

Introduction

Clock Boosting, supported in Lattice Semiconductor’s FPGA device families, is the introduction of clock skew on a target flop to increase the setup margin. The implementation of this clock skew is accomplished in two different approaches. One approach is to use programmable delay taps built into the architecture in the LatticeSC™ and LatticeSCM™ device families. The second approach inserts delays determined algorithmically into the clock tree to provide additional clock skew in the LatticeEC™, LatticeECP™, LatticeXP™ and MachXO™ device families. The two approaches are described below.

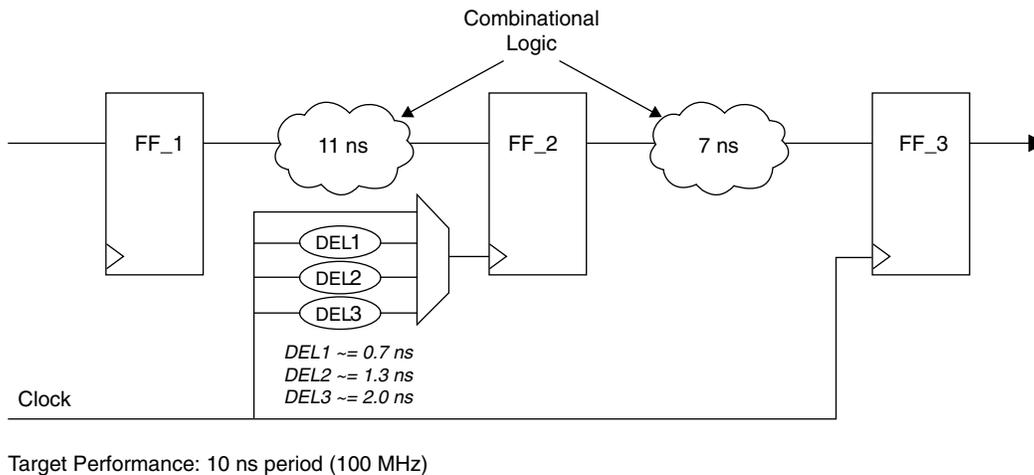
LatticeSC/M Device Families

Clock Boosting is the introduction of clock skew on a target flop to increase the setup margin. To implement this clock skew, every programmable flip-flop in the device features programmable delay elements located in front of the clock inputs. The automated Clock Boosting tool will attempt to meet setup constraints by introducing delays to as many target registers as needed to meet timing. In effect, it borrows from the slow path setup time. The following bullets summarize how Clock Boosting is accomplished in the LatticeSC/M device families.

- A 4-tap delay cell structure in front of the clock port of every flip-flop in the device (including I/O flip-flops).
- Ability to borrow clock cycle time from one easily-met path and give this time to a difficult-to-meet path.

Clock Boosting is typically most useful in designs that are only missing timing on a few paths for one or two preferences. If the design is missing timing by more than a few nanoseconds on any given path, Clock Boosting will not be able to schedule skew in a way that will eliminate enough timing errors to make the critical preference.

Figure 1. LatticeSC/M Clock Boosting Example



The example illustrated in Figure 1 shows two register-to-register transfers that both need to meet the 10 ns period constraint. By using delay cell DEL2 to delay the clock input on flip-flop FF_2, the first register transfer will make its period constraint with a new minimum period of ~9.7 ns and the second register transfer will make its period constraint by ~8.3 ns.

