Xbox360$_{TM}$ Front Side Bus – A 21.6 GB/s End-to-End Interface Design

David Siljenberg, Steve Baumgartner, Tim Buchholtz, Mark Maxson, Trevor Timpane
IBM Systems & Technology Group
Jeff Johnson
Cadence Design Systems, Inc.

Abstract – With a bandwidth of 21.6 GB/s, the Front Side Bus (FSB) of the Microsoft Xbox360$_{TM}$ is one of the fastest, commercially available Front Side Bus interfaces in the consumer market. This paper explains the end-to-end system approach used in designing the bus that achieved volume production ramp 18 months after design start. The 90 nm SOI-CMOS CPU and 90 nm bulk CMOS GPU designs are described. The chip carrier, circuit board, and signal integrity analyses are described. The design approach used to achieve high volume, low cost, and short development time is explained.

I. Introduction

This paper describes the design and performance of the Front Side Bus (FSB) used in the Xbox360$_{TM}$ video gaming console. The CPU design is described in [1]. Section II gives an overview of the FSB function and requirements. Section III explains the design of the physical (PHY) interface circuitry that attaches the chips to the channel. Section IV explains the channel implementation and provides the results in the context of the end to end design approach [2]. Section V summarizes the FSB performance and Section VI provides a conclusion.

II. FSB Overview

Figure 1 shows the architecture of the FSB. It is a duplex point to point source synchronous link.

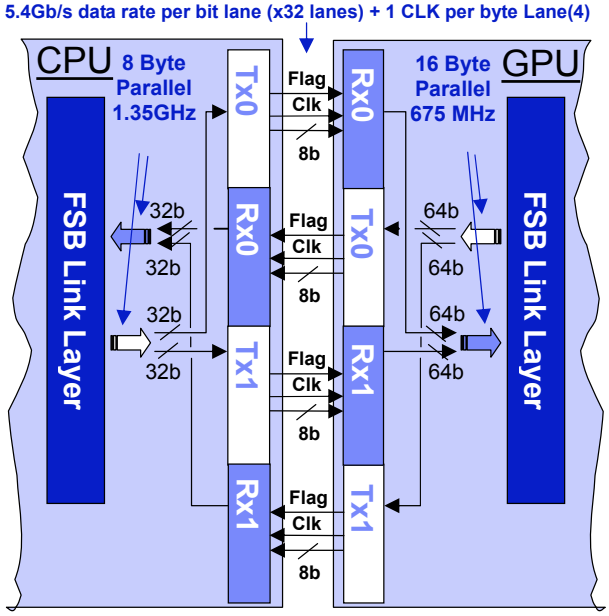

Fig. 1. FSB Block Diagram

The FSB is the only bus in or out of the CPU. This means that bandwidth and latency were high priorities in the design. Given the short path (70 mm) and low latency required the data is not coded or scrambled.

III. PHY Description

Two physical interface cores (PHY's) were developed with similar architectures. IBM designed the CPU PHY in 90 nm SOI process. The GPU is implemented in 90 nm TSMC bulk technology. IBM contracted with Cadence Design Systems, Inc. to design the GPU PHY.

The PHY's were designed with a source synchronous architecture. This allowed the use of a much simpler clock alignment algorithm and a more relaxed jitter budget. It enabled the use of non-coded data which lowers the overall latency of the system.

The main function of the transmitter is to provide a serialization of packet data sent from the FSB link layer and to drive it differentially on the channel. It also sends a clock to latch the receive data. The CPU Transmit (TX0 and TX1) cores receive this data in 4b parallel streams along with an associated clock edge that launched the data. The parallel data is then serialized and sent, along with a synchronous clock, either off-chip to the GPU receiver (RX) or to corresponding on-chip RX cores in an internally wrapped test mode. Drive levels and pre-emphasis are adjustable with register control.

The receive section amplifies, latches and deserializes the incoming data. Since there are several bit times of skew between bit lanes the receive clock must be phase shifted to the middle of the data windows in the receiver. This is done with phase rotators in each of the receiver channels. After the data is deserialized, it is transferred to the core logic through an elastic interface that absorbs the asynchronous phase relationship between the received clock and the core logic clocks.

The PHY contains a digital core that interfaces between both the analog transmit/receive sections and the FSB link layer logic. The digital core primarily includes register functions, elastic buffering, test support, and custom transmit/receive data distribution circuits. The PHY makes extensive use of control registers to control and monitor PHY functions such as driver amplitude and clock alignment.

The PHY designs used a balanced approach to achieve the system requirements. For example, Electrostatic Static Discharge (ESD) requirements dictated higher return losses than are traditionally used in high speed links. Since one party controlled all aspects of the circuits and channel,

1-4244-0630-7/07/$20.00 ©2007 IEEE.

appropriate engineering tradeoffs could be made to maintain margin in the overall link.

