

EMA TimingDesigner

The industry's most accurate static timing analysis

Digital Timing

Using TimingDesigner to Generate SDC Timing Constraints

Learn about:

- Using TimingDesigner to generate SDC for development of FPGA designs
- Using TimingDesigner to establish necessary signal adjustments for interface design
- Validating FPGA interface designs by importing delay results to update waveform relationships

Revision:

- Revised: 08/03/2008
- For use with: TimingDesigner 9.1 and higher

Jerry Long – Product Marketing Manager, EMA Design Automation

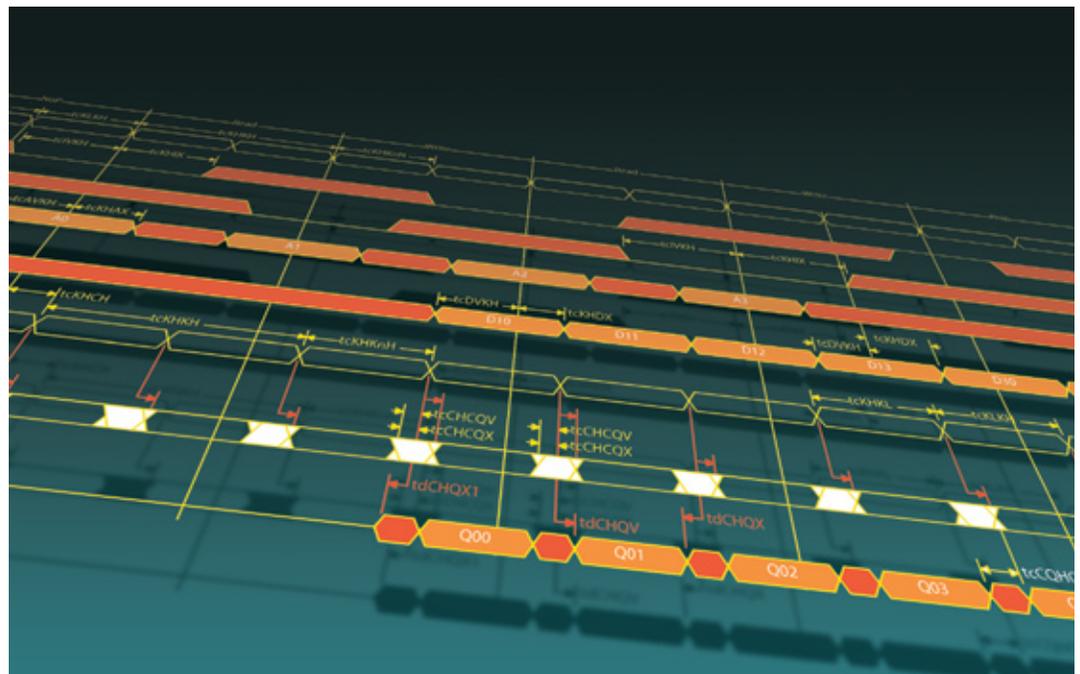


Table of Contents

Introduction	3
Complex Interface Timing Challenges.....	4
Constraining Designs.....	5
Constraint Complexities	5
Creating Constraints with TimingDesigner.....	6
Input Delay Representation.....	6
Output Delay Representation.....	7
Extracting Post-fit Delay Timing.....	7
A Two Pass Fitter Approach	7
Summary.....	8
Where to Go for Help	9
Support.....	9
Support Contact Information.....	9
General Contact Information	9

Table of Figures

Figure 1 - TimingDesigner's GUI windows allow easy capture of design interface characteristics	4
Figure 2 - Examples of center-aligned (top) and edge-aligned clock/data relationships	4
Figure 3 - Example of a DDR center-aligned data input interface to generate SDC design constraints	6
Figure 4 - Example of a DDR center-aligned data output interface to generate SDC design constraints.....	7
Figure 5 - Example of determining clock shift to achieve a balanced clock/data relationship.....	8

Introduction

“Timing issues previously deemed insignificant are now impacting design schedules and can no longer be dealt with after the fact.”

