Timing Constraints Lab

Timing Constraints Lab

<u>Introduction</u>

This lab uses Timing Constraints and Pin Assignments in a **User Constraints File** (**.UCF**) to improve the results of the Flash design.

The purpose of this lab is to provide students the opportunity to use a UCF file with timing constraints in their design flow. Xilinx recommends grouping logic within a schematic, while making timing constraints in a UCF file. Since pin assignments are easily made in all design entry tools, and grouping logic is best done in the schematic environment, customers will continue to have the capability to make <u>all pin assignments and many timing specifications in either a schematic, new Constraint Editor or a UCF file</u>. However, since M1 software makes it easy to modify timing specifications in a UCF, and many customers feel this is very useful.

This design is to be implemented in an XC4003E-3 device. The specifications require the design to perform at 20Mhz, but it would help if it ran faster. The design has some locked I/O pins, and there are three Timing Specifications entered (refer to Figure 1).

After a UCF with 50ns Global Timing Constraints and locked I/O pins has been created, tighten the Timing Specifications to get better performance. Reimplement the design and modify the UCF file until the best performance is found.

It is important to specify loose specs, since the compiler will need room to effectively place the design. Determine which timing constraint is limiting the designs performance and try to reduce it as much as possible to determine the system clock frequency.

<u>*About the FLASH project within the c:\F15_labs\timing directory*</u>

This FLASH project is different from the FLASH design in the Sample Designs directory C:\FNDTN\ACTIVE\PROJECTS, which is installed with the software. This version of the FLASH project is contained in the directory **`c:\F15_labs\timing`** (see **Figure 2**). It uses an external clock, uses the clock enable port of the flip-flops rather than using HALFCLK for a clock signal, and has its constraints in a UCF file. These constraints and pin locations are **not configured** for the XS40 board, and **we will not download this design** in this lab exercise.

About Timing Constraints

Both Timing and Location constraints can be entered into the top-level schematic or a UCF file. While UCF constraints take priority over duplicate constraints in a schematic, it is very important that there are no duplicate constraints in the project.

```
# This is the User Constraints File provided for FLASH
#
NET CLK PERIOD = 50 NS;
NET “U14/P_Q?” OFFSET = OUT 50 AFTER “CLKIN”;
NET “U15/*_P” OFFSET = OUT 50 AFTER “CLKIN”;
NET P_Q7 LOC = P60;           # P41 for XS40 board
NET P_Q6 LOC = P59;           # P40 for XS40 board
NET P_Q5 LOC = P58;           # P39 for XS40 board
NET P_Q4 LOC = P57;           # P38 for XS40 board
NET P_Q3 LOC = P66;           # P35 for XS40 board
NET P_Q2 LOC = P65;           # P81 for XS40 board
NET P_Q1 LOC = P62;           # P80 for XS40 board
NET P_Q0 LOC = P61;           # P10 for XS40 board
NET OFL_P LOC = P41;     # P26 for XS40 board
NET A_P LOC = P49;            # P60 for XS40 board
NET B_P LOC = P48;            # P50 for XS40 board
NET C_P LOC = P47;            # P57 for XS40 board
NET D_P LOC = P46;            # P59 for XS40 board
NET E_P LOC = P45;            # P51 for XS40 board
NET F_P LOC = P50;            # P58 for XS40 board
NET G_P LOC = P51;            # P56 for XS40 board
```

Figure 1. The FLASH.UCF for this lab.

