Summary

In this application note we will discuss how to constrain a CPLD design and how to verify that the design has met timing. Fundamentally, CPLD timing is the same as FPGA timing; however, the CPLD timing constraints are a subset of the FPGA timing constraints. For further information on the limitations of CPLD timing constraints see AR #23562.

Introduction

Constraints

There are four types of constraints available in a CPLD design.

- The PERIOD Constraint
- The OFFSET Constraint
- The Pad-to-Pad Constraint
- The FROM-TO Constraint

All timing constraints constrain paths between either pads or synchronous elements. Pads are defined as input or output pins, and synchronous elements are defined as any clocked elements (for example, registers) in the design. Even though they are not strictly clocked, it is worth noting that that latches are considered to be synchronous elements by the timing analysis tools.

PERIOD constraint

The PERIOD constraint covers paths between synchronous elements clocked by a global clock net.

The PERIOD constraint does not optimize combinatorial paths from input pads to output pads, paths from input pads to synchronous elements, or paths from synchronous elements to output pads.

OFFSET constraint

The OFFSET constraint covers paths FROM input pads TO synchronous elements (OFFSET IN) and FROM synchronous elements TO output pads (OFFSET OUT).

Pad-to-Pad Constraint

The Pad-to-Pad constraint covers paths that start at and end at I/O pads. These paths may not contain any synchronous elements.

FROM-TO constraint

The FROM-TO constraint defines a timing constraint between two specific groups. A group can be user-defined or predefined. For synchronous paths, a FROM-TO constraint controls only the setup path, not the hold path. The setup time is the time before the clock edge that the data must be valid, and the hold time is the time after the clock edge that data must remain valid.

The predefined groups available for CPLD users are FFS (flip-flops or registers), LATCHES and PADS. User defined groups can be created using the TNM (Timing Name) constraint.

© 2008 Xilinx, Inc. All rights reserved. XILINX, the Xilinx logo, and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.
Adding Timing Constraints

Figure 1 shows the paths covered by each constraint.

**Figure 1: Timing Paths**

The delay from ADATA to FLOP1 is covered by an OFFSET IN constraint. The delay from BUS to FLOP4 is also covered by an OFFSET IN constraint. The PERIOD constraint covers the delay from FLOP1 to FLOP2, from FLOP2 to FLOP3 and from FLOP4 to FLOPS. The delay from FLOP3 to OUT1 and FLOP5 to OUT2 is covered by an OFFSET OUT constraint. The delay from CDATA to OUT2 is covered by a Pad-to-Pad constraint.

In order to explain the concepts more thoroughly we will apply CPLD timing constraints to a real design and discuss how to analyze the timing. The design used in this example is taken from XAPP387: PicoBlaze Microcontroller. The target device is XC2C256-6TQ144. The zip file for the application note contains a sample project in the demo_test directory.

Adding Timing Constraints

You can write the timing constraints by hand in the UCF or by using the Constraints Editor GUI.

**Writing the Timing Constraint Manually in the UCF**

There is only one clock net in this design which is called clk. The PERIOD constraint and OFFSET constraints reference this clock. The PERIOD constraint is used to constrain the delay from register to register inside the device.

The PERIOD is dependant on the clock frequency, for example, if the input clock is 100 MHz the PERIOD constraint is 10 ns. The timing tools can interpret the PERIOD either in units of time or frequency. The OFFSET constraint covers the delay from input pins to registers inside the device or the delay from the internal registers to the output pins.

The code in UCF is below :

```ucf
NET "clk" BUFG=CLK; # this tells the tools to assign 'clk' to a global clock buffer
NET "clk" TNM_NET = "clk"; # gives the net clk a group name as clk
TIMESPEC "TS_clk" = PERIOD "clk" 10 ns ; # add 10 ns PERIOD constraint the on clk time group, this constraint name is TS_clk.
OFFSET = IN 8 ns BEFORE "clk" ; # add 8ns offset in constraint
OFFSET = OUT 8 ns AFTER "clk" ; # add 8ns offset out constraint
TIMESPEC "TS_P2P" = FROM "PADS" TO "PADS" 10 ns; # PADS is a predefined group, meaning any I/O pad. PADS to PADS constraint is 10 ns.
```
If the OFFSET constraints are created as above then it is a global constraint that will be apply to all I/Os. If you wish to place an OFFSET constraint on individual or groups of instances or nets this can be done using a TNM and TIMEGRP constraints. Firstly the group is created using the TNM constraint. Then place the OFFSET on that specific group.

