DESIGN TECHNIQUES FOR HIGH-SPEED SOURCE SYNCHRONOUS BUSES

Douglas Burns, Principal Consultant
Daniel Nilsson, Senior Consultant
Stephen Coe, Senior Consultant
Dr. Walter Katz, Chief Scientist
Barry Katz, President

Signal Integrity Software Inc.
www.sisoft.com

SiSoft

Abstract

System designers are faced with the challenge of increasing inter-chip bandwidth without adding IO cells to an ASIC library. One technique for doing this is Source Synchronous communication. An 833MHz DDR cache design and a 24 inch long, 500MHz DDR inter-processor system bus are used to illustrate the complexity and issues surrounding source synchronous design. For both cases, the design requirements and technical challenges are described. Each analysis considers numerous affects including, inter-symbol interference, reflections, termination, interconnect loss, etc. The complexity presented in these designs demonstrates the need for a methodology that can handle all of the design variables in an automated way. This automation must include simulation generation, waveform rules analysis, waveform timing extraction, and static timing analysis. Presented simulation and laboratory results validate the accuracy of the design methodology. With this methodology in place, design of 1GHz source synchronous buses should be possible.

Authors Biography

Douglas J. Burns graduated Magna cum Laude from the University of Massachusetts in 1981 with his BSEE and received his MSEE from Northeastern University in 1986. Between 1981 and 2000, Doug worked for Honeywell, Digital, and Compaq Computer. He has led VLSI design implementation teams, signal integrity teams, ASIC implementation teams, and influenced the system architecture of ALPHA systems. Doug consulted on SI and implementation issues across many groups within Digital/Compaq, and has a successful track record of bringing products to market. Current, Doug holds four patents in computer system design and has two additional patents pending. Doug's expertise spans a wide range of engineering disciplines including: system architecture, VLSI design, package design, timing analysis, interconnect analysis, cross-talk analysis, electromagnetic modeling, IO buffer analysis and selection, simultaneously switching output analysis, termination selection, clock distribution and skew analysis, and high-speed bus design.

Daniel Nilsson received his MS in Electrical Engineering from Luleå University Of Technology, Sweden in 1999. He started his career in 1998 at Ericsson Microwave Systems in Mölndal, Sweden when he joined their High Speed Board Design group. During the years at Ericsson, Daniel worked in defense electronics as well as telecommunication equipment projects. The tasks involved hardware design as well as software design. One of the major contributions as Ericsson was the system partitioning and signal integrity work on Ericsson's next generation AXE telecommunication switches based on the Alpha Microprocessor. Other projects involved a graphics interface for a Modular Airborne Computer System (MACS). His expertise lies within the areas of high speed interconnect design and analysis, system partitioning, and computer system administration. Daniel joined Signal Integrity Software Inc. in 2001 as a Signal Integrity Consultant.

Stephen R. Coe received his BS in Electrical Engineering from the University of Massachusetts in 1983. Steve began his signal integrity career when he joined Digital Equipment Corporation in 1987. During his 13 years there, he solved the signal integrity issues for many of Digital's high performance computers. Steve influenced the design of many high performance memory, PCI and AGP systems. He has been a major contributor to the signal integrity methodology utilized in the design of VAX and Alpha based workstations and servers. He holds patents on several Alpha server products. More recently, Steve worked on the design of high-speed network appliances including routers, switches and hubs at Nortel Networks. Steve is an expert in the analysis of interconnect, crosstalk and timing, as well as interconnect topology and termination selection, I/O selection, and electromagnetic modeling of interconnect, packages and connectors.

Dr. Walter Katz is a pioneer in the development of constraint driven printed circuit board routers. He developed SciCards, the first commercially successful autorouter. Walter founded Layout Concepts and sold routers through Cadence, Zuken, Daisix, Intergraph and Accel. More than 20,000 copies of his tools have been used worldwide.

Walter developed the first signal integrity tools for a 17 MHz 32-bit minicomputer in the seventies. In 1991, IBM used his software to design a 1 GHz computer.

Walter holds a PhD from University of Rochester and a BS from Polytechnic Institute of Brooklyn.

Barry Katz has been at the forefront of high-speed design throughout his career, devoting his efforts to solving the problems faced by designers of leading edge high-speed systems. In 1995, Barry founded Signal Integrity Software, Inc. (SiSoft) where he has assembled a team of world-class experts committed to solving the most challenging high-speed design problems facing the industry today by delivering a comprehensive design methodology, software tools, and expert consulting. Barry has been a major influence to the signal integrity methodology utilized by numerous companies and has personally led multiple signal integrity design teams. He has expertise in all aspects of high-speed design including: timing analysis, interconnect analysis, crosstalk analysis, electromagnetic modeling of interconnect, packages and connectors, IO buffer analysis and selection, decoupling analysis, simultaneously switching output analysis, interconnect topology, termination selection, clock distribution and skew analysis, and high-speed bus design.

Barry holds a MSEE from Carnegie Mellon and a BSEE from University of Florida.

SiSoft

Introduction

Design Problem

Advances in silicon technology have allowed designers to develop IC's with internal clock rates that were once considered unobtainable. In order to utilize the increased processing capabilities of these higher internal frequencies, system designers are being challenged to increase IO bandwidth. Unfortunately, advances in IO drivers, packages, and PCB's are failing to keep pace with transistor scaling. Thus the need to get increased bandwidth from existing IO devices. To meet system requirements, designers are moving away from conventional synchronous design. One technique that is being exploited is Source Synchronous. This technique, while not new, has enjoyed a renewed interest because it allows a significant increase in IO bandwidth while using existing IO technology. Source synchronous design utilizes a clock signal (strobe) generated by the same chip that is transmitting the data. The designer must consider PVT (process, voltage, temperature), signaling rate, simultaneous switching noise, intersymbol interference (ISI), crosstalk, clock jitter, etc. As the bus switching frequency increases, the significance of these affects increases, thus more simulations must be run to ensure that all issues have been considered. This will result in a large simulation database and a comprehensive design methodology must be put in place. There is no magic bullet that solves all high-speed design issues. What a designer needs is a good understanding of the design issues and an environment that allows them to examine how these issues interact.