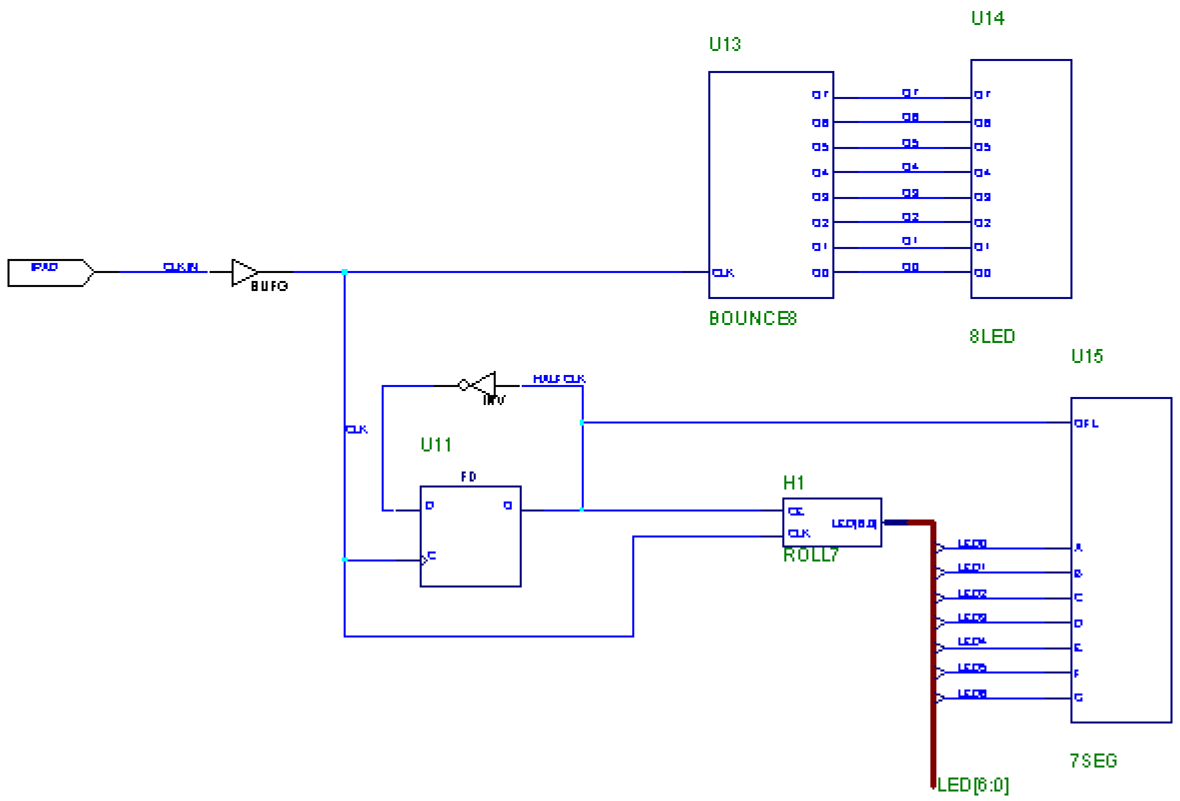

Figure 2. The FLASH project within the c:\F15_labs\timing directory.

There are two types of constraints: **global** and **path specific**. Since over constraining the project can prevent a place and route solution from being found, path specific timing constraints are the most effective way to prevent over constraining the design. However, path specific constraints, such as:

TS01=FROM:MOE:TO:LARRY=50NS;

are most effective when the designer knows the project well enough to group the logic using the Timing Name constraint (**TNM**). Global constraints are effective at quickly improving the design performance, without requiring the designer to understand the implementation very well.

Syntax remark: Note that the syntax of the **TS01** timing spec above, which is written for the schematic, includes colons (**:**). However, the timing specs in the UCF file in Figure 1 includes spaces instead of colons. This is one of three significant differences in syntax between timing specification syntax in the schematic versus in the UCF file. The other difference has to do with Bus notation. In schematics, bus taps are difficult to assign to pins, while in a UCF file, you would simply specify, for example,

```
NET DATA<1> LOC=P111;
```

In this lab, the PERIOD and OFFSET global constraints will be used to effectively improve the system clock frequency of the project with little designer effort.

NET CLK PERIOD = 50NS;

The PERIOD constraint effectively minimizes all paths between registers.

NET “U14/P_Q?” OFFSET = OUT 50 AFTER “CLKIN”;

The OFFSET constraint minimizes all paths between pads and registers, much the same as does an older constraint term called FROM:PADS:TO:FFS. However, this constraint considers the global clk buffer delay, which the older constraint does not. The AFTER in this constraint is used to specify the allowable clock-to-pad delay for the FPGA directly, while using BEFORE would specify the allowable setup requirement for the external receiving device.

This use of PERIOD and OFFSET versus FFS:TO:FFS is the third major syntax difference between UCF and schematic.

Note that the net “U14/P_Q?” refers to all outputs that are contained within the U14 symbols hierarchy, and have a node name that starts with the letters P_Q. In this example the (?) is a wildcard that can refer to any single character.

NET “U15/*_P” OFFSET = OUT 50 AFTER “CLKIN”;

Likewise, the asterisk (*) is a wildcard that can represent any number of unknown characters. In this case, the net “U15/*_P” refers to all outputs that are contained within the U15 symbols hierarchy, and have a node name that ends with _P and is any length.

Note that the net is always the node that is connected to the output or input pad.