For example if all outputs had a requirement of 8 ns with the exception of two outputs that had a tighter requirement of 7.5 ns, you could group the two outputs and place the TIMEGRP with OFFSET constraint on them. For example:

```
INST "output<1>" TNM = "out_grp";
INST "output<0>" TNM = "out_grp";
TIMEGRP "out_grp" OFFSET = OUT 7.5 ns AFTER "clk" ;
```

**Use Constraints Editor to Create the Constraints in the GUI**

Double click the User Constraints → Create Timing Constraints in the process window. In the Xilinx Constraints Editor, select the Global tab and enter 10 in the Period window (it is assumed to be nS units), this will create your period constraint. The CPLD tools do not support duty cycles other than 50/50. Under Pad to Setup enter 8; this creates the OFFSET IN constraint. Under Clock to Pad enter 8; this creates the OFFSET OUT. Finally, add 10 for the Pad to Pad constraint.
Reading the Timing Report

Create the Timing Report.
1. In the Project Navigator GUI right click on Generate Timing under Optional Implementation Tools and select Properties. Double check to ensure that the Timing Report Format is set to Summary.


There are four parts in the timing report:
- Performance Summary Report
- Timing Constraint Summary
- Data Sheet Report
- Path Type Definition: The acronyms in the Performance Summary section of the report are defined in this section of the timing report.

Performance Summary Report
Contains information on the device used and the software version.

Timing Constraint Summary
This section is important as it reports if your design has met timing. The OFFSET constraints get renamed to ‘Auto_TS_200x’ because we did not assign them a Time Spec value in the UCF. Here is an example of constraints that have met timing.

Table 1: Timing Constraint Summary

<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Requirement (ns)</th>
<th>Delay (ns)</th>
<th>Paths</th>
<th>Paths Failing</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS_clk</td>
<td>10.0</td>
<td>10.2</td>
<td>619</td>
<td>48</td>
</tr>
<tr>
<td>TS_P2P</td>
<td>10.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AUTO_TS_2001</td>
<td>8.0</td>
<td>4.5</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>AUTO_TS_2002</td>
<td>9.8</td>
<td>4.4</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Note: If the constraint has the status N/A or 0, there were no paths of that type in the design for the software tools to analyze

Note: The OFFSET IN constraint is adjusted by tGCK, meaning the delay on tGCK is added to the requirement. The requirement in this case is 8 ns and tGCK is 1.8 ns, therefore the requirement used in the analysis is 9.8 ns.
If your design does not meet timing, there are a number of options available. You can change the optimization template from density to speed. When the template is set to density, the design partitioning and placement result in a slower speed, but uses resource sharing to allow more logic to fit into a device. Optimizing for speed uses less resource sharing, but flattens the logic, which results in fewer levels of logic (faster). Optimize for density is the default setting.

Alternatively, you can target a faster speed grade device. Finally, you can modify the design to balance combinatorial logic between registered elements by inserting extra registers to achieve pipelining. Changing the design will have the biggest effect on timing. XST has a separate optimization setting which has some impact on final results, although usually not as big an impact as the optimization performed by the fitter. The best strategy depends on how much you miss timing. For example, if you miss timing by 3-5 ns, changing options may be sufficient. However, if you miss timing by a greater margin (for example 10 ns), you probably need to redesign or investigate more carefully the cause of problem.

**Table 2** shows results with **density** (default) selected as the optimization strategy:

**Table 2: Results with Density**

<table>
<thead>
<tr>
<th>Performance Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. Clock Period</td>
<td>10.200 ns</td>
</tr>
<tr>
<td>Max. Clock Frequency (fSYSTEM)</td>
<td>98.039 MHz</td>
</tr>
<tr>
<td><strong>Limited by Cycle Time for clk</strong></td>
<td></td>
</tr>
<tr>
<td>Clock to Setup (tCYC)</td>
<td>10.200 ns</td>
</tr>
<tr>
<td>Setup to Clock at the Pad (tSU)</td>
<td>2.600 ns</td>
</tr>
<tr>
<td>Clock to Pad to Output Pad Delay (tCO)</td>
<td>4.500 ns</td>
</tr>
</tbody>
</table>

**Note:** The report contains a parameter tCYC which is the Clock to Setup register to register cycle time. It includes source register clock-to out (tCO), routing delays (tF), logic delays (tLOGI), and destination register setup time (tSU). The maximum clock speed calculated uses tCYC, and is therefore limited by the cycle time.

The internal clock speed defines the maximum frequency for clocking the internal logic. In the text report, the external maximum frequency is also reported. It is defined as the maximum frequency at which the I/Os can be clocked. **Table 3** shows results with **speed** selected as the optimization strategy:

**Table 3: Results with Speed**

<table>
<thead>
<tr>
<th>Performance Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. Clock Period</td>
<td>7.200 ns</td>
</tr>
<tr>
<td>Max. Clock Frequency (fSYSTEM)</td>
<td>138.889 MHz</td>
</tr>
<tr>
<td><strong>Limited by Cycle Time for clk</strong></td>
<td></td>
</tr>
<tr>
<td>Clock to Setup (tCYC)</td>
<td>7.200 ns</td>
</tr>
<tr>
<td>Setup to Clock at the Pad (tSU)</td>
<td>2.400 ns</td>
</tr>
<tr>
<td>Clock Pad to Output Pad Delay (tCO)</td>
<td>4.500 ns</td>
</tr>
</tbody>
</table>

With **speed** selected as the optimization strategy the frequency increased from 98 MHz to 138 MHz, a 40 MHz improvement! This, however did not come without cost to area. The density optimized design used 85 macrocells and 315 product terms, and the speed optimized design used 99 macrocells and 400 product terms. This demonstrates an increase in area due to logic duplication and flattening, but shows the benefits are from a performance perspective.
The Data Sheet Report shows the maximum speeds for the design and contains details on the worst case paths. There are two parts in the performance section.