To illustrate the issues with source synchronous design and the type of methodology needed to obtain accurate results, two case studies are presented. The first is an interprocessor system (IPS) bus. It is comprised of a 72 bit bi-directional databus and 32 bits split into 2 unidirectional address/command buses. (Figure 1). For the databus, a set of data strobes (In/Out) was used for every 9 bits. The address/command bus used a single strobe for each address bus. This bus was required to operate over a number of different physical implementations. Design variables included the use of connectors, trace loss factors, bus length, etc. The target data rate for both the data and address was 500MHz.

The second design is an external cache utilizing 9 SRAM components (Figure 2). The CPU provided a strobe to each SRAM component for the purpose of capturing address, control, and data writes. Each SRAM provides a strobe back to the CPU to allow the CPU to capture data on reads. The data is signaled at an 833MHz rate. The control bits are signaled at a 416.66Mhz rate and the address signaled at a 208.33MHz rate.

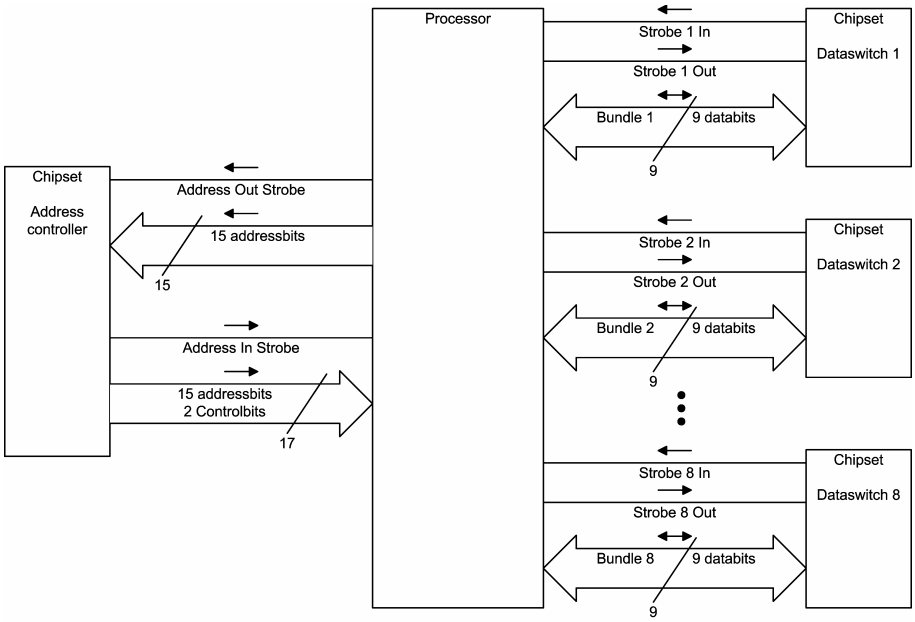

Figure 1

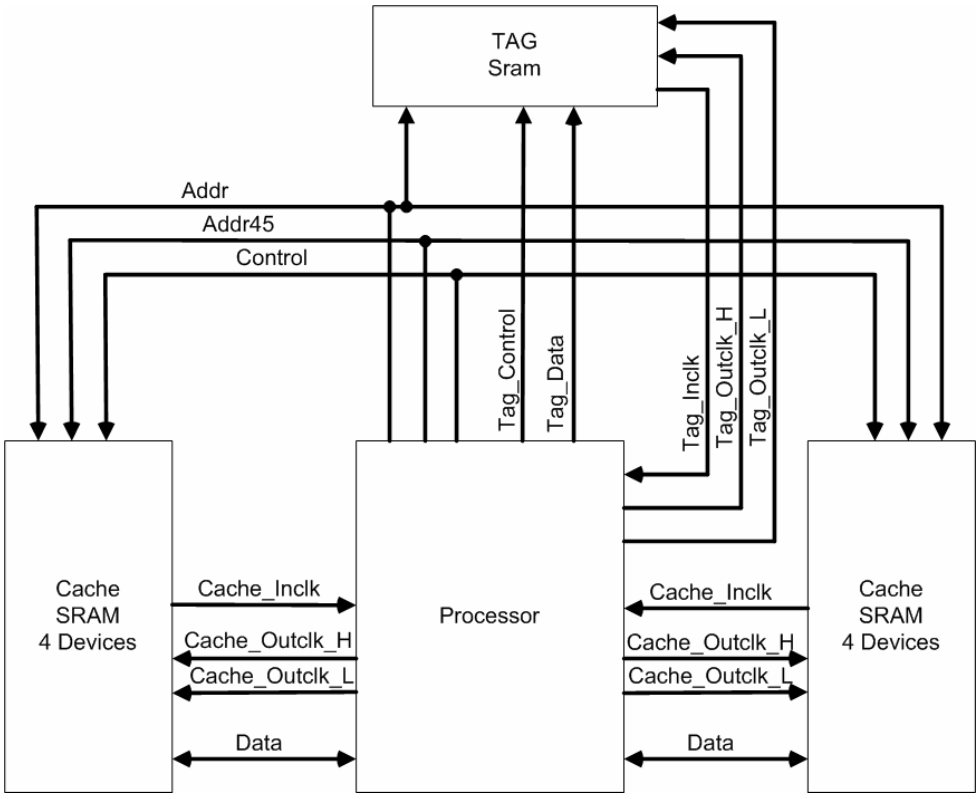

Figure 2

Design Methodology

Both of these designs represent many challenges for any design methodology. The interprocessor bus requires a methodology that can easily handle attributes of multiple configurations while the cache design incorporates complex routing topologies and multi-faceted timing relationships. Despite the differences in their implementations, each of these designs are similarly plagued with the issues of PVT, ISI, etch variations, etc.

To successfully design these interfaces, the following features are required in the design methodology:

1. Validated Library
   This is the foundation of any design and as such, the quality of the input dictates the quality of the result. A design incorporates many types of information; including the simulation models, timing models, design netlist, and allowed transfers. Each of these pieces of information is interrelated and the methodology needs to ensure consistency between them. Differences must be reported.

2. Model Characterization
   IO cell specifications give only a partial view of how an IO cell operates. These specifications are developed with a set of assumptions based on the cell use and input conditions. A thorough methodology needs to examine the IO over a range of frequencies, slew-rates, and input amplitudes. Data captured from these simulations can then be used to define the limits of operation for an IO. These limits must be able to be codified into a waveform and timing analysis tool to ensure that device assumptions are not violated in the actual design environment.