About the Timing Report Format

The Timing Report Format options control the type of report the user would like generated by the Timing Analyzer. Since the Timing Analyzer is responsible for generating the Logic Level Timing Report, these report options also apply to it as well.

- **Post Layout Timing Report**
  This report indicates whether the timing constraints are being met and the maximum frequency for the clocks. It also indicates if errors exist.

- **Report Paths Using Advanced Design Analysis (No Timing Constraints)**
  This report calculates the path delays for each of the default timing constraints: flip-flop to flip-flop, pad to flip-flop, and flip-flop to pad. The paths are listed in decreasing order of length, and is limited by the number the designer specifies with the Limit Report to_Paths per Timing Constraint option.

- **Report Paths in Timing Constraints**
  This report calculates the path delays for each of the timing constraints specified in the projects UCF file.

- **Report Paths Failing Timing Constraints**
  Select this option to generate an error report that lists timing errors and associated net/path delay information. If a constraint is not met, the report gives the number of items scored, the number of errors encountered, and a detailed breakdown of the error.

- **Limit Report to ____ Paths per Timing Constraint**
  Use this option to set the maximum number of reported paths for each timing constraint. The report displays worst case paths. Choose No Limit or a value from 1 to 10. The default setting is 10. This option applies to all report formats.

About the Logic Level Timing Report

The Logic Level Timing Report is used to determine if the timing constraints entered into the project are reasonable, without having to wait for the design to place and route. This option produces a timing report prior to place and route but after map, and contains no net delays. The generated timing report provides a summary analysis of the timing constraints based only on the block delays.

The "**Logic Level Timing Report**" generated by the Flow Engine only considers block delays and estimated net delays. It is very important that new designers keep in mind that worst case net delays are approximately equal to the estimated block delays. This means that the reported block delay in the Logic Level Timing Report, should be doubled to determine if the timing specification is reasonable.

For example, if Figure 4 is the **Logic Level Timing Report**, the 20.876ns delay is the total estimated delay for the FFS to PADS timing constraint. Since the report contains estimated net delays, one can still assume that the net delays could be as high as the block delays. Since the logic delays are 84.5% of the total delay, logic accounts for 17.7ns of the total delay. Since this is the longest delay path, the timing constraints on the first implementation should be no less than 35.4ns. From this report, it is apparent that the Flow Engine will not have any trouble meeting a 50ns timing specification. By implementing the design with a 35.4ns timing constraint, a user can save time trying to get an unrealistic timing constraint met by the flow engine.

```
===========================================================================
Timing constraint: Default period analysis
 200 items analyzed, 0 timing errors detected.
 Minimum period is   6.952ns.
 Maximum delay is  20.876ns.
---------------------------------------------------------------------------
Delay:    20.876ns H1/HEX2 to U15/B_P

Path H1/HEX2 to U15/B_P contains 4 levels of logic:
Path starting from Comp: CLB.K (from CLK)
To                   Delay type         Delay(ns)  Physical Resource
                                                   Logical Resource(s)
-------------------------------------------------  --------
CLB.YQ               Tcko                  2.820R  H1/HEX2
                                                   H1/HEX3
CLB.G2               net (fanout=17)    e  1.082R  H1/HEX3
CLB.Y                Tilo                  2.000R  H1/U16/S
                                                   H1/U16/U
CLB.F2               net (fanout=1)     e  1.082R  H1/U16/U
CLB.X                Tiho                  4.310R  H1/U16/F
                                                   H1/U16/F/2.0
                                                   H1/U16/F
IOB.O                net (fanout=1)     e  1.082R  H1/U16/F
IOB.PAD              Tops                  8.500R  U15/B_P
                                                   U15/B_P.OUTBUF
                                                   U15/B_P
-------------------------------------------------
Total (17.630ns logic, 3.246ns route)     20.876ns
      (84.5% logic, 15.5%% route)
```

Figure 3. The Logic Level Timing Report in the 'Report Paths in Default Timing Constraints' format.

After the Flow Engine finished compiling, the Flow Engine generated the "**Post Layout Timing Report**" report given in Figure 5. From this information, it is evident that:

- The timing specification of 50ns was easily attainable.
- Approximately 39% of the 28.9ns delay was due to routing, unlike the estimation of 15.5%.
- If the timing specification was reduced to 35.4ns the flow engine would not have had any trouble meeting that timing specification.

