardware and software requirements and USB driver instructions. Go to the Efinity Installation User Guide for that information. Added steps for using the Package Planner tool. Updated the introduction to explain which device is used in the tutorials.
June 2022	6.5	Updated steps for Create Your Project. (DOC-809) Pointed to new Sourceforge location for GTKWave. (DOC-797)
December 2021	6.4	Updated machine memory requirements. Added a note about how to load place-and-route data for the Floorplan Editor. Added the FTDI FT4232H Mini Module when installing drivers. (DOC-597)
June 2021	6.3	Supported Ubuntu version is v18.04 or higher. v16.04 is end of life. (DOC-433) Updated Windows USB driver installation instructions.
December 2020	6.2	Added the requirement to install the Microsoft Visual C++ 2015 x64 and x86 runtime libraries for the standalone Programmer.
November 2020	6.1	Updated instructions on installing the USB driver for Windows.
June 2020	6.0	Windows 7, Red Hat v6, and CentOS v6 no longer supported. Added FTDI Dual RS232 HS mini module in steps to install the USB driver. Added Where to Learn More topic with documenation listing. Minor changes to the Debugger tutorials.
December 2019	5.0	Updated for the v2019.3 software. Added tutorials for the Debugger automated and manual flows. Added information on targeting ModelSim and NCSim for simulation.
August 2019	4.5	Updated for the v2019.2 software. Added command-line instructions for using the **efx_run.bat** command.
April 2019	4.4	Updated for the v2019.1 software.
January 2019	4.3	Updated for the v2018.4 software. Updated information on using the Programmer. Fixed typos.

Date	Version	Description
October 2018	4.2	Updated for the 2018.3 software. Updated the Interface Designer steps. Added Python as an optional requirement. Minor changes throughout.
June 2018	4.1	Removed Python requirement; as of this release, Python is included with the software. Added the requirement that Windows users install the Microsoft Visual C++ 2015 x64 runtime library.
April 2018	4.0	Updated for v2018.0 software release. Added steps for using the Efinity Interface Designer. Added steps for programming with the Programmer GUI and Trion development board.
November 2017	3.1	Updated to target the Quantum family.
May 2017	3.0	Updated for v2017.0 software release. Renamed Floorplan Viewer as Floorplan Editor.
May 2016	2.0	Updated for v2016.0 software release. Documented Timing Browser and Tcl Console.
July 2015	1.1	Updated Floorplan Viewer information
May 2015	1.0	Initial release.