A. CPU PHY

Figures 2 and 3 show the block diagrams of receiver and transmitter sections. All the shaded blocks outlined in blue are implemented in Current Mode Logic (CML). The non-shaded blocks outlined in black are constructed of static CMOS or differential cascode voltage switch (DCVS) circuit topologies. All the jitter sensitive circuitry running at 2.7GHz and above is designed on the Analog VDD power domain which has a range of 1.0 to 1.15 volts. The Core VDD domain is shared with the CPU Core Logic and has a range from 0.8V to 1.3V depending on the process parameters of the FET's. This requires a translation circuit to transmit and receive signals across the power boundary.

B. CPU PHY Receiver

The receiver, shown in Figure 2, is constructed of 3 major functional blocks, the Data Slice, Clock Slice and Clock Dist. In each receiver byte lane, the Data Slice block is instantiated 9 times, once for each of the 8 data bits and a ninth time for a flag signal. The flag is used by the FSB logic to delineate packets. The Clock Slice and the Clock Dist are each only instantiated once for each receiver byte lane. The Clock Slice contains the circuitry which receives the differential clock from the GPU. The preamplifier used in the Clock Slice is the same as the preamplifier used in the Data Slice. It has a fixed 1dB of de-emphasis and midband gain of 6dB. The output of the preamplifier is converted into 2 clock signals that are 90 degrees out of phase by the Poly Phase Filter block. The 2 clock signals are commonly referred to as I & Q clocks (In-phase and Quadrature). The I & Q clocks fan out to 2 blocks. In normal functional mode, the 4:2 Test Mux will select the Poly Phase Filter outputs and pass them on to the Clock Dist block. There is a test mode which switches the 4:2 Test Mux to the outputs of a Phase Rotator. This Phase Rotator is controlled by a state machine that can slowly rotate the I & Q clocks through all of their possible phases for complete functional test coverage of the downstream deskew system. In addition to rotating slowly through all possible phases, the state machine can modulate the I & Q phases every system clock cycle to create a simulated high frequency jitter scenario for jitter tolerance characterization.

The Clock Distribution is implemented with a balanced H-tree topology to generate the copies needed for the 9 Data Slices. All the layout was hand placed and hand wired to control the characteristics of the wires and shields. Thicker metal layers at higher levels were used to reduce the wiring resistance and capacitance. The wiring of the clocks is very important to minimize the skew between the I & Q clocks. Any skew between the I & Q clocks directly reduces the jitter tolerance of the data recovery system.

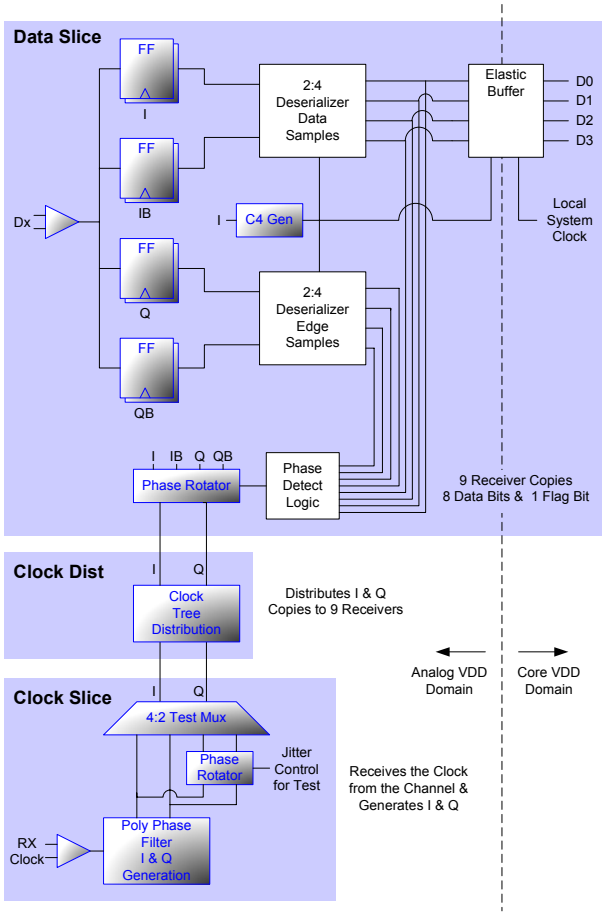

Fig. 2. CPU Receiver Block Diagram

The Data Slice blocks are functionally the most sophisticated circuits in the receiver. All 9 data slice instantiations operate independently because the skew between any received data bit is uncorrelated. The preamplifier has the same characteristics as the preamplifier used in the Clock Slice. The data is amplified and distributed to 4 master slave flip-flops. These flip-flops are the primary sampling latches for the data recovery system. The decisions made by these latches are deserialized by the 2:4 Deserializer blocks. The 4 bit nibbles produced by the I and IB clocks correspond to the data samples and the 4 bit nibbles produced by the Q and QB clocks correspond to the edge samples. The Phase Detect Logic block processes the results of the data samples and edge samples and determines whether the I, IB, Q and QB clocks are sampling too early or too late. The Phase Detect Logic issues the appropriate code to the Phase Rotator block to correct the sampling phases. This data recovery system has enough digital filtering to guarantee stable phase alignment under all data patterns and jitter conditions.