Note that the reports shown in figures 4 and 5, were generated by selecting the "**Report Paths in Default Timing Constraints**" report format.

```
===========================================================================
Timing constraint: COMP "U15/F_P" OFFSET = OUT 50.000 nS AFTER COMP "CLKIN" ;
 20 items analyzed, 0 timing errors detected.
 Minimum allowable offset is  33.041ns.
---------------------------------------------------------------------------
```

```
Slack:    16.959ns path CLKIN to H1/HEX0 relative to
          28.943ns delay constraint H1/HEX0 to U15/F_P and
          50.000ns offset CLKIN to U15/F_P


Data path H1/HEX0 to U15/F_P contains 4 levels of logic:
Path starting from Comp: CLB_R8C6.K (from CLK)
To                    Delay type          Delay(ns)  Physical Resource
                                                     Logical Resource(s)
-------------------------------------------------    --------
CLB_R8C6.XQ           Tcko                   2.820R  H1/HEX0
                                                     H1/HEX1
CLB_R10C7.F3          net (fanout=18)        3.808R  H1/HEX1
CLB_R10C7.X           Tilo                   2.000R  H1/U16/I
                                                     H1/U16/I
CLB_R7C8.F4           net (fanout=4)         4.816R  H1/U16/I
CLB_R7C8.X            Tiho                   4.310R  H1/U16/B
                                                     H1/U16/B/2.0
                                                     H1/U16/B
P50.O                 net (fanout=1)         2.689R  H1/U16/B
P50.PAD               Tops                   8.500R  U15/F_P
                                                     U15/F_P.OUTBUF
                                                     U15/F_P
-------------------------------------------------
Total (17.630ns logic, 11.313ns route)    28.943ns
      (60.9% logic, 39.1%% route)
```

Figure 5. Section of the Post Layout Timing Report generated by the Flow Engine.

Procedure

Opening an existing project within Foundation

1) Open the "**Foundation Project Manager**" by clicking on **Start→Programs→Xilinx Foundation Series→Xilinx Foundation Project Manager**.

2) In the "**Project Manager**" window, click on **File→Open Project**.

3) Go to the **`c:\F15_labs\timing`** directory within Foundation and click on the "**FLASH**" project, and click on the "**Open**" button.

4) In the "**Foundation Project Manager**" window, open the "**Schematic Editor**" by clicking on its icon.

Editing a UCF file in the Foundation Project Manager

5) In the “**Foundation Project Manager**”, click on “**FLASH.UCF**” to open the User Constraints File. Once the editor is open, changes can be made to this file.

6) To save any changes to this UCF file, click on **File→Save**.

7) Close the UCF file editor.

Implementing a project within M1

8) While in the “**Foundation Project Manager**”, click on the “**M1**” button to generate an EDIF netlist and to automatically create a new project in the M1 software.

9) After the translation is complete, M1 will show the design as the current project in the “**M1 Design Manager**” window.

10) 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 sure that the XC4003E-3-PC84 is selected.

11) The **FLASH.UCF** file needs to be specified as the Constraints file within M1. Click on the “**Options**” button within the “**Implement**” window that was just brought up. At the top of the window, a box to enter the “**User Control File**” name will be seen. Browse for the UCF file that was created. When it is found, highlight it and click on the “**Open**” button. The file will now appear in the User window.

12) Click on **Implentation: Edit Templates → Timing Reports →Produce Logic Level Timing Report**, to determine if the timing specifications are realistic.

13) Click on “**Produce Post Layout Timing Report**” to obtain a summary report of the actual timing results.

14) Choose the “Report Paths in Timing Constraints” report format. Also, select the “**Limit Report to_Paths per Timing Constraint**” option to “**1**”.
Click on the “**Place & Route**” tab and set the effort level slider bar to 4 out of 5, near the Best Results side. Set “**Routing Passes**” to Auto, and “**Delay Based Clean-up Passes**” to 0, and make sure Use “**Timing Constraints During Place and Route**” is checked.

15) Click on “**OK**”, and click on the “**Run**” button within the “**Implement**” window. This will start the implementation process and bring up the “**Flow Engine**” window.

16) While the Design is being processed, monitor the implementation progress through the Flow Engine's Message window at the bottom of the "**Flow Engine**". While the Place&Route Engine is running, review the "**Logic Level Timing Report**" by clicking on **Utilities→Report Browser→Logic Level Timing Report.**

17) After the **FLASH** project has been implemented, the Log File can be reviewed, which contains all of the messages that were shown in the Flow Engine Message window during implementation. After reviewing the file, click on the "**OK**" button.

