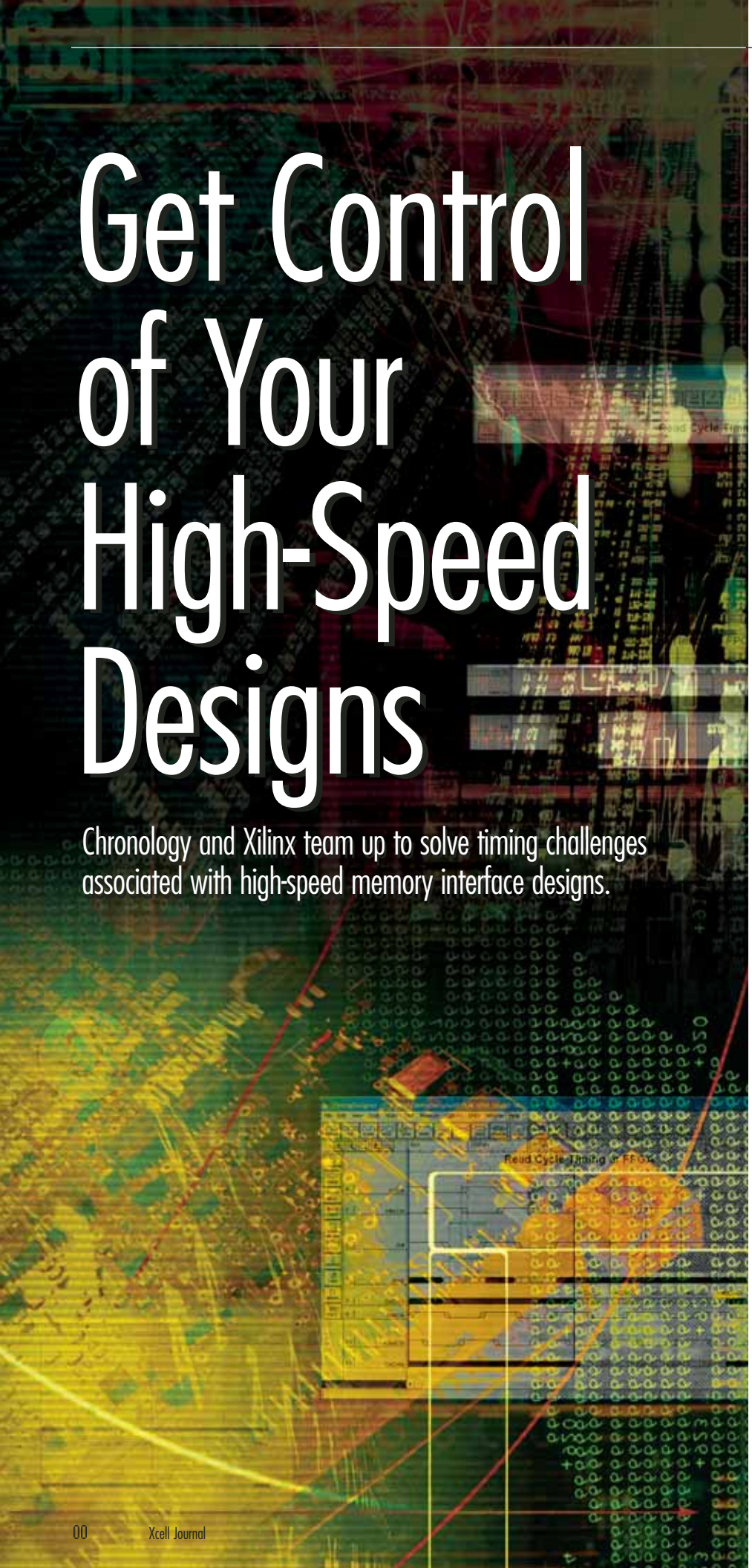
Get Control of Your High-Speed Designs

Chronology and Xilinx team up to solve timing challenges associated with high-speed memory interface designs.



by Jerry A. Long
Technical Marketing Manager
Chronology Division, Forte Design Systems
jlong@forteds.com

It's a given – the ever-increasing demand to get your designs to process more information in less time. Designing with new high-speed technology helps you meet market demands, but it's likely you're also encountering difficult interface design challenges. Timing issues previously deemed insignificant are now impacting design schedules and can no longer be overlooked.