As technology advances, so do the complexity of the problems it exposes. Nowhere is this more evident than in high-speed interface design. Timing issues previously deemed insignificant are now impacting design schedules and can no longer be dealt with after the fact. Design innovations such as double-data rate memory devices (DDR, DDRII, & DDRIII), with their source synchronous capabilities and continuing speed improvements, have increased the impact static timing issues have in resolving high-speed system interface operation. Margins for data setup and hold requirements are tight which leaves minimal room to secure an accurate data capture and presentation window. Faster edge rates also magnify physical design effects, which cause signal integrity issues that require additional settling time, shrinking timing margins further.

FPGA manufacturers are keeping pace with devices that are extremely register rich and offer advanced I/O features that directly support these high-speed interface protocols. In addition, they provide intricate timing control capability with fully programmable phase-locked loop networks. These features allow design of memory controllers, data exchangers, and pipeline networks, with full clock/data relationship control, dynamic termination, and I/O technology networks to comply with the latest interface styles available. While these new advances in FPGA technology

are of tremendous help, they don't alleviate the importance of static timing, which if not monitored can render entire designs useless. To take full advantage of these highperformance features, you still must analyze your available timing options, incorporate signal integrity and other physical delay effects, and do so throughout the design process.

TimingDesigner® from EMA Design Automation is considered the industry standard timing analysis tool to aid in complex interface design and development as it provides an easy, selfintuitive method to address static timing issues using interactive timing diagrams. It is ideal for high-speed, multi-frequency designs where it is essential to accurately model and analyze signal relationships between devices on a board or between embedded functions on an ASIC or FPGA. It can evaluate comprehensive sets of timing alternatives and provide direction to the most complex of timing challenges, enabling designers to manage and monitor timing margins throughout the design process. In addition, TimingDesigner can generate place and route constraints in SDC format that reference design specific timing measurements, and allows direct use of post place-and-route timing information for signal delays providing visual verification of the interface signal relationships required for desired FPGA interface operation.

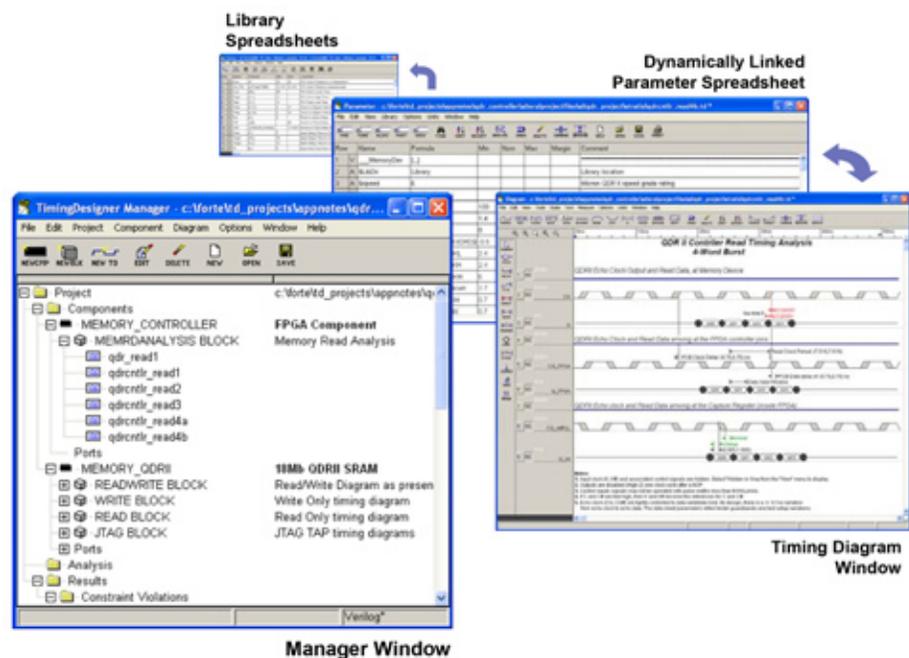


Figure 1 - TimingDesigner's GUI windows allow easy capture of design interface characteristics

Complex Interface Timing Challenges

“While SDC does an effective job, there is still the complexity factor of implementing constraints correctly so that timing relationships are accurately represented.”

