Multi-Pass Place and Route Lab

Multi-Pass Place and Route Lab

Introduction

This lab will use a project similar to the FLASH project used in previous labs. However, in this lab, we will use the Multi-Pass Place and Route, and the Router Options to find the best performance for this design.

The Place and Route Options window in the Implementation Options dialog box, is the push-button method of improving the Flow Engine's place and route.

The MPPAR command is used to run multiple place and route iterations on a selected version. The number of place and route iterations that the software should run and the number of iterations to save can be specified (see Figure 1). Each iteration that is saved is added as an implementation revision in the M1 Design Manager.

The Re-entrant Route Options are used to further route designs that have trouble meeting timing constraints, or designs that do not completely place and route (see Figure2).

Since proper placement is much more important in getting the best performance and utilization, it is best to run MPPAR through multiple iterations.

It is recommended that customers **only** utilize the Re-entrant Route feature if their design has not placed and routed sufficiently to meet their timing specifications. It is not usually necessary to use these features to get a project to place and route completely. These tools are available to help customers get **peak utilization** and **peak performance** out of their FPGAs.

About the Place and Route Push-Button Tab

The following options are available in the Place and Route Tab of the Implementation Options dialog box (refer to figure 1).

- **Placement Options**
  Use this option to specify the algorithms that the placer uses to place a design. Use the Placement Effort slider bar to select an effort level setting of 1,2,3,4, or 5. Higher effort provides better placement results at the expense of longer run times. The default value is 4.

- **Run_Routing Passes**

Use this option to set the maximum number of routing passes that the router runs in a design. The router attempts to completely route a placement with each pass. Users can set the number of passes to a value from 1 to 1000 or to Auto. Auto runs the router until it routes to 100% completion or intelligently determines it cannot complete the routing. A higher number of iterations provide better routing results at the expense of longer run times. The default is Auto.

- **Run_Delay Based Clean-up Passes**
  Use this option to further route an already routed design. The router makes routing decisions based on computed delay times between sources and loads on the routed nets, and reroutes to minimize the delays. This option is useful to manually route a portion of the design and then automatically route the design or to run additional delay reduction passes. Set the number of delay-based cleanup passes to run by choosing a number from 1 to 5. Xilinx recommends running two delay-based cleanup passes. The default is 0.

- **Use Timing Constraints During Place and Route**
  Select this option to produce a high-performance implementation of the design. The router uses timing constraints in the design file to place and route the design within the specified constraints. By default, this option is on.

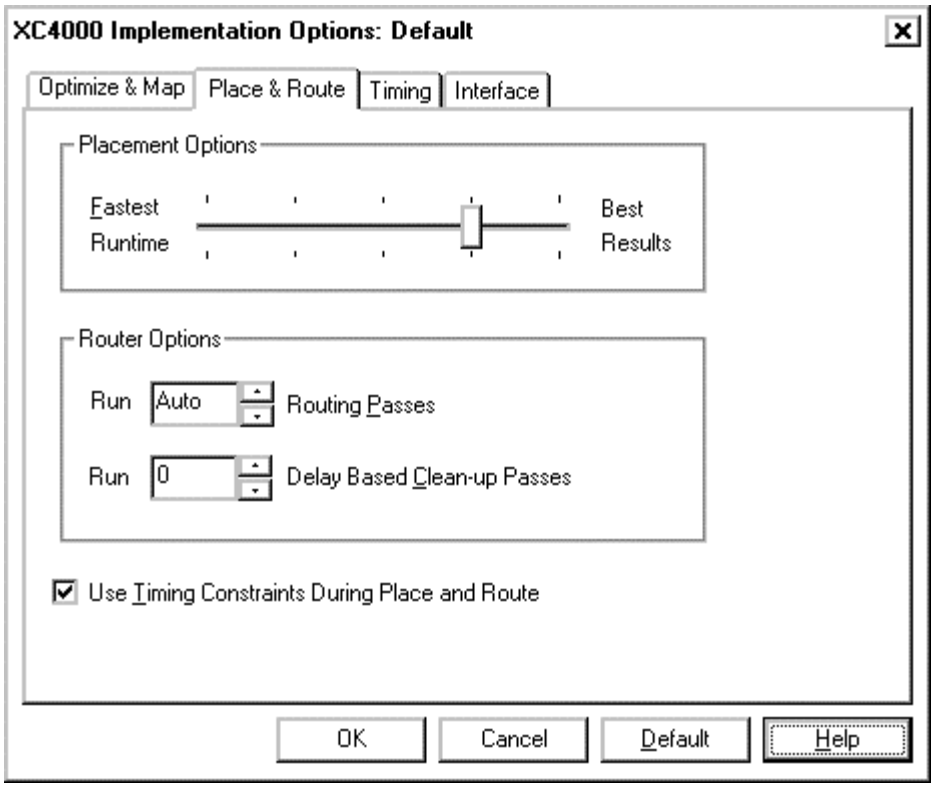

Figure 1. The Place and Route Options in the Implementation Options dialog box.