Design teams are seeking extremely efficient, low-latency components that can operate at very high frequencies. With the introduction of QDR™ (quad data rate) SRAMs (co-defined and developed by members of the QDR SRAM Consortium and Xilinx), designers now have first-rate performance devices available for their design projects.

Created for systems operating above 200 MHz on Xilinx Virtex™-II FPGAs, QDR SRAM devices provide an unprecedented four data transactions per clock cycle. QDR SRAM devices implemented on Virtex-II Pro™ Platform FPGAs include advanced features that support the design of high-speed memory interfaces.

To fully take advantage of the high-performance features of these new devices, however, you must account for the intricate timing issues associated with high-speed interface design (Figure 1), including:

- Analyzing timing options
- Incorporating signal integrity and physical effects into timing considerations
- Managing and monitoring timing margins throughout the design process.

If you isolate timing considerations within the confines of each stage of a project, the design can be extremely costly in terms of time, effort, and resources. You simply cannot compromise accurate exchange of timing-critical data among engineering teams without putting design specifications and release schedules at risk.

Chronology's interactive TimingDesigner™ specification and analysis tool meets the stringent demands of designing high-speed memory interfaces implemented on Virtex-II and Virtex-II Pro Platform FPGAs.

New Design Challenges

Along with the increased throughput advantage of high-performance devices comes the inevitable interface design challenge of ensuring accurate data transfer.

QDR SRAM devices allow four data transactions per clock cycle by providing separate read and write data buses, each with DDR (double data rate) performance characteristics. Separate data buses require:

- Twice the I/O pin requirement for data
- Two clocks
- "Center-aligned" data presentation.

Virtex-II Pro technology allows you to leverage these features with its abundant I/O resources and high-performance clock management circuitry. However, the benefit of designing with a Virtex-II Pro device does not by itself address all the timing challenges.

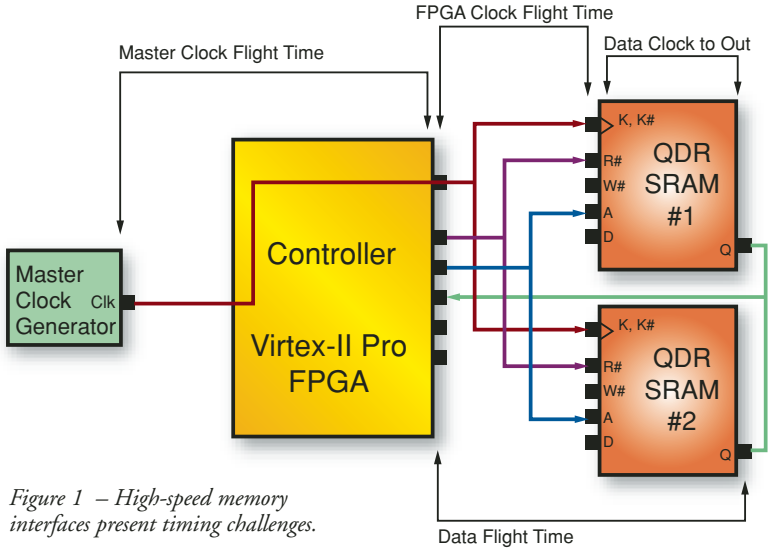


Figure 1 – High-speed memory interfaces present timing challenges.

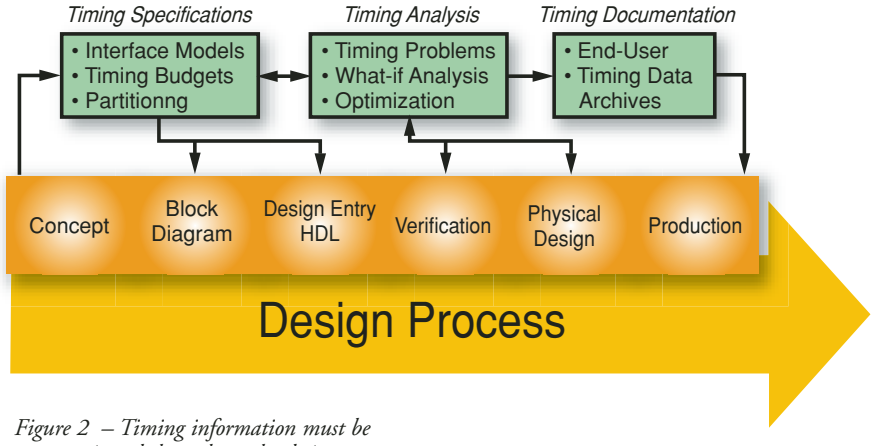


Figure 2 – Timing information must be communicated throughout the design process.