3. Solution Space
   In the design phase, each net class is explored. The methodology must allow variation of etch length, impedance, topology, termination, driver type, stimulus, and ac noise. In addition, this analysis needs to support both uncoupled and coupled simulation. From this data, the actual design limits of operation and the rules for layout are derived.

4. Automated Waveform Processing
   Given the large range of variables that need to be analyzed and the fact that each simulation may contain a large number of edges (to excite ISI and support coupling analyses), manual review of the simulations is nearly impossible. The methodology must support automated waveform rules analysis and timing extraction. The extracted data should report on slew rates, overshoot (both static and dynamic), ringbacks, non-monotonic clock waveforms, and failures to cross Vil/Vih limits.

5. Automated Timing Analysis
   As with waveform analysis, the large numbers of simulations make manual (or even script based) data extraction difficult. The methodology must support a means of extracting the timing information from the simulations and apply the appropriate

source/target/clock data to determine setup and hold margins. For the source synchronous nets, this methodology must be able to recognize which clocks are associated with which data bits. Detailed analysis of each network class needs to be reported.

6. IO Buffer Models
   Simulation models for signal integrity investigations are available in transistor level and behavioral formats. Behavioral allows the fastest simulation times, but transistor level models are often required for design accuracy. Models are usually available in only one of these two formats. In any case, the methodology must allow seamless support of any combination of transistor and behavioral models.

7. Consistent Pre/Post Layout
   Significant effort goes into setting up the simulations for the pre-layout analysis and the results of the pre-layout effort are used to generate the routing rules for the design. These rules include min/max etch lengths, etch topologies, and termination styles. As the layout process advances, the methodology must support:
   1. A common simulation environment for both pre and post-layout
   2. Automatic setup of the post-layout simulations. Environmental conditions, noise limits, input stimulus, measurement conditions, and measurement nodes need to be inherited from the pre-layout environment.
   3. Behavioral and transistor level models
   4. Single and multi-board extraction of networks from a mixture of host CAD systems.

   With these key features, the job of post layout simulation becomes a validation of the pre-layout work.

8. Reusability
   Many projects are followed by speed-ups, mid-life kickers, vendor process changes, or design repackaging projects. These follow-on projects generally utilize most of the devices found in the original design, but may boost frequency or add additional system functionality. Re-use of the design library and major pieces of the design analysis description data allows these new projects to be completed more quickly. The methodology must support any combination of serial or parallel development of these sibling projects.

9. Result Reporting
   How data is reported is critical. The methodology must provide the means to allow the designer to view design results in the detail that they need to solve a problem. Reports that summarize various views (roll ups) of the data to reports that provide explicit detail on every edge of every simulation are required and is a process we call Progressive Discovery.

Steps to Design Success

To meet the needs of these designs, a methodology called SiAuditor™ was developed and used throughout.

Library Development

The first task is to identify the information to be kept under library control. For the SiAuditor methodology, we place enhanced IBIS components, Hspice Models, and timing Models under version control. This is to ensure that the changes to the library can be tracked. The enhanced IBIS components are pin based description of which IO driver model is associated with a pin of a device and also includes the model descriptions. These descriptions contain standard IBIS measurement statements (Vmeas, Vil, Vih, etc) as well as the device model. For SiAuditor, the device model can be an IBIS behavioral model, Hspice transistor model, or both.

Timing models are files that specify the output delays and setup/hold times of each device pin and capture other relevant pin-to-pin timing relationships.

Hspice models, provided by component vendors, are "wrapped" into a standardized format that allows them to be used interchangeably with IBIS models.

Careful creation of all of the above files is critical to the overall design methodology. The time spent to develop these files will result in large productivity gains further in the design process.

Design Partitioning

With all of the library files completed, we can now focus on the actual design. The first step is to classify the networks in the design into a Transfer Net. A Transfer Net describes how devices are connected at a level of abstraction that automatically maps both the logical and physical implementation to the Transfer Net List. Signals are considered to be in the same Transfer Net if they have the following characteristics:

1. Identical source driver
2. Identical receiver
3. Same number of node connections
4. Same clock connections
5. Support the same bus transfers

For the cache design 12 transfer nets are defined and are shown in Table 1. In simplified terms, the cache can be described as follows: The cache contains a data bus that operates at an 833MHz data-rate. The cache control operates at one-half the data rate and the address signals transition at one-fourth the data rate. The cache address also drives to a tag store that utilizes it's own control and data ports, but that operate in a manner similar to the cache's.

Table 1 Cache Transfer Nets

Transfer Net	Features
ADDR	Connects to Cache &Tag SRAM, Updates at ¼ cache data rate, Unidirectional
ADDR45	Connects to Cache SRAM, Updates at ¼ cache data rate, Unidirectional
CONTROL	Connects to Cache SRAM, Updates at ½ cache data rate, Unidirectional
DATA	Connects to Cache SRAM DQ pins, Bi-Directional
CACHE_INCLK	Connects to Processor cache clock input, Unidirectional
CACHE_OUTCLK_H	Connects to Cache SRAM Clock High input, Unidirectional
CACHE_OUTCLK_L	Connects to Cache SRAM Clock Low input, Unidirectional
TAG_DATA	Connects to Tag SRAM DQ pins, Bi-Directional, ¼ cache data rate
TAG_CONTROL	Connects to Tag SRAM, Unidirectional, ¼ cache data rate
TAG_INCLK	Connects to Processor tag clock input, unidirectional
TAG_OUTCLK_H	Connects to Tag SRAM Clock High Input, Unidirectional
TAG_OUTCLK_L	Connects to Tag SRAM Clock Low Input, Unidirectional

The interprocessor system bus requires 7 transfer nets, as shown in Table 2.