About MPPAR

The following options are available in the MPPAR dialog box (see Figure 2).

- **Starting Strategy**
  Specify a starting point (1 to100) with which to begin the place and route attempts. This number corresponds to a cost table index and different starting strategies result in different place and routes.

- **Iterations to Attempt**
  Specify the number of place and route iterations to attempt. It is important to note that each iteration will generate the identical results if the design files, constraints, implementation options, and starting strategy are all the same. So to reproduce any of the iterations, the original netlist, constraints file, implementation options and starting strategy are essential.

- **Iterations to Save**
  Specify the number of place and route iterations to save. This option compares every result to every other result and leaves the best iteration attempts. A score assigned to each revision determines the best iterations. This score takes into account such factors as the number of unrouted nets, the delays on nets and the conformance to the timing constraints. The lower the score, the better the implementation.

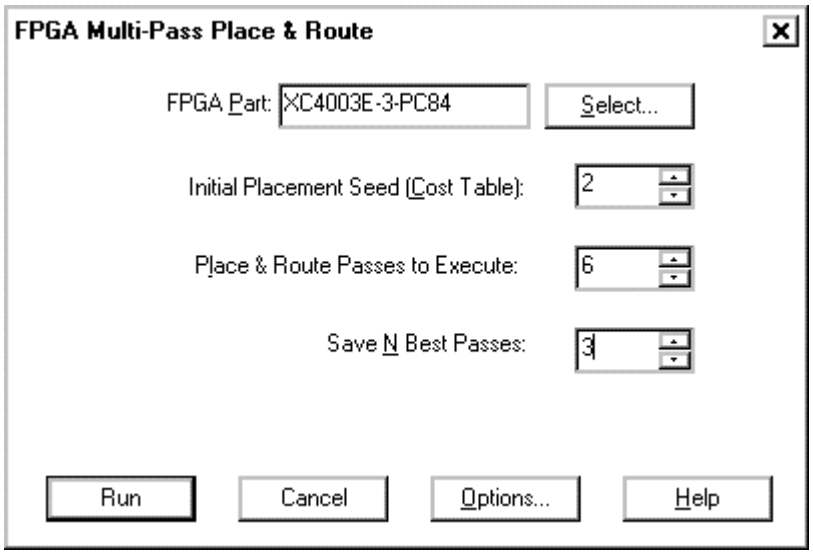

Figure 2. The MPPAR dialog box.

About the MPPR Report

After implementation, the Flow Engine creates a Multi-Pass Place and Route (MPPR) report that can be opened through the report browser. To enter the browser, click on ***Utilities>Report Browser.*** The MPPR report contains a table with the following information:

- **Level/Cost**
  Level corresponds to the revision number for this project. Cost refers to the cost table chosen for this revision.

- **Design Score**
  The design score is a weighted value given to each revision created by the MPPAR function. The lower the score the better the placement, and the closer M1 came to meeting the timing specifications. The Design Score is only relative to the revisions generated by the MPPAR tool in a single iteration. In other words, do not compare the design scores to a different multi-pass implementation.

- **Timing Score**
  This score is an indicator of how close the MPPAR function came to meeting all the timing constraints with each revision. The lower the number, the better the tool came to meeting the timing specifications. If the timing score is zero, all timing specifications were met.

- **Number Unrouted**
  This is the number of unrouted nets for each revision.

- **Run Time**
  The total run time, and the run time for each revision can be estimated by reviewing these statistics.

About Re-entrant Routing

The following options are available in the **Advanced Options** dialog box (refer to figure 3). The Re-entrant Routing option is **only recommended** to be used with designs that have trouble meeting timing constraints, or designs that do not completely place and route.

- **Implementation State**
  Use this option to update the Flow Engine to the last completed state. In this case, since the Re-entrant Routing command will be used, this should be set to “Mapped”. This will tell the Flow Engine that the revisions' mapping is satisfactory, and it can re-start the Flow Engine at the Place and Route state.

- **Allow Re-entrant Routing**

Use this option to further route an already routed design. This activates the Re-entrant Route Options section, and allows modifications to its contents.

- **Cost-based Cleanup Passes**
  Cost-based cleanup makes routing decisions by minimizing the number of switch matrices each net must use. Set the number of cost-based cleanup passes desired by choosing a number from 1 to 5. Xilinx recommends running only one cost-based cleanup on each pass, since running extra cost-based cleanup passes usually provides only minor improvement.