Complex interface design presents many timing challenges, most of which aren't really new. Each device has unique data/clock relationships (i.e. edge-aligned or center-aligned) and timing requirements (i.e. input setup, input hold, and clock-to-out) for successful data operations. In order for an interface to effectively communicate

data, these protocol and timing requirements must be adhered to. For FPGA and ASIC designs, these operational requirements can be referenced through the effective implementation of design constraints, the most common methodology being SDC.



Figure 2 - Examples of center-aligned (top) and edge-aligned clock/data relationships

Constraining Designs

SDC format became an open source language in 1999 for constraining complex design descriptions through synthesis and as a result, became the defacto industry standard for ASIC and some FPGA design flows. While SDC does an effective job, there is still the complexity factor of implementing constraints correctly so that timing relationships are accurately represented. Add to that the fact that it's mostly a manual process and you end up with a constraint system that is very powerful, but at the same time susceptible to application errors, and a source of confusion for correct timing analysis results. Supplying constraint information for a design basically falls into three categories of constraint commands: Clock constraints, Data constraints, and Timing Exceptions.

Clock constraints are the SDC commands that specify the characteristics of all clocks used in the design. They include all characteristics of the input clock(s) (or main clock driving the design), all internally generated clock settings that are essentially derivatives of the input clock(s), and any clock(s) that are to be driven out of the device for things like source synchronous interfaces. Creating clock constraints is relatively straight forward, with minimal complexity involved, and can be used in the design

implementation phase for FPGAs and ASICs as they define how certain clocks are to be treated and influence their creation.

Data constraints give relational positioning information of data signals, or absolute skew requirements for any incoming or outgoing data signals for the device and are typically referenced to a clock source. They are sometimes used in the implementation phase, depending on the complexity of the routing engines (i.e. if they take advantage of timing driven place-and-route techniques), and describe the 'boundaries' for expected fitting and routing delays. Data constraints can be quite complex due to the nature of the interface under design.

Timing exceptions are mainly to control a timing analysis engine, incorporated after the implementation phase of a design, so that the correct analysis results are calculated. These include false path controls for restricting the analysis to appropriate data paths, and labeling of multi-cycle paths, all to give a more accurate and relative timing analysis result. Exceptions are not used in the implementation phase, only in the timing analysis phase.

“...use TimingDesigner to setup your interface requirements in a timing diagram...”

Constraint Complexities

FPGA manufacturers recommendations on controlling their tool set based on design characteristics for specifying data constraints can be quite confusing and complicated. They take into account whether data is to be edge-aligned or center-aligned, whether delay information is based on expected skew, or based on actual external delay factors such as PCB trace lengths and external device setup and hold requirements. The confusion comes from the complex nature of the manufacturers timing analysis engine.

Most all FPGA manufacturers incorporate timing analysis engines in their development system software to provide an indication of correct design operation with regard to timing. For a post-fit design implementation, the timing analysis engine runs strictly on the notion of incrementally adding up routing delays and primitive element pass-throughs along with clock to output delays, and compares that with design requirement characteristics passed to it through design constraints. With these elemental entities, the analysis engine can algorithmically calculate a timing situation and provide designers with an indication of the designs expected performance.