Table 2 IP System Bus Transfer Nets

Transfer Net	Features
SYS_ADDIN	Unidirectional, Chipset to Processor, Data
SYS_ADDOUT	Unidirectional, Processor to Chipset
SYS_DATA	Bidirectional
SYS_ADDINCLK	Unidirectional, Chipset to Processor, Address Clock
SYS_ADDOUTCLK	Unidirectional, Processor to Chipset, Address Clock
SYS_DATA_INCLK	Unidirectional, Chipset to Processor, Data Clock
SYS_DATA_OUTCLK	Unidirectional, Processor to Chipset, Data Clock

This design is less complex than the cache design in that it contains no multi-drop networks. Figure 3 depicts the ADDR transfer net for the cache. Figure 4 depicts the SYS_DATA transfer net for the IPS bus. These Transfer Nets completely describes the data buses and it is important to note that the Transfer Nets contain no information about the physical implementation. Thus the Transfer Nets are re-usable for any physical implementation of these interfaces. The actual parts (CAD part, IBIS component, and Timing model) are wrapped inside of the SiAuditor Part. Design speed variants are easily handled by changing these part wrappers.

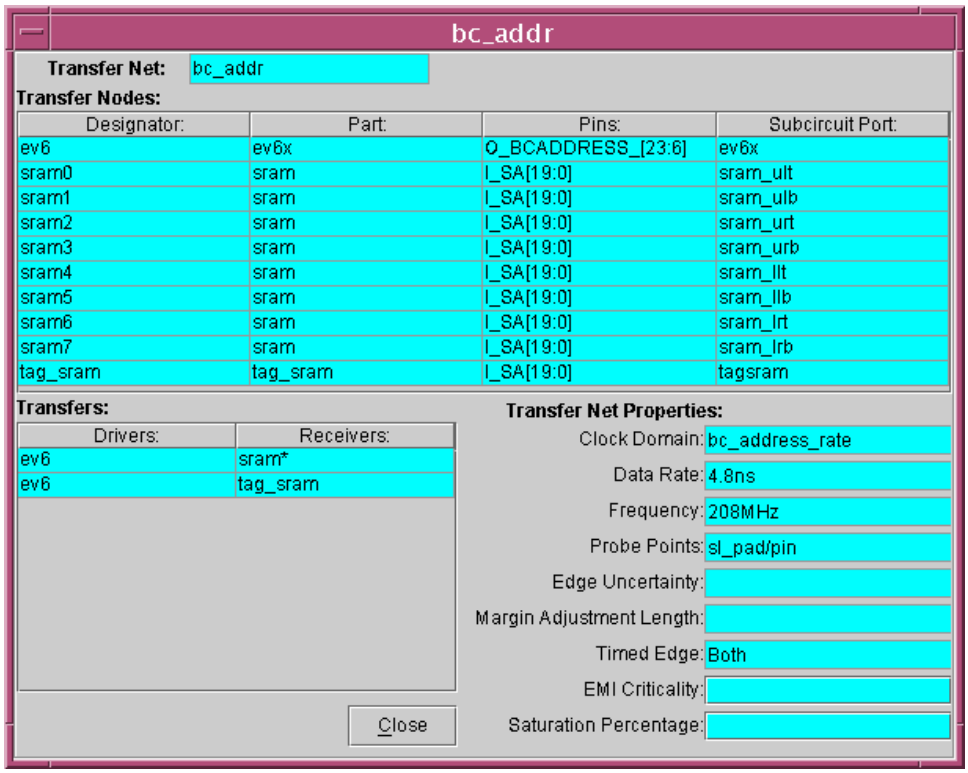

Figure 3

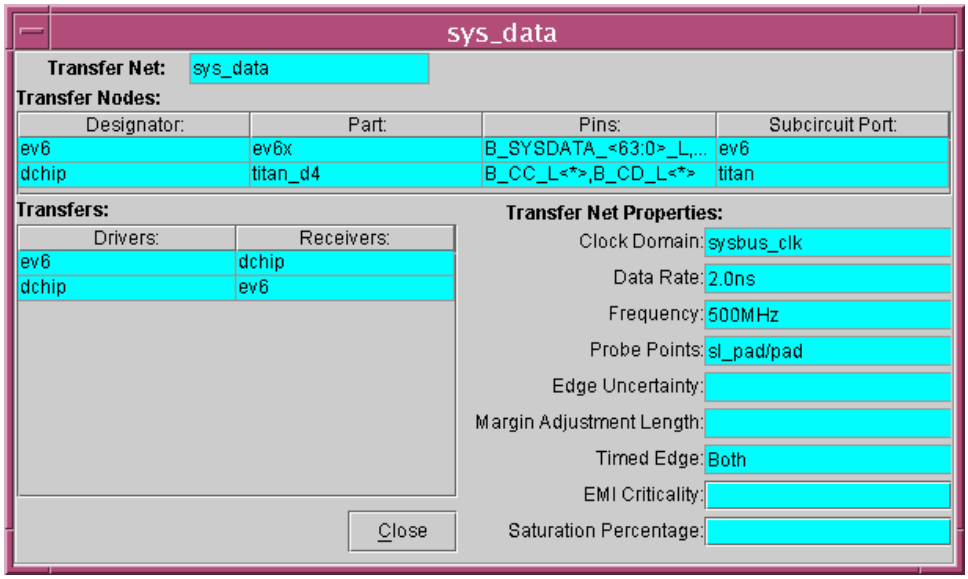

Figure 4

Supporting Data Files

Other data to be used by the simulation and analysis tools include:

1. Clock domain: This identifies each of the network switching rates and groups networks based upon their datarate.
2. Clock skew: This describes the clock skew found between components.
3. Topology Files: These files are spice subcircuits that describe the network connection. These are generic files that allow parameters to be passed into them. Passed parameter include etch lengths, impedance, termination voltages and termination values

Design Analysis

Once the libraries are setup, the design partitioned, and the supporting data files defined, we have everything needed to simulate, waveform analyze, and extract timing. To ensure that the models meet the design requirements, an IO characterization is performed. The results of this analysis informs the designer about maximum IO operating rates, the effects different slew rates cause, and what the minimum allowable input swings can be. Using this, the designer can fine tune the timing models and the waveform measurement levels in the IBIS component.

Solution Space Setup

The first step in the cache analysis involves set up of the solution space for each network. This involves making educated guesses about etch lengths, topology, termination, and so on. This data is entered and simulations started. Figure 5 shows an example solution space for the cache address.