- **Delay-based Cleanup Passes**
  Delay-based cleanup makes routing decisions based on computed delay times between sources and loads on the routed nets, and reroutes to minimize the delays. Set the number of delay-based cleanup passes desired by choosing a number from 1 to 5. Xilinx recommends running one or two delay-based cleanups on each pass.

- **Re-entrant Routing Passes**
  Xilinx recommends running Re-entrant routing passes only on designs that have been routed to completion and not met timing constraints, or on designs that have not completely routed.

- **Use Timespecs During Re-entrant Route**
  The Flow Engine uses timing constraints in the design file to route the design within the specified constraints. If the design is using timing constraints, assert this option during re-entrant routing. Usually, this option is only turned off if the design cannot route with timing constraints.

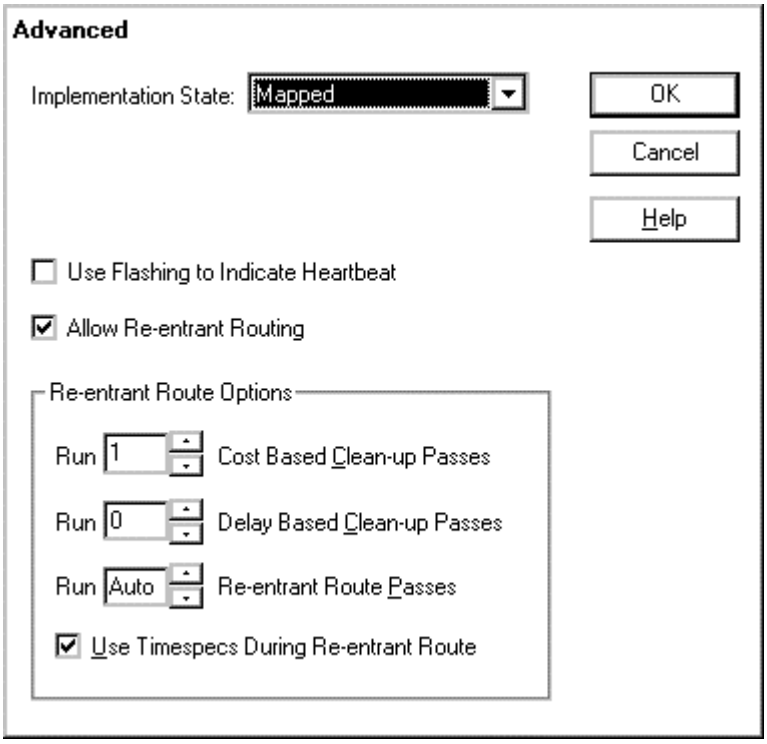

Figure 3. Advanced Options dialog box.

Design Description

Although this project resembles the FLASH project used in an earlier lab, this version of FLASH is in the MPPAR directory, and does not have any pin assignments and only has one global timing constraint (TS02=FROM:FFS:TO:FFS=8NS). This project is in the c:\m1labs\mppar directory and **should not** be confused with the FLASH project in the c:\m1labs directory (see figure 4). The MPPAR project should be implemented in an XC4003E-3 FPGA.