- Summary of maximum speed this design can run, as seen in Table 3 above
- Detailed net timing information

In the text based report, the source of the analyzed path is at the top of the table and the destination is on the left.

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>address&lt;4&gt;Q</td>
<td>address&lt;4&gt;D</td>
<td>3.900</td>
</tr>
<tr>
<td>address&lt;4&gt;Q</td>
<td>address&lt;5&gt;D</td>
<td>4.200</td>
</tr>
<tr>
<td>address&lt;4&gt;Q</td>
<td>address&lt;6&gt;D</td>
<td>6.900</td>
</tr>
<tr>
<td>address&lt;4&gt;Q</td>
<td>address&lt;7&gt;D</td>
<td>6.900</td>
</tr>
</tbody>
</table>
Using Xilinx Timing Analyzer to View the Detailed Trace Delays

1. In Project Navigator double click on **Analyze Post-Fit Static Timing**:

![Figure 5: ISE Process Tree](image)

2. In the **Timing Analyzer**, click **Analyze → Analyze Timing Constraint**. This will create a timing report for your design.

The left part of timing report is the summary of the report. Any constraints that fail to meet the timing constraint will be highlighted in red. The delay numbers used in the timing report can be found in the device density specific data sheet.

Reading the Timing Analyzer Report

For example, the period is constrained to 10 ns. Here is an example of one path covered by this constraint. The parameters shown in the timing reports can be found in device datasheet with further details in the timing model application note. See [xapp375](#).

```
==================================================================================================
Timing constraint: TS clk: PERIOD clk: 10 000ns: HIGH 5 000ns

Slack: 6.100ns (requirement - data path)
Source: address(0, Q)
Destination: instruction(11), D
Requirement: 10.000ns
Data Path Delay: 3.900ns (Levels of Logic = 3)

Data Path: address(0), 0 to instruction(11), D
Delay type Delay(s) Logical Resource(s)
tODI 0.400 address(0)
tP & tLOGO + tSOI 2.500 instruction(11), D
```

![Figure 6: Analyzer Report](image)
The path analyzed by the constraint is shown in red in Figure 7.

Figure 7: Timing Path Analysis

You can check in the data sheet of XC2C256 (ds094) for the numbers used in the timing analysis.

\[ T_{COI} = 0.4; \ T_F = 1.7; \ T_{LOGI1} = 0.5; \ T_{SUI} = 1.3 \]

Table 5 shows the values for -6 speed grade

Table 5: Values for -6 Speed Grade

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T_{COI})</td>
<td>Clock to Output Valid</td>
<td>0.4</td>
</tr>
<tr>
<td>(T_F)</td>
<td>Feedback Delay</td>
<td>1.7</td>
</tr>
<tr>
<td>(T_{LOGI1})</td>
<td>Single P-term Delay Adder</td>
<td>0.5</td>
</tr>
<tr>
<td>(T_{SUI})</td>
<td>Setup Before Clock</td>
<td>1.3</td>
</tr>
</tbody>
</table>
OFFSET IN

As in the timing report, the OFFSET constraint is adjusted by tGCK in the Timing Analyzer.

Figure 8: OFFSET Constraint Adjusted

The path analyzed by the constraint is shown in red in the below diagram and the GCK path is in blue:

You can check in the data sheet of xc2c256 (ds094) for the numbers used in the timing analysis. Shown in table are the values for -6 speed grade

Table 6: Values for -6 Speed Grade

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_IN</td>
<td>Input Buffer Delay</td>
<td>2.4</td>
</tr>
<tr>
<td>T_F</td>
<td>Feedback Delay</td>
<td>1.7</td>
</tr>
<tr>
<td>T_LOG11</td>
<td>Single P-term Delay Adder</td>
<td>0.5</td>
</tr>
<tr>
<td>T_SUI</td>
<td>Setup Before Clock</td>
<td>1.3</td>
</tr>
</tbody>
</table>

Conclusion

The Timing Report is good tool to check to see if your design meets its timing requirements. When it does not meet timing requirements, it is an excellent tool to find out why. For CPLD designs, Xilinx recommends that users add timing constraints to their design and verify them in the Timing Report, as well as through simulation. This practice will minimize the need for board level debugging.
The following table shows the revision history for this document:

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description of Revisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>02/07/08</td>
<td>1.0</td>
<td>Initial Xilinx release.</td>
</tr>
</tbody>
</table>

Xilinx is disclosing this Application Note to you “AS-IS” with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Xilinx. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. XILINX MAKES NO REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL XILINX BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM YOUR USE OF THIS APPLICATION NOTE.