The main areas of concern for the cache analysis involved etch lengths and termination techniques. As can be seen, the simulations include a wide variety of combinations. By simulating these combinations, a better understanding of the design limits and the flexibility that can be provided to the layout designer is identified. Analysis of the all transfer nets for the entire cache required 1084 simulations. This covered multiple etch configurations and PVT corners. Since each of these simulations contained six or more waveform edges, over 7000 edges need to be processed.

The IPS bus was concerned with issues of etch loss factors, connector performance, multiboard etch variations, and termination. This bus was expected to be physically long, so a more complex data stimulus was needed to excite the ISI components of the bus. Analysis of the all transfer nets for the entire IPS bus required 540 simulations. This covered multiple etch type, termination values, and PVT corners. Each of these simulations contained numerous waveform edges, resulting in over 11,000 edges being processed for the IPS bus.

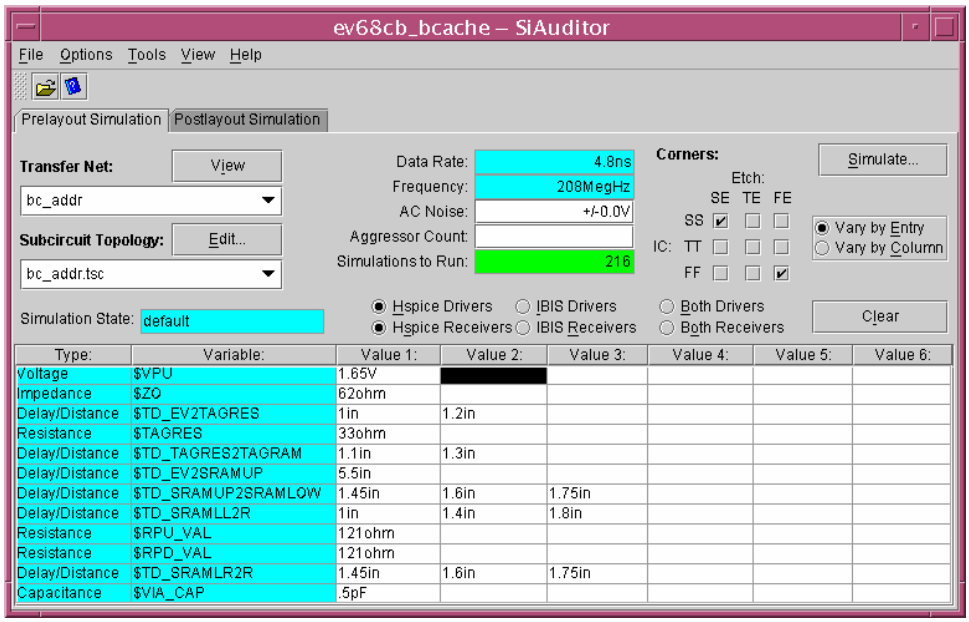

Figure 5

Waveform Processing

Generating all the simulations is straightforward. It is the analysis and interpretation of each edge that takes time. Given the need to examine 7000 edges for the cache and 11,000 edges for the IPS bus, a designer is quickly overwhelmed. The resulting amount of data and the potential for having to perform multiple iterations, requires an automated waveform processor that rigorously checks each waveform edge for the conditions specified by the designer. Checks that need to be done include:

1. Signal slew rate
2. Static Overshoot
3. Dynamic Overshoot
4. Ringback
5. Clock Monotinicity
6. Vih/Vil crossing

These checks must be done for both coupled and uncoupled simulations. Figure 6 is a representative waveform report for the Cache address net. Similar reports are generated for every transfer net. The waveform report defines which simulations in the solution space fail, the amount of the failure and where in the failing simulation deck that the failure can be found. This allows the designer to quickly view problems both from a tabular report and from the actual waveforms. By sorting the data, a designer can define the cases that best meet the design waveform requirements.

During the design of the IPS bus and cache subsystem, we discovered the following issues:

1. Cache control ISI
   The original topology consisted of an eight sram design. These loads were split into a “left” and “right” cache sub-block that consisted of 4 srams. Rams were placed on a double sided board and routed from the cpu to the first sram pair, from the first pair to the second, and then from the second to a parallel termination network. This was acceptable for cache designs with up to 600MHz data rates, where these signals would be switching up to 300MHz. At higher frequencies, the intersymbol interference on these control nets became pronounced and the signal quality could no longer be guaranteed. Waveform reports identified reduced switching levels, decreased slewrates, and non-monotonic edges. To fix this problem, the termination was moved to a tee point between the srams on both the left and right sides. This causes the capacitive reflection that comes back from the srams to appear at the same time at the termination network. The resulting energy from the reflection goes into the termination. Length matching between the termination and the srams, and between the left and right cache sides, is critical for this to work. After making these changes, the network operated correctly at 416Mhz (833MHz datarate).

2. IPS bus etch loss factors limits
   With a target speed of 500MHz DDR, a total etch length of 24 inches, and the need to cross multiple boards, design of the IPS bus required careful attention. Between the target speed and the bus length, 4 bits of information would be in flight before the 1st reflection on the bus occurred. Thus, ISI affects needed to be simulated to determine a proper operating point. One system implementation required a PCB stackup with narrow etch traces to maintain the etch impedance. Of concern was that the losses due to the narrow etch would eliminate the narrow margins found in the present design. With the methodology used, various etch models were created that represented various etch width /loss factors. Coupled and uncoupled simulations that used simple stimuli and complex ISI stimuli were performed. Comparisons were made between multiple etch widths of constant impedance. Simulations showed signal amplitude loss due to the higher etch loss factors of narrower etch. When run with ISI patterns, these higher losses helped reduce the impact of ISI. In addition, coupled simulations showed better than expected results. This turned out to be due to the reduced levels of signal crosstalk caused by the closer proximity of the signals to the planes and their slightly wider spacing. The impact of the etch loss factor was decreased by the reductions in the impact of ISI and crosstalk. Waveform and timing reports captured the differences between each etch case and provided the designer with an etch width/signal margin/timing margin trade-off.