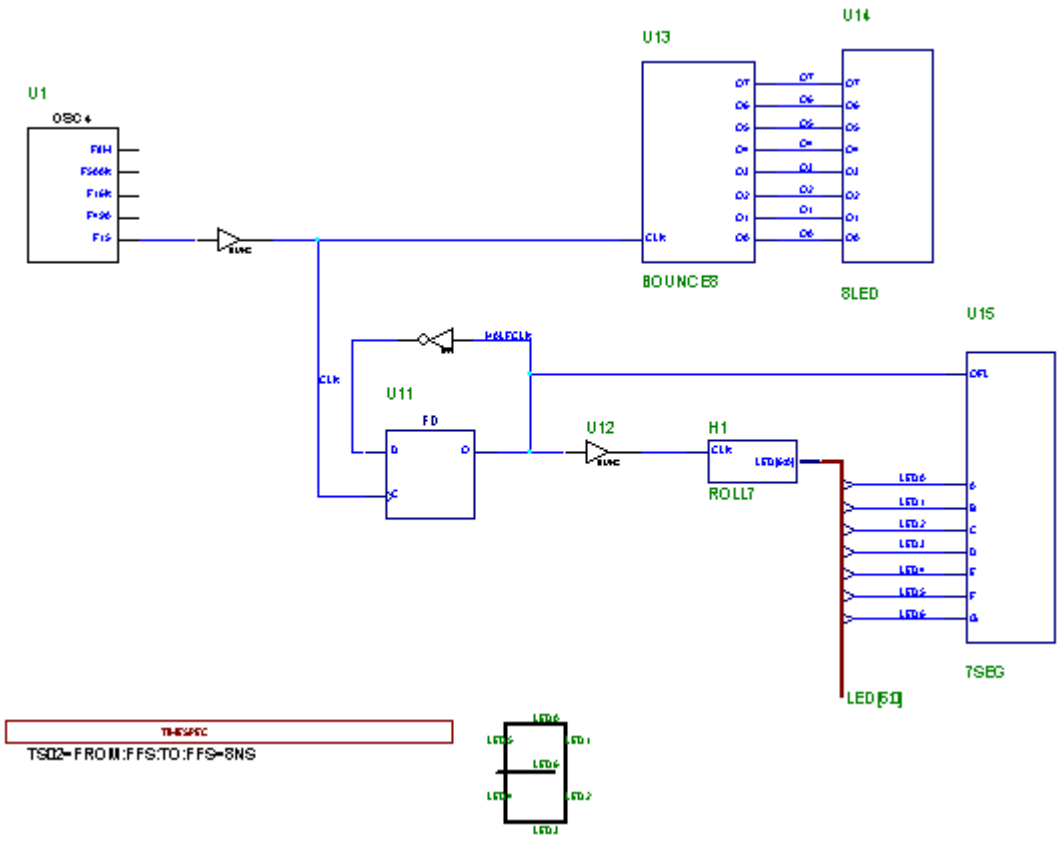

Figure 4. The NEW FLASH Project.

Procedure

Opening an existing project within Foundation

1) Open the "**Foundation Project Manager**" by clicking on ***Start>Programs>Foundation Series>Foundation Project Manager***.
2) In the "**Project Manager**" window, click on ***File>Open Project***.
3) Go to the c:\m1labs\ directory by clicking on "**..**" until the root directory is shown, and then clicking on "**mppar**" directory. Then click on the "**FLASH**" project and click on the "**Open**" button.
4) After the FLASH project has been loaded into the Foundation Project Manager, open the "**Schematic Editor**" by clicking on its icon. Review the design.

Implementing the MPPAR project in the Run-Only Flow Engine

1) While in the "**Foundation Project Manager**", click on "**M1**" button to generate an EDIF netlist and to automatically create a new project in the M1 software.
2) After the translation is complete, M1 will show the design as the current project in the "**M1 Design Manager**" window.
3) In the "**M1 Design Manager**" window, click on the left most button on the horizontal scroll bar, or click on ***Design>Implement***. The Design Manager will prompt for the correct device by bringing up the "**Implement**" window. Make certain that the **XC4003E-3-PC84** is selected.
4) Click on "**OK**", and click on the "**Run**" button. This will start the implementation process and bring up the "**Flow Engine**" window.
5) It is always important to give the M1 software a first attempt at trying to place and route the design. After all, if the design meets all the expectations, there is no reason to bother with any advanced features.
6) Enter the **Timing Analyzer** by clicking on ***Tools>Timing Analyzer>Timing Constraints*** to determine the maximum register-to-register delay.
7) Answer question #1 in the **Questions** section of this lab.

Using the MPPAR

1) Click on "**ver1**" in the M1 Design Manager, and then click on ***Design>FPGA Multi-Pass Place & Route***.
2) Once the MPPAR dialog box comes up, make certain that the "**XC4003E-3-PC84**" device has been chosen.
3) Select a Starting Strategy of 2. It is not necessary to start with a strategy of 1, since this strategy is chosen by default when the Run-Only Flow Engine was started.
4) Select 6 iterations to attempt.
5) Select 3 iterations to save.
6) Click on "**Run**".

Using the Report Browser

1) After PAR is complete, open the report browser by clicking on **Utilities>Report Browser** and double click on the MPPAR report.
2) Answer questions 2 through 4 in the **Questions** section of this lab.

Questions

1) After completing the first implementation, was the timing specification met?

2) How many different implementations were attempted?

3) How many implementations met the timing specification?

4) How can a user tell which revision will most likely have the smallest register-to-register delay?

Conclusion

This lab shows that the Multi-Pass Place and Route is the new and effective tool for improving the utilization and performance of a design. The MPPAR element enables selecting a better placement quickly. This is especially useful since optimizing placement improves performance much more than optimizing routing.

Multi-Pass Place and Route Lab Answers

1) After implementing the MPPAR project in the Run-Only Flow Engine, the timing specifications were **not** met. We took time for this step to give M1 a first chance to route the design. Note that M1 always uses cost table #1 with the first revision.

2) After using the Multi-pass Place and Route capability, seven implementations have been successfully completed. The Run-only Flow used cost table #1, and since the MPPAR tool started with cost table #2 and performed six iterations (cost tables #2 through #7), seven different implementations are now available.

3) Of the seven implementations, approximately half met the timing specifications.

4) Since the design score reflects how well the design met the timing specifications, it is likely that the revision with the lowest design score in the MPPR Report will meet the timing specification. If these scores were compared to another MPPAR run with different timing specifications or selected options, these scores would be meaningless.