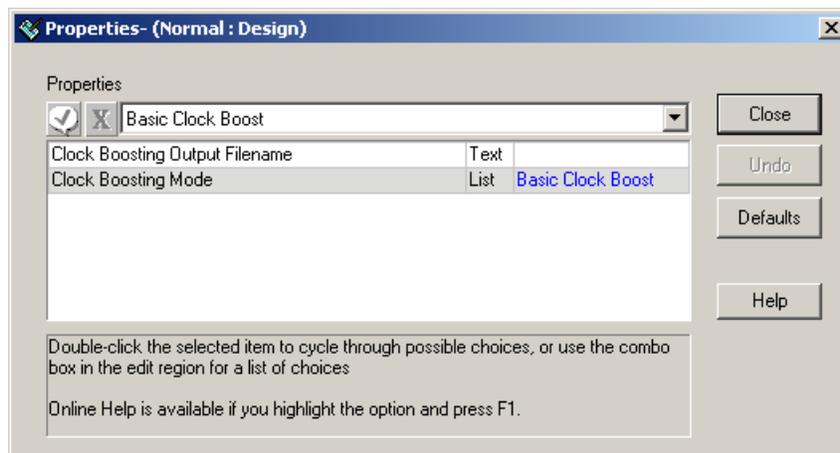
The D1, D2, and D3 delays shown in Figure 1 are variable depending on the speed grade and Lattice FPGA device family.

To Run Clock Boosting in the Project Navigator:

1. In the Project Navigator Sources window, select the target device (LatticeSC/M).
2. In the Processes window, right-click **Clock Boosting** under the **Place & Route Design** process, and then select **Properties** to open the Properties dialog box.
3. Select the **Clock Boosting Output Filename** property from the property list and type the name of the output file name in the edit region (<file_name>.ncd).
4. To select the Clock Boosting Mode click on **Clock Boosting Mode**, then click on the down arrow, to the left of the Close box, then select from **Basic Clock Boost**, **“Maximize Frequency or Hold-time Correction only”**.
5. Click **Close** to close the dialog box.

As shown in Figure 2, the original .ncd and .prf files as well as the output .ncd file are typed into the corresponding entries. Checking **Maximize Frequency** will push the tool to improve the frequency beyond the input preference requirement. This is generally only useful for benchmarking.

Figure 2. Clock Boosting Properties Window for LatticeSC/M



Other important considerations on the practicality of using clock boosting:

- Some circuits show big improvements, while others have no gain. Clock Boosting results are design-dependent.
- Clock Boosting uses maximum and minimum delay values.
- Automatic clock boosting identifies skew and tries to fix hold time issues, it will not cause more hold time violations. If the designer would like to double check the results, run Trace twice, once with regular, maximum delay analysis, and again with minimum delays. The designer should then read over both resultant .twr timing reports to make sure there are no timing errors. The minimum delay analysis is done by checking the Check Hold Times checkbox in the Trace Options GUI window.

Based on recent LatticeSC/M benchmark data, Clock Boosting achieves 11% f_{MAX} gain on average. The test results range from 0% to 28%.

LatticeECP/EC, LatticeXP and MachXO Device Families

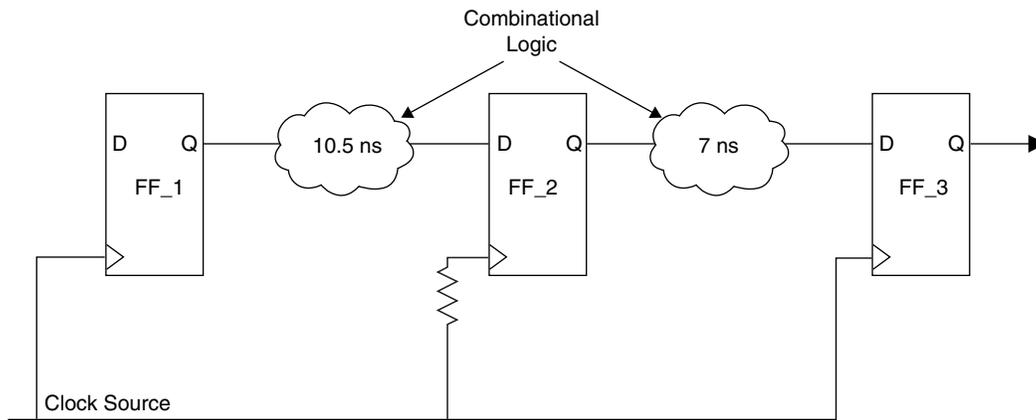
Clock Boosting, supported in the LatticeECP/EC, LatticeXP and MachXO device families, is the introduction of clock skew on a target flop to increase the setup margin. To implement this clock skew, every synchronized flip-flop in the device uses a placed and routed NCD file. This process first finds the optimal clock skew for each flip-flop, then inserts unused general routing resources into the clock tree to provide additional clock skew. The Clock Boosting algorithm will attempt to meet setup constraints by introducing delays to as many target registers as needed to

meet timing, in effect, it borrows from the slow path setup time. The following bullets summarize how Clock Boosting is accomplished in these device families.