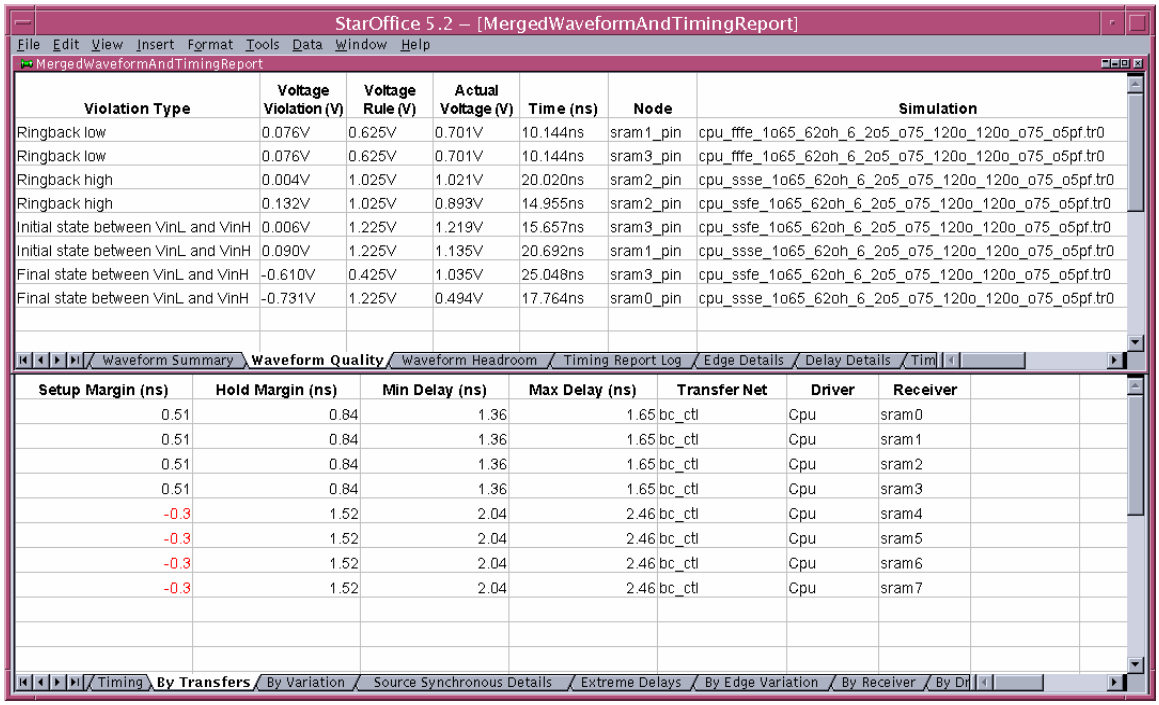

Figure 6

Timing Reports

The final question that always gets asked is “How fast can you run?” In the best case, timing passes and the designer is interested in only the final setup and hold margins. Unfortunately, there always seems to be a worst case. When this happens, a designer needs to be able to see more details, a progressive discovery of the results. Figure 6 shows a representative timing report for the cache address. This report provides the details of which events and simulations cause the design to fail. By reviewing the data, the designer can identify network combinations that, if eliminated, may solve his design problem. These combinations are then compared to the waveform analysis reports. Combinations that provide the best timing and waveform properties are then provided to the layout designer as a set of design rules.

During the design of the cache and IPS buses, the following timing issues were addressed.

1. Cache timing criteria and tradeoffs made to meet address and data constraints

   The timing on the cache interface required positioning the outclks and inclks to meet the setup and hold timing on writes, reads, for the address, control, and tagcontrol signals. In addition, there was a requirement to minimize the skew between the loop timing on the tag rams versus the data rams.

   The read timing was straight forward. Etch was added to the taginclks and datainclks to position the clock in the middle of the data valid window such that setup and hold timing margins were not violated.

The outclk timing was a bit more complex. The delay of when the data switched relatative to the address was programmable at the granularity of cpu clock cycles while the delay of when the outclks switched relative to the data was programmable at the granularity of cpu clock phases. This allowed positioning of the outclks at a granularity of 400pS. Unfortunately, this was insufficient granularity due to the tight margins of the interfaces. Thus, etch tuning was required as well. Consideration was given to using only interconnect to tune the clocks. This would have had the advantage of maintaining a constant setup time over increased cache frequency however, this had negative impact because of the fact that this extra tuning etch was uncorrelated with respect to the data etch and could have a variation of 40pS/inch. If the adjustments were done strictly in etch, this would have resulted in additional loss of margin, on the order of 120pS for both the setup and hold of all timing relative to the outclks. In addition, it turned out that as cache speeds increased, setup and hold requirements decreased thus having the tuning performed through programming would help to maintain the clock position in the middle of the eye as the bit time is reduced.

The loop timing requirement was to minimize the skew between when all read data and all tagdata arrived back at the CPU. Minimizing this skew was accomplished by insuring that the sum of the tagoutclk and taginclk interconnect delays were the same as the sum of the dataoutclk and datainclk interconnect delays while not compromising setup or hold margins at the SRAMs or CPU.

These calculations needed to be performed for multiple CPU speeds, for multiple vendors, and multiple physical implementations. With the SiAuditor methodology, design constraints and interface programming were captured allowing fast analysis time for multiple designs.

2. IPS Bus Clock Alignment Clock

A system design, nearing production release, encountered a significant increase in failures. The signal integrity and timing analysis were performed prior to the installation of the SiAuditor methodology at the customer site. The problem was identified as a timing mismatch on a source synchronous bus. Further study identified the system feedback clocks that provided tuning to the source synchronous interface as the cause. The lab data found an offset, not predicted by the original simulation methodology. The SiAuditor toolkit was employed to re-examine these networks. Simulations were performed over the entire silicon process range and the reports showed that the “SS” corner resulted in a significant timing mismatch. This mismatch was not evident at any other process corner. Examination of the lab results showed that the ASIC was indeed skewed toward the “SS” corner and that waveform results matched simulations. Given that the system was entering production, there was a strong desire to utilize the existing system etch. Fortunately, the networks already contained a parallel termination network. SiAuditor examined over 160,000 combinations of resistors and capacitors between the 2 timing paths. To excite the ISI affects, each simulation combination utilized 20 waveform edges. After waveform processing, approximately 1000 combinations provide the desired waveform properties (slew

rate, Amplitude, etc). Timing analysis found more than 500 combinations that met the signal timing relationship between the two clock paths, but only 5 combinations met both the waveform and timing requirements. Of these 5, the one that provided the best overall margins was chosen. The final result modified the existing parallel termination network values, but was able to use the existing etch with no re-work. Margin analysis in the lab showed that this change had significantly increased design margin beyond that measured in previous systems.