Since these built-in timing engines don't possess a direct ability to determine when a signal edge will occur, they instead deal with the edge relationships in a more indirect approach, relying on 'relational' information from the data constraints that describe when to expect data to occur in relation to its latching clock edge. Results are calculated based on reference clock period, the internal primitive and interconnect delays associated with the data signals, and the clock/data alignment style for the interface. The accuracy of this information is dependant on the user to understand how the analysis engine interprets situations for setup time, hold time, and clock-to-output times based on the style of design they are implementing, because the values designers enter as constraints directly affect the analysis results. All told, this can be a complex and confusing process.

Creating Constraints with TimingDesigner

An alternative approach is to use TimingDesigner to setup your interface requirements in a timing diagram, have it generate the numbers necessary for a complete design 'fit', then extract the post-route delay information from the FPGA's timing analysis tool. This approach alleviates the necessity of understanding the complexities inherent with the timing analysis engine of FPGA tools to reporting correct timing numbers.

Simply set up your timing diagram to accurately reflect the originating signal characteristics, the affects of board delay and other externally

influenced delay elements, and then place Measure events in the appropriate place for either input or output data constraints. These added events have the ability to generate specific SDC constraint information into a constraint file. Once the SDC file has been generated, the design can be synthesized and fit into the FPGA part via the respective development system. After fitting is complete, the post-fit timing information can be generated and then imported into TimingDesigner to be applied to the associated signals.

“This approach alleviates the necessity of understanding the complexities inherent with the timing analysis engine of FPGA tools to reporting correct timing numbers.”

Input Delay Representation

Figure 3 below represents a source synchronous DDR input situation. Note the location of the Measure event for the determination of the ‘maximum’ input delay constraint and the location for the ‘minimum’ input constraint. This placement is consistent with the definition for specifying SDC input delay, where the delay value represents the amount of time before a signal is available after a clock edge. Note that the minimum data available position is where the

data begins to transition to the new state, and the maximum data available position is where the data ends its transition to the new state. This means that the minimum input delay situation for an edge that occurs before the clock edge would have a negative value. This is consistent with manufacturer’s timing analysis engine expectations for input delay.

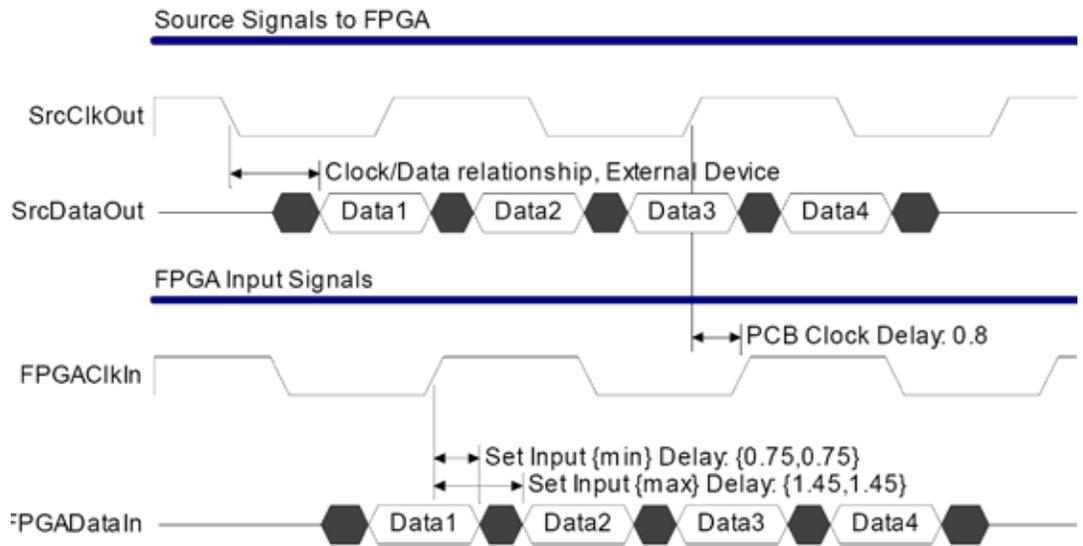


Figure 3 - Example of a DDR center-aligned data input interface to generate SDC design constraints

Output Delay Representation

“The final results will yield a complete timing picture which can be imported into TimingDesigner for accurate application of delay affecting the diagram waveform relationships.”