- Unused general routing resources are inserted into the clock path for each of the targeted registers to provide additional clock skew for every synchronized flip-flop in the device.
- Ability to borrow clock cycle time from one easily-met path and give this time to a difficult-to-meet path

Clock Boosting is typically most useful in designs that are only missing timing on a few paths for one or two preferences. If the design is missing timing by more than a few nanoseconds on any given path, Clock Boosting will not be able to schedule skew in a way that will eliminate enough timing errors to make the critical preference.

Figure 3. LatticeECP/EC, LatticeXP and MachXO Clock Boosting Example



Target Performance: 10 ns period (100 MHz)

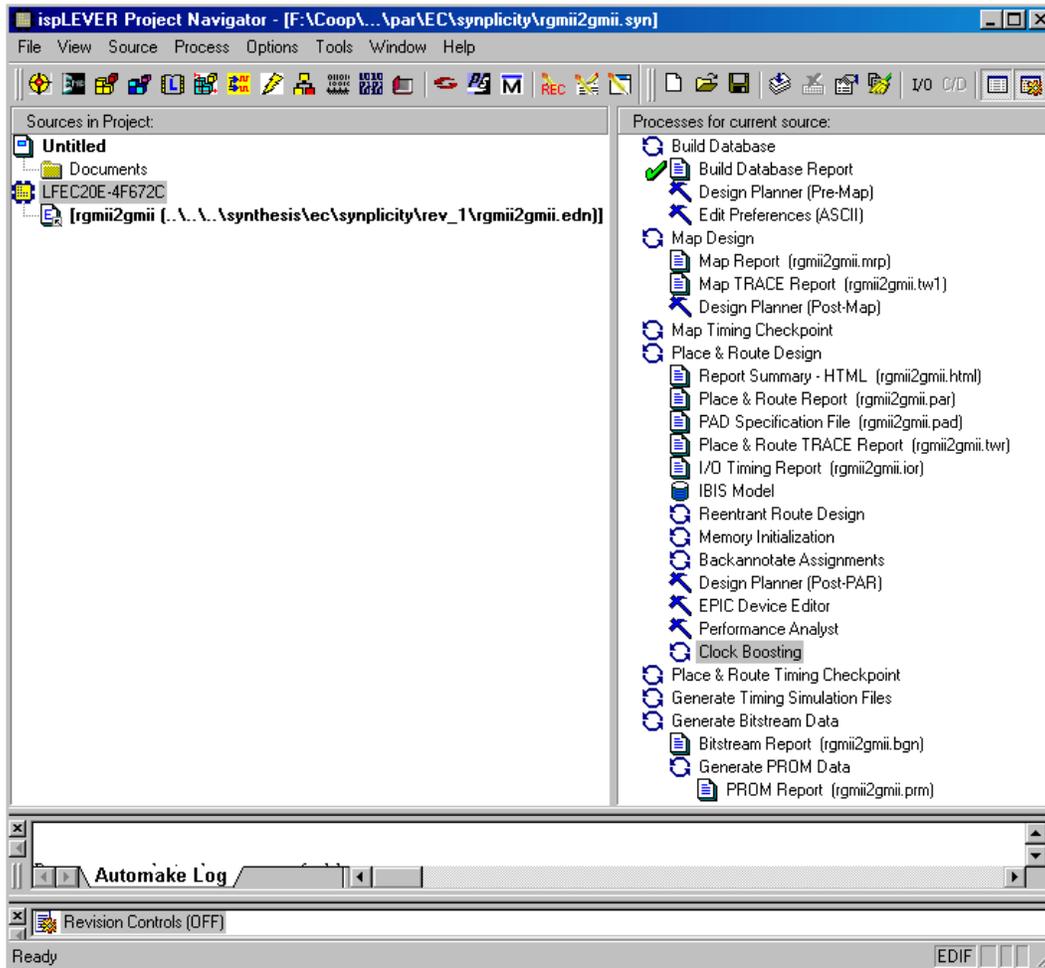
The example illustrated in Figure 3 shows two register-to-register transfers that both need to meet the 10 ns period constraint. By inserting unused general routing resources into the clock tree (total delay = 700 ps to delay the clock input on flip-flop FF_2, the first register transfer will make its period constraint with a new minimum period of ~9.8 ns and the second register transfer will make its period constraint by ~7.7 ns.