Post Layout Analysis

This is the part of the design cycle where there is the most pressure. It is also the part of the program where shortcuts can hurt the most. To avoid the temptation of shortcuts, the same environment used during pre-layout needs to be utilized by the post layout analysis. Simulation models, noise limits, waveform and timing measurement points all need to be the same as what was previously used. This needs to be applied to every net in the design and a comprehensive set of simulations and analysis needs to be performed. Waveform and Timing data are reported in a fashion similar to pre-layout, the difference being that the variations in post-layout are caused by real net-to-net variations with the assumed pre-layout rules.

Post-layout analysis should not be considered its own step in design implementation. By using an integrated methodology, post-layout becomes the validation of the pre-layout work and is used to identify areas where the post-layout differs from the routing rules. An example of where these differences may occur is in the device dispersion. Pre-layout simulations may assume that dispersion etch is insignificant or utilize a small length to account for it. In reality, some packages may require significant lengths of dispersion etch to escape from the package pin field.

Lab Results

This is where the success of a methodology gets measured. Figure 7 is the final timing analysis for the cache. Operation under worst-worst case conditions fails data hold margin by 27ps. This was deemed to be functional, though further refinement of the system environment and simulations limits could be done to determine a more realistic worst-case limit.

	A	B	C	D	E	F
1	Setup Margin (ns)	Hold Margin (ns)	Min Delay (ns)	Max Delay (ns)	Transfer Net	Driver
2	1.241	2.798	1.384	1.437	bc_tag_ctl	cpu
3	0.536	2.110	0.796	2.091	bc_addr	cpu
4	0.040	-0.027	1.030	1.177	bc_data	cpu
5	0.114	-0.014	0.924	1.059	bc_data	sram
6	0.698	0.607	1.544	1.929	bc_ctl	cpu
7	0.720	2.638	1.274	1.907	bc_addr_45	cpu
8	0.269	3.799	0.865	0.889	bc_tag_data	cpu
9	0.361	3.294	0.732	0.775	bc_tag_data	tag_sram
10						
11						

Extreme Delays | By Edge Variation | By Receiver | By Driver

Figure 7

Figure 8 shows the TT simulation results for the cache data and it's associated input clock. Figure 9 represents the data collected from the lab. Spice simulations are measured at 0.95V. The corresponding point in the lab measurements is the center gridline. Lab measurements were made at the midpoint of the plateau and correlate well with the spice data. The spice waveforms show a setup time of 800ps and a hold time of 400ps. A review of the lab waveforms shows setup time of 700ps and a hold time of 500 ps. Both spice and the lab measurements have symmetric plateaus on the clocks and offset plateaus on the data. Spice measurements were made at the IC pad, while lab measurements were at the device pin.

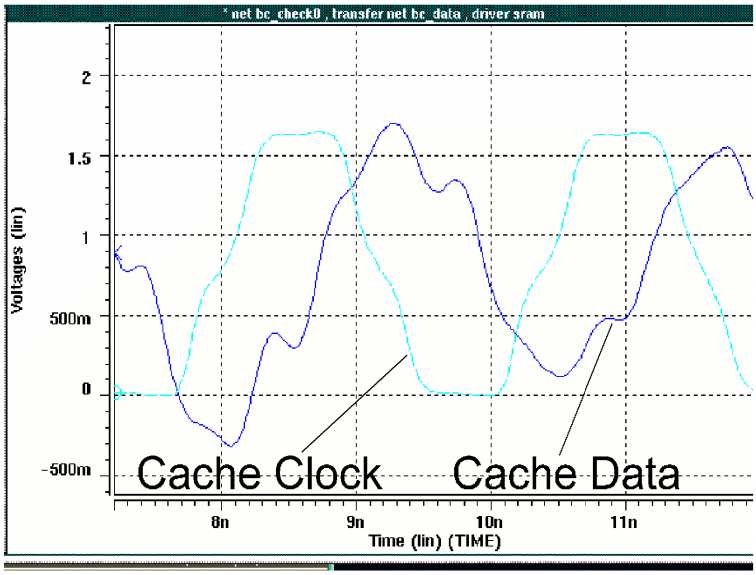

Figure 8

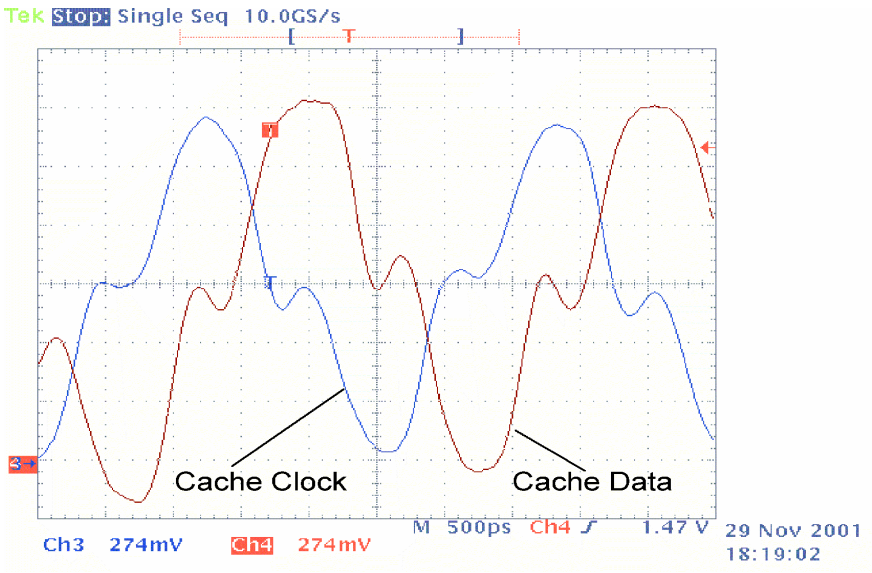

Figure 9

Figure 10 is the final timing analysis for the IPS bus. Extracted post layout data for each net has been analyzed and at a 500MHz datarate, there is 250ps of margin. By simulation, the theoretical limit for this bus is 571MHz. Systems with this bus operating at 500MHz are currently in production.