For output delay, the diagram would be created as illustrated in Figure 4 below with the originating signal characteristics emerging from the FPGA and propagating to the associated external device input pins. This allows observation of the required clock/data relationship needed for the external device to correctly capture the necessary data, and easily illustrates any necessary signal correction at the source.

Output delay for SDC is defined to be the amount of time necessary for a data signal before the clock edge occurs. The Measure events in this case are applied on the signals emanating from the FPGA with the ‘maximum’ delay corresponding to what’s known as data setup time, and the ‘minimum’ delay corresponding to the data hold time. Note that any data edge that occurs after the corresponding clock edge will be a negative value.

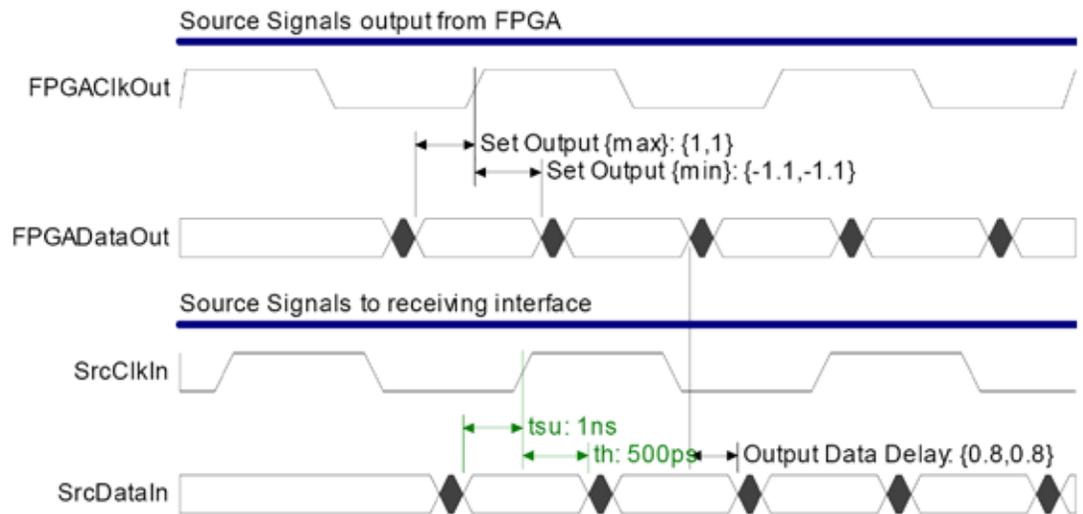


Figure 4 - Example of a DDR center-aligned data output interface to generate SDC design constraints

Extracting Post-fit Delay Timing

Once the design constraints have been established for the design, the next steps are to synthesize and fit the design into the targeted device and achieve a resulting timing netlist that contains the incremental signal delay information. Each FPGA manufacturer supplies their own fitting software algorithms that determine how the design will be implemented and how the various connectivity decisions will be made. Tightly coupled with those algorithms are the timing netlist data that reports delay associated with each incremental element for the signal path. The final results will yield a complete timing picture which can be imported into TimingDesigner for accurate application of delay affecting the diagram waveform relationships.

After all, what we are trying to achieve here is the resulting effects of our implemented design as fit into the FPGA device. We already have the desired signal relationships established in the diagrams that incorporate how our FPGA interacts with the other critical devices in our communication paths. So the natural progression is to update these diagrams with the delay information from our FPGA tool set and verify that our design will still meet timing.

“TimingDesigner delivers SDC constraint generation and accurate critical path timing analysis results required for highspeed interface design.”