The software will search for an unused general routing resource with the proper delay to fit the timing requirements.

To Execute Clock Boosting in the Project Navigator

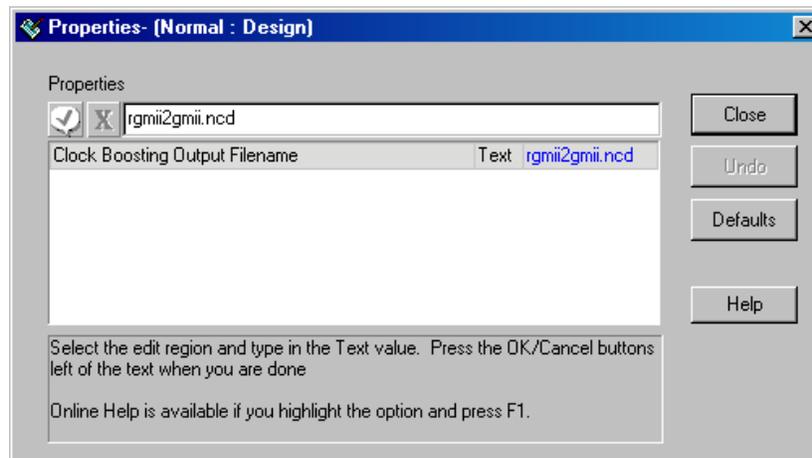
1. In the Project Navigator Sources window, select the **target device** (LatticeECP/EC, LatticeXP, MachXO).
2. In the Processes window, right-click the **Clock Boosting** under **Place & Route Design** process, and then select **Properties** to open the Properties dialog box.
3. Select the **Clock Boosting Output Filename** property from the property list and type the name of the output file name in the edit region (<file_name>.ncd).
4. Click **Close** to close the dialog box.
5. In the Processes window, double-click the **Clock Boosting** under **Place & Route Design** as shown in Figure 4.

Figure 4. Project Navigator Window



As shown in Figure 5, the <output>.ncd file is typed into the Clock Boosting Properties window.

Figure 5. Clock Boosting Properties Window for LatticeECP/EC, LatticeXP and MachXO



If the designer does not want to save a different Clock Boosting output file, they can do the following in the Project Navigator:

1. In the Project Navigator Sources window, select the **target device**.
2. In the Processes window, double-click the **Clock Boosting** under **Place & Route Design** as shown in Figure 5.

The output of the stand-alone Clock Boosting (SCB) will rename the original <design>.ncd file to <design_save>.ncd then override the original <design>.ncd file with the SCB results.