Using the Timing Analyzer

18) To determine if the implementation met the timing specifications, a simple method is to select the "**Product Post Layout Timing Report**" prior to place and route. Figure 5 was generated from this report.
You can obtain the same information using the Timing Analyzer. In addition, Timing Analyzer allows you to explore many different "what if" timing options, such as use of different speed grades, filtering out portions of the design for analysis, or analyzing specified portions of your design.

(Optional)

19) We'll use the Timing Analyzer for now. Click on **Tools→Timing Analyzer** . Clicking on the "**Timing Analyzer**" icon on the Vertical Tool Bar will also start it.

20) In the "**Timing Analyzer**" window, click on the **Analyze→Timing Constraints**, or click on the "**? TC**" icon on the horizontal tool bar. The Timing Analyzer will then open up the report. Enter the longest path delay for each timing constraint into the chart in the Questions section of the lab.

21) It is important to remember that any projects system clock frequency is determined by the longest of three different delays: Pad to Flip-Flop, Flip-Flop to Flip-Flop, and Flip Flop-to Pad. Whichever of these delays is longer will be used to calculate the project's system clock frequency. Enter the calculated system clock frequency into the chart in the **Questions** portion of the lab.

22) After reviewing these reports, answer the remaining questions in the lab.

23) After completing the questions, exit the **M1 Design Manager**.

Questions

	PERIOD constraint	**U14/P_Q? constraint**	**U15/*_P constraint**	**System Clock Frequency**
1st TimeSpec	50 ns	50 ns	50 ns	20 Mhz
1st Iteration				
2nd TimeSpec	34 ns	34 ns	34 ns	29 Mhz
2nd Iteration				
3rd TimeSpec	29 ns	29 ns	29 ns	37 Mhz
3rd Iteration				

1) Were the PERIOD and OFFSET timing specifications met?

2) Which timing specification is going to limit the system clock frequency and how much is this delay?

3) Try replacing the timing specifications in the UCF file with the timing specifications given in the chart for the second and third implementations. After each modification of the UCF, reimplement the project to see if better performance can be achieved. Complete the chart after each successive change to the UCF.

Conclusion

In this lab, Timing Constraints are shown as the primary way of communicating a performance expectation to the M1 Development System. Even though it is not the only way, it is one of the most commonly used techniques to get the best performance and utilization. Xilinx recommends that timing constraints be entered in a User Constraints File. Following this design flow allows the designer to keep all timing specifications in a single independent file that can be edited outside of the design file. While the PERIOD and OFFSET specifications cannot be entered in a schematic design file, these and all other constraints can be entered in a UCF or schematic file.

Answers (Note: Results may vary slightly)

1) Yes, all Timing Specifications provided were met on the first iteration.

2) U15/*_P, the clock to output time for the U15 macro is going to limit the FLASH designs performance. This is because it has the longest delay path. If the delay path could be lowered significantly, improvements in the circuits' performance could be realized. To determine the longest delay path, review the "**Design Performance**" report in the Timing Analyzer. From this report, it is apparent that the longest delay path leaves a flip-flop, and goes to an output pin. After the first iteration, the longest path constrained by "U15/*_P" was approximately 35 ns, which is used to calculate a maximum clock frequency of approximately 29 Mhz. Your results may vary.

	PERIOD constraint	**U14/P_Q? constraint**	**U15/*_P constraint**	**System Clock Frequency**
1st TimeSpec	50 ns	50 ns	50 ns	20 Mhz
1st Iteration	~ 10 ns	~ 18 ns	~ 35 ns	~ 29 Mhz
2nd TimeSpec	34 ns	34 ns	34 ns	29 Mhz
2nd Iteration	~ 10 ns	~ 15 ns	~ 30 ns	~ 34 Mhz
3rd TimeSpec	29 ns	29 ns	29 ns	34 Mhz
3rd Iteration	~ 10 ns	~ 15 ns	~ 29 ns	~ 36 Mhz

3) The performance improved successively because the Timing Constraints were much tighter. In the first iteration, the compiler was told that all delays could be very large and the compiler chose to make the longest U15/*_P path approximately 35ns long, even though it could have made it significantly smaller. Likewise, the other constraints have plenty of room to meet the designers needs, because they do not require as many levels of logic as the U15/*_P paths. When the Timing Constraints were tightened for the second iteration, it was found that there was still a great deal of room for the PERIOD and U14/P_Q paths to improve. They both improved because the M1 flow engine was given a tighter timing constraint. On the last iteration, all of the delay paths barely improved since the design was near the optimal solution for this pin placement.