	A	B	C	D	E	F
1	Setup Margin (ns)	Hold Margin (ns)	Min Delay (ns)	Max Delay (ns)	Transfer Net	Driver
2			4.41	4.73	sys_addr_inclk	cchip
3			4.44	4.69	sys_addr_outclk	cpu
4			4.52	4.84	sys_data_inclk	dchip
5			4.49	4.71	sys_data_outclk	cpu
6	0.25	0.28	3.57	3.78	sys_data	cpu
7	0.53	0.37	3.52	3.86	sys_data	dchip
8	0.54	0.35	3.48	3.74	sys_addr_in	cchip
9	0.43	0.38	3.55	3.75	sys_addr_out	cpu
10						
11						

Extreme Delays | By Edge Variation | By Receiver | By Driver | By Edge

Figure 10

Figure 11 shows the TT simulation results for the IPS bus and it's associated input clock. Figure 12 represents the data collected from the lab. Spice simulations are measured at 0.95V. The corresponding point in the lab measurements is the center gridline. Lab measurements were made at the midpoint of the plateau and correlate well with the spice data. The spice waveforms show a setup time of 900ps and a hold time of 1100ps. A review of the lab waveforms shows setup time of 1000ps and a hold time of 1000 ps.
ISI play's a large part in how the waveform appears. The spice data use aggressive patterns to generate the worst case ISI affects. In the lab, it is difficult to generate the same ISI traffic on the IPS bus needed to match the spice simulations. Spice measurements were made at the IC pad, while lab measurements were at the device pin.

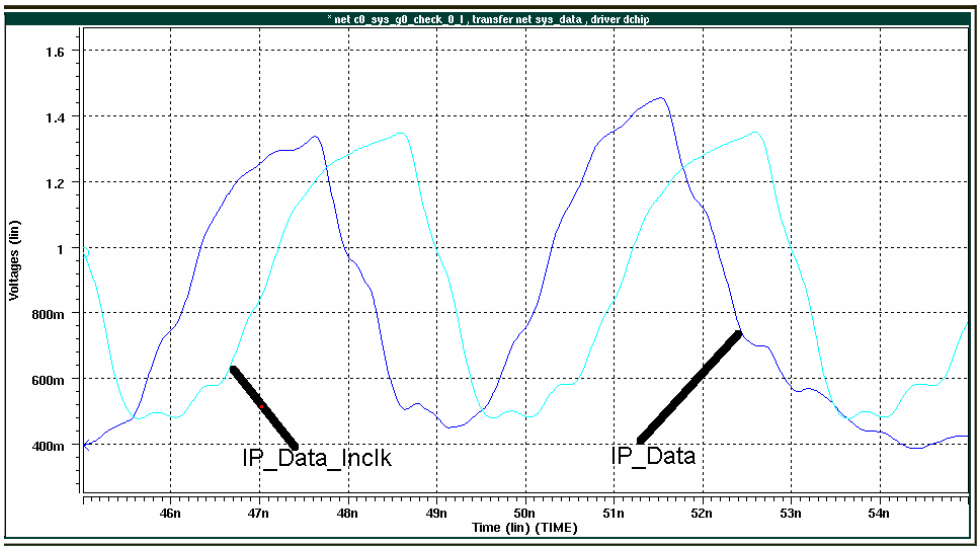

Figure 11

SiSoft

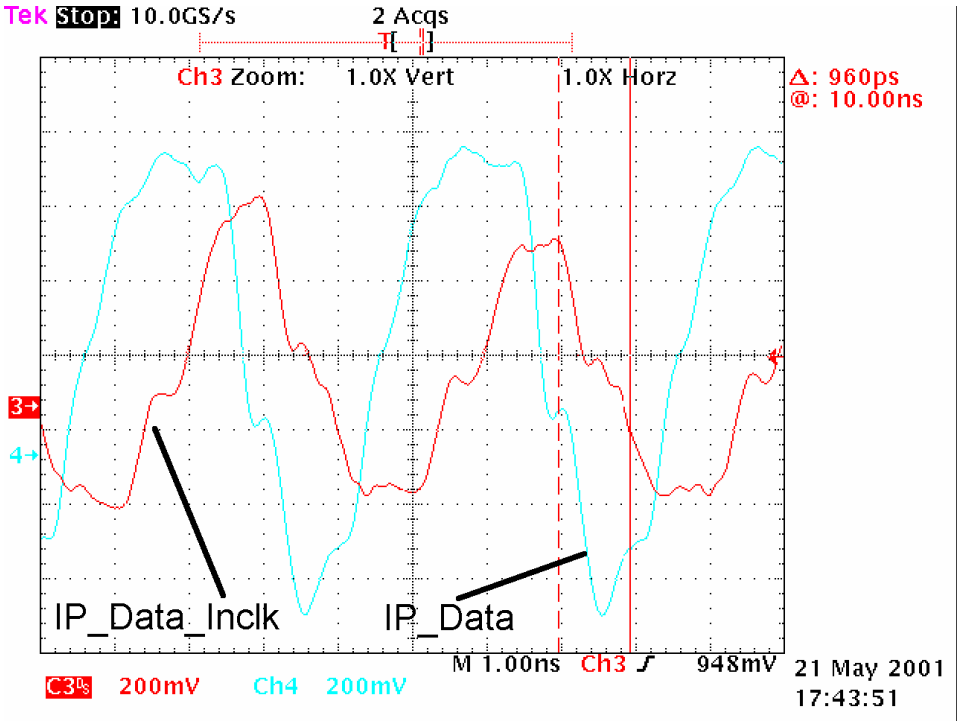

Figure 12

Summary

Source synchronous design requires the generation of a large database of simulation results that places a large burden on the designer who needs to interpret this data. A design methodology that allows a designer to quickly process the large data sets that are generated during the analysis of high-speed nets is presented as a solution for accurate and repeatable results. Example case studies of real problems encountered by designers were utilized to show the complexity of the source synchronous problem and how this methodology solved these problems. Finally, laboratory measurements of the signals in the case studies were presented that show correlation between the simulated and measured environments. The design examples showed real design problems that previous methodologies had failed to expose. To solve these types of issues, a new approach was needed. This lead to the development of the SiAuditor methodology, which has been proven to be a key tool in the development of high-speed networks.