A Two Pass Fitter Approach

Most all designs will require two iterations through the FPGA tool set, the first to establish the initial path and element delays of the corresponding fit, and the second to fine tune the phase control for our clock/data relationships. Initial fitting basically establishes the elements the design will need to use along with their associated delays and allows the base line to apply to the waveform relationships from which any further design adjustments can be made. This is necessary because each FPGA logical element has delay to apply to a passing signal, along with the routing elements used for connectivity. These delays are omni-present and aren't affected by internal clock shifts due to phase-locked loop settings or other external signal affects.

Once the base line is established, we can then examine the effect on the signal waveforms in our timing diagram, and then make decisions

on how much clock signal adjustment we'd like to make to achieve the desired signal relationships. We can determine how much clock shift will be needed in order to center-align it with our data valid window for instance, or to possibly establish a more perfectly edge-aligned relationship for designs requiring it. It's simply a means of establishing the exact difference between the current location of the clock edge and the desired location using Measure events and various formulas in TimingDesigner's dynamically linked Parameter Spreadsheet for calculating the necessary shift as illustrated in Figure 5. SDC constraints can then be regenerated, the design re-fit to accommodate the changes, and again import the delay results into TimingDesigner to verify the results.

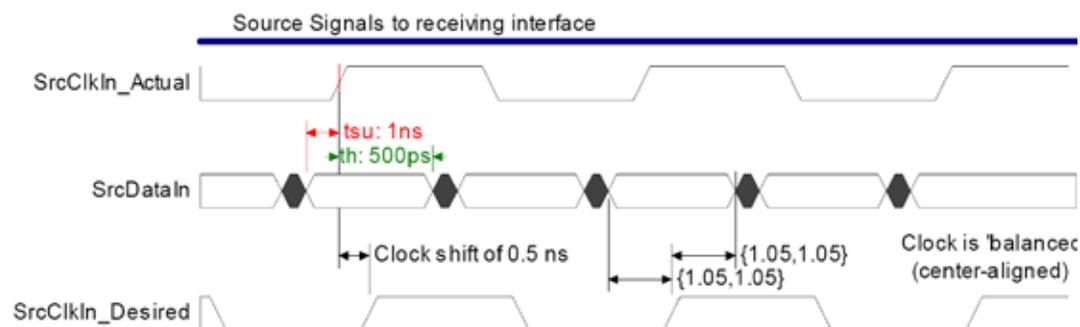


Figure 5 - Example of determining clock shift to achieve a balanced clock/data relationship

Summary

High-speed designs often have stringent specifications and tight release schedules, so there's a valid need for an interactive timing specification and analysis tool to obtain fast and complete timing margin analyses to address all of the factors that can ultimately affect design success. TimingDesigner used along with FPGA tools provides a powerful solution set that allows you to tackle these timing challenges. The versatile clock configurations and abundant I/O resources of today's FPGA devices allow high

throughput data transfer, while TimingDesigner delivers SDC constraint generation and accurate critical path timing analysis results required for highspeed interface design. Together, these tools allow you to accurately capture and exchange critical interface timing information, and allow visual verification that your design will perform as desired.

For more information please visit: www.TimingDesigner.com.

Where to Go for Help

Support

EMA Design Automation is committed to providing unsurpassed customer service and support. Our support site includes an extensive portfolio of TimingDesigner information to keep you productive.

You'll get access to product downloads, a searchable knowledge base, online tutorials, training information, FAQs, user manuals and more. Please visit our website <http://support.ema-eda.com/> to access more detailed information.

Support Contact Information

Technical Support.....	techsupport@ema-eda.com
License Information.....	update@ema-eda.com
Phone (within North America).....	800.813.7494
Phone (outside North America).....	585.334.6001
Fax.....	585.334.6693

General Contact Information

EMA Website.....	www.ema-eda.com
TimingDesigner Website.....	www.timingdesigner.com
Information and Sales.....	info@ema-eda.com
Jobs.....	resumes@ema-eda.com