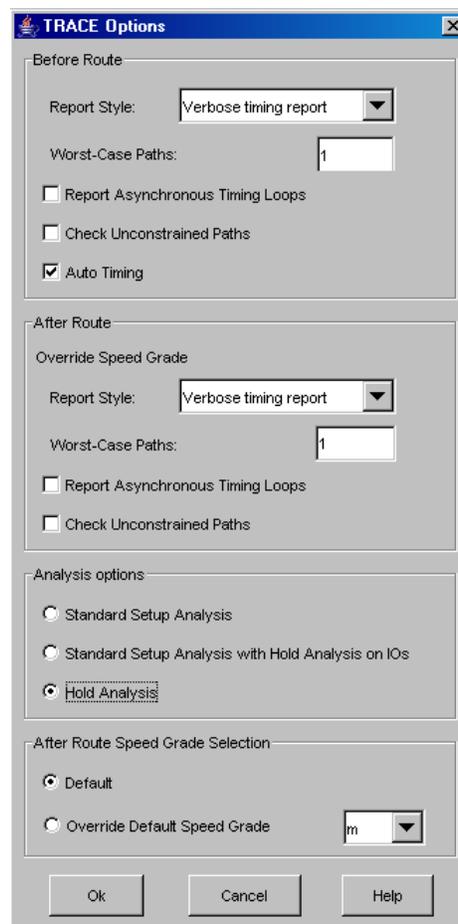
The SCB par file <design_cb>.par will show the timing report after SCB or “No clock boost for this design”. It is suggested that the designer run Trace before SCB so that the <design>.twr file can be compared with the <design_cb>.par file.

Considerations on the practicality of using Clock Boosting (SCB):

- Some circuits show large improvements, while others have no gain. Clock Boosting results are very design-dependent.
- Clock Boosting identifies and tries to fix hold time issues. SCB will not create more hold time violations. To double-check the results, run Trace twice, once with regular, maximum delay analysis, and again with minimum delays. Read both resultant .twr timing reports to make sure there are no timing errors. The minimum delay analysis is done by checking the **Hold Analysis** checkbox in the **Trace Options window** in the **Analysis options** section. In the Project Navigator GUI go to the **Tools** pull-down menu and select **Trace Options...** See Figure 6 for the Trace Options window.

Clock Boosting is a post process, used after the regular route is completed. Clock Boosting achieves a 5% f_{MAX} gain on average. The test results range from 0% to 7%.

Figure 6. Trace Options Window



Running SCB in Command Line

To run Place and Route (par) and SCB together:

- `par -exp parCB=ON`
- PAR is located in the `ispTOOLS\ispFPGA\bin\nt` directory.

To run SCB alone:

- Run regular par. It can get the `<design>.ncd` file.
- Then run the stand-alone SCB.

```
mv <design>.ncd <design>_route.ncd
```

```
par -w -p -k -i 3 -exp parCBOnly=ON <design>_route.ncd <design>.ncd <design>.prf
```

If there is no need to save a copy of the original ncd, then:

- `par -w -p -k -i 3 -exp parCBOnly=ON <design>.ncd <design>.ncd <design>.prf`

Where: w = Overwrite. Allows the overwrite of an existing file (including the input file). If the specified output is a directory, this allows files in the directory to be overwritten.

-w = Overwrite.

-p = Do not run placement.

-k = Skip constructive placement. Run optimize placement and then enter the router.
Note: Use -k -p to do reentrant routing.

-i = Run n iterations of the router.
Default: Run until router decides it will not complete without diverging.

-exp = Explorer string, to turn on/off special place and route options.

If the clock skew cannot be improved via the command line flow, the following statement will appear:

- “No clock boost for this design”

The advantages of using the stand-alone SCB vs. running Place and Route and then SCB include:

1. Designers can get both regular and SCB .ncd files for reference.

The regular Place and Route and Timing Analyzer can be run. If not satisfied with the results, try the stand-alone SCB and compare the results.

2. The use of the stand-alone SCB is the same as using CST for LatticeSC/M devices.

Note that the SCB takes a little extra time to load. Tests show that the stand-alone SCB takes about 10% - 20% of regular Place and Route time. The stand-alone SCB takes about 5% more CPU time than Place and Route + SCB.

Revision History

Date	Version	Change Summary
October 2006	01.0	Initial release.
April 2007	02.0	Added support for LatticeECP/EC, LatticeXP, and MachXO device families.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)

e-mail: techsupport@latticesemi.com

Internet: www.latticesemi.com