Ever-Shrinking Timing Margin

As operating frequencies rise beyond 200 MHz, the period within which data can be captured and presented decreases to 5 ns or less. At these frequencies, the margin for setup and hold times dwindles. Shrinking margins mean less room for securing an accurate data capture and presentation window.

Further complicating matters are increased crosstalk susceptibility and transmission line effects that result from higher operating frequencies. Signal integrity issues are a significant factor in high-speed designs, and you must monitor them as your design progresses. The combined effects of parasitic PCB traces, IC packaging, bond wires, and physical die characteristics adversely impact signal quality. These effects require additional settling time, therefore shrinking the timing margin you have available for reliable data capture.

Communication Is Critical

Project teams cross multiple functional organizations, each with their own sets of software tools and workflows to meet their specific design needs. For high-speed design, it is imperative to monitor and manage timing margin limitations throughout the design process. This places new automation and accuracy requirements on the exchange of design specifications among engineering teams.

Too often, critical timing margin information is miscommunicated among engineering teams. These teams depend on text documents created by manual interpretation of large complex report files that were generated by tools designed for other uses. Such timing documents are often laborious to maintain.

They also introduce the opportunity for error, especially because timing margin information changes frequently throughout the course of a design project. To reduce the risk of error, teams must have a common, automated means to accurately communicate and exchange timing data (Figure 2).

TimingDesigner Analysis Tool

When designing high-speed interfaces, you need:

- A means to detect timing problems early
- The ability to display visual representations of design requirements
- A tool to easily implement alternative solutions.

The TimingDesigner interactive timing specification and analysis tool is being used by interdisciplinary project teams when designing high-speed interfaces for Virtex-II and Virtex-II Pro FPGAs and QDR SRAM memory devices. TimingDesigner allows you to create interactive timing diagrams for capturing interface specifications,

analyzing component interface timing, and communicating design requirements among project engineering teams.

Timing Diagrams and Issues

The way TimingDesigner incorporates the use of timing diagrams is innovative and extremely versatile. You can create diagrams that focus on interface protocol specifications, where each functional protocol operation is represented with a single diagram.

Alternatively, you can create diagrams to focus strictly on a particular signal path, with each signal representing a different propagation point along the entire critical path. TimingDesigner's built-in static timing analysis engine immediately calculates and reports worst-case timing margins as you explore design alternatives. This allows you to easily model, analyze, and manipulate the most critical parts of your design.

Component Diagrams

When you want to examine high-speed memory interface timing based on your design components, TimingDesigner generates a separate diagram for each component involved. Each diagram represents the particular operation of the interface protocol, such as a read operation for the QDR SRAM.

Additionally, each diagram includes related signals and the timing requirements associated with the operation. Event relationships such as delays for control signal events, data signal setup and hold requirements, and clock jitter tolerances are also represented in the TimingDesigner diagram.

This feature gives you quick and accurate visual reference for the interface requirements. It also allows you to create a complete interface protocol specification while promoting diagram re-use as multiple diagrams accumulate in an interface protocol library.

Once these component-based diagrams are drawn, you can easily merge them to perform timing analysis for the entire interface circuitry (Figure 3). The receiving component interface diagram is linked to the initiating component diagram via delays associated with anticipated PCB trace information. You can also create additional signals and clocks at incremental points along the timing path. This technique provides an added degree of visibility of signal propagation delays, helping to illustrate complete timing closure for critical timing paths.

Signal Path Diagrams

When you need to focus on individual critical paths, TimingDesigner can create timing diagrams based on designated signal paths. In this case, you create a signal for each point of interest along a specific signal propagation path. By doing this, each stage that may influence the designated signal – such as I/O buffer delays, PCB flight path delays, signal integrity delays, or other known influences – is illustrated in the critical path analysis. This approach reveals the impact on timing margins, and ultimately, the source of timing failures. The capability to trace each critical path from start to finish gives you confidence in achieving timing closure.

Other Timing Diagrams

You can also use TimingDesigner diagrams to accurately represent complicated clock phase and de-skew relationships, which are required for the design of QDR SRAM clocking signals.

Using the various clock features provided in TimingDesigner, such as accurate jitter representation and derived clock elements, you can generate diagrams that can illustrate the clocking characteristics needed for practically any high-speed interface. You can then easily recognize the clocking requirements and confidently program the digital clock managers (DCMs) in Virtex Platform FPGAs for dependable QDR clocking schemes.

Built-In Static Timing Analysis

No matter what style of timing diagram you choose to implement, TimingDesigner's built-in static timing engine delivers accurate signal timing analysis at any stage in the design process. At each design stage, the static timing analysis engine can instantly trace all of the delay paths specified in the timing diagrams, remove common uncertainties, adjust for track delays, select critical paths, compute worst-case timing margins, and flag violations.

This intuitive use of timing diagrams offers quick evaluation of design alternatives – in addition to fast calculation of timing margins and insight into solving complex timing problems early in the design process.

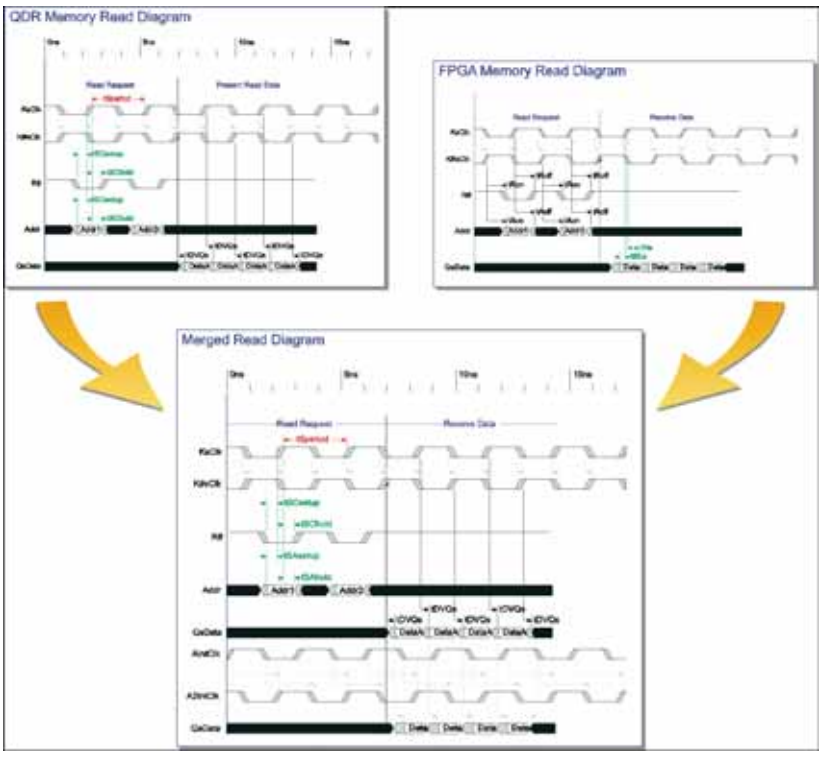


Figure 3 – Merged diagrams allow for complete interface timing analysis.

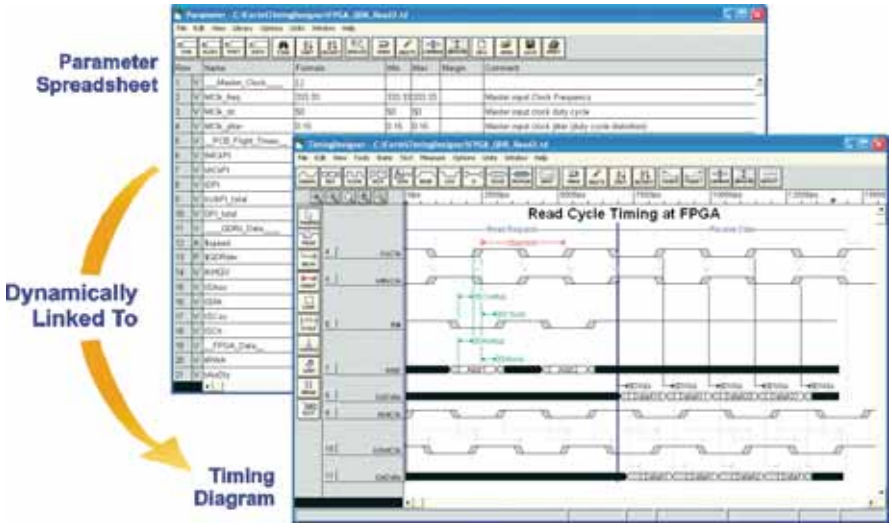


Figure 4 – A dynamically linked spreadsheet provides immediate feedback for easy exploration of timing alternatives.

What-If Timing Analysis

All edge relationship events entered in a TimingDesigner diagram (for example, delays, constraints, guarantees, and measures) are automatically generated in a dynamically linked Parameter Spreadsheet (Figure 4). The tabular format of the timing events represented in the diagram enables you to easily manipulate value fields, as well as comment fields, for event descriptions.

The Parameter Spreadsheet also supports the parameterization of timing diagrams with variables to represent various timing scenarios. These variables can be referenced from events in the diagram.

For example, clock characteristics such as frequency, period, phase shift, and jitter can be represented with variables in the spreadsheet, and then used in other formulas to create cycle-accurate timing relationships. This aspect of TimingDesigner provides added flexibility when designing QDR SRAM interfaces that require center-aligned data or when illustrating various clocking signals of your Virtex-based design.

The Parameter Spreadsheet gives you the ability to create complex formulas to model path delay variables, signal loading, temperature, and other elements that may impact timing relationships.

Using variables in the Parameter Spreadsheet facilitates quick and accurate "what-if" analysis. Throughout the design process, you can evaluate circuit functionality and performance using different sets of parameterized values, providing a highly flexible and productive design environment.

For example, you can evaluate numerous operating frequencies by simply changing the variable used for clock frequency. You can also use this method to evaluate multiple Virtex-II and Virtex-II Pro clocking schemes by providing variables for phase relationships to determine the most appropriate phase increment for the DCMs.

Library Spreadsheet Flexibility

To further expand timing analysis flexibility, you can use the TimingDesigner Library Spreadsheet to create part-specific timing information that can be used by any timing diagram. This is especially useful for devices where a single diagram (or set of diagrams) can model the device functionality and then reference a Library Spreadsheet for specific speed grades or voltage grades.

Expanding on this concept, you can easily switch in and out of libraries of different timing relationships to evaluate cost and performance trade-offs and help determine the best speed grade option for a specific design.

Timing libraries are also very effective when analyzing pre-route versus post-route timing characteristics of a design. You can use post-route timing reports from Xilinx ISE tools (or other layout tools) to create a TimingDesigner library. Because the library file format is a comma-separated ASCII file, creating multiple timing libraries specific to a particular routing run is easy. You can insert these libraries into the existing timing diagram analysis for a post-route confirmation that critical timing paths have been satisfied.

TimingDesigner Saves Time

TimingDesigner diagrams are an excellent documentation medium for interface specifications. The easy-to-understand format of these diagrams reduces the chances for error due to misinterpretation by other project team members. The format provides a clear and concise view of available timing margins.

Each design team can interact with the timing diagram, making available their incremental contribution to the overall timing path result. Thus, you get immediate notification of potential timing problems that may be uncovered during different stages of your design process.

Using TimingDesigner's OLE capability, timing diagrams can be used in their native format or embedded or linked into commonly used publishing and presentation tools. This provides a simple way to convey critical timing path information among design groups. Additionally, TimingDesigner exports common image formats, including MIF, EMF, TIF, and EPS. Import of VCD and FSDB simulation files is also supported.

Conclusion

Together, the Chronology Division of Forte Design Systems and Xilinx provide a powerful solution set that allows you to tackle the timing challenges of high-performance design. High-speed designs often have stringent specifications and tight release schedules, so you need an interactive timing specification and analysis tool to obtain fast and complete timing margin analyses – addressing the factors that can ultimately affect your design success.

The versatile clock configurations and abundant I/O resources of the Virtex-II Pro Platform FPGA allow high throughput data transfer, while TimingDesigner delivers accurate critical path timing analysis results required for interface design with high-speed memory devices such as the QDR SRAM.

To learn more about TimingDesigner and the Chronology Division of Forte Design Systems, please visit *www.timingdesigner.com.*