The phase rotators can be bypassed. This allows the interface to be run at 1 GB/s. At this speed skew, critical analog timings, and signal integrity issues are unimportant. This mode allows hardware debug of other parts of the system to proceed while PHY debug proceeded in parallel.

The Elastic Buffer block is a CMOS circuit driven by 2 clocks that are the same frequency but have an arbitrary phase alignment. The data samples are written into a parallel register on the edge of the Write clock which has a known phase relationship to the data. The Read clock unloads the register in the middle after a synchronization step has been invoked. The phase of the 2 clocks is allowed to wander apart in phase by ½ the width of the parallel register and the read clock still reads valid data.

C. CPU PHY Transmitter

A block diagram of a TX core is shown in Figure 3. Each TX core is functionally divided into transmit slices which support 8 data ports, 1 flag port, and 1 clock port. Additionally, each TX core provides its own bias circuitry, clock distribution, and clock generation.

Each data or flag slice is responsible for serializing a 4b parallel data stream and sending a serial bit stream across the channel or internally wrapped to the CPU receivers. The clock slice shares common clock inputs with the data and flag slices, but does not have need for a 1.35GHz clock distribution or a FIFO. The data and flag bits are transmitted serially along with a 2.7GHz clock emanating from the clock slice.

The transmit core must deal with multiple clock frequencies and phase relationships. The full rate 5.4 GHz clock is received differentially from a PLL outside the PHY. Clock generation internal to the TX core uses frequency dividers to create synchronous half-rate (2.7GHz) and quarter-rate (1.35GHz) clocks.

Transmit data from the FSB link layer is referenced to a quarter rate grid clock shared by the FSB logic. Although this grid clock is derived from a common PLL there is an unknown and varying phase relationship between the grid clock and the full rate PLL clock. Since transmit data is sourced from the grid clock but serialized using full rate clocks, an asynchronous interface exists which must be handled. This is performed by sending a frequency divided 675MHz version of the grid clock, referred to as the "data sample", along with the transmit data to the TX core. Since this "data sample" clock and the link layer data share the same timing relationship the Clock Synchronization block oversamples the "data sample" clock with the full rate PLL clock. This can be used to create grid-phase synchronous versions of quarter-rate and half-rate clocks for transmit-FIFO serialization allowing the transmit data from the link layer to be captured correctly. For this synchronization technique to be successful it is critical that bit to bit skew is minimized when forwarding the transmit data and "data sample". This handles the asynchronous interface without the latency of an elastic buffer.

The TX FIFO and driver functions collectively perform the 4:1 serialization of the transmit data D0-D3. Inside the FIFO, bit pairs D0/D2 and D1/D3 are captured into two-entry queues using 2:1 steering muxes clocked by the quarter rate clock, (1.35 I). When the quarter rate clock is high D0 and D1 are latched into their respective queue positions by the half rate clock, (2.7 I). When the quarter rate clocks are low D2 and D3 are latched into their respective queue positions by the half rate clock.

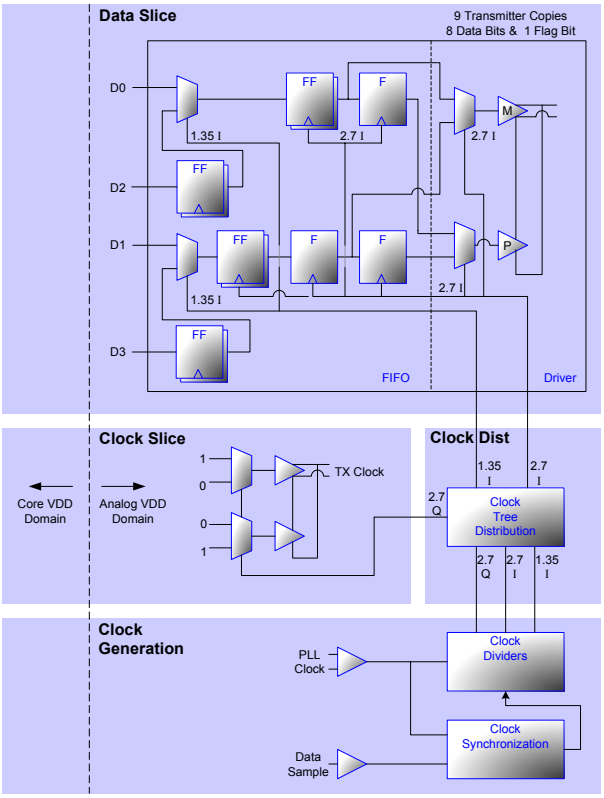

Fig. 3. CPU Transmit Block Diagram

These two-entry queues now represent two parallel data streams running at 2.7Gbps. A final 2:1 steering mux inside the driver alternately selects data from each FIFO queue using the half rate clock, (2.7 I), presenting data at the driver ports at the baud rate of 5.4Gbps. The main driver (M) is built from seven 2mA segments and a precompensation (P) segment that are selected from digital register controls depending on the drive levels required.

D. GPU PHY

The GPU PHY architecture differs from the CPU version in one main aspect. The GPU PHY is integrated into an ASIC that has a slower core clock than the custom designed CPU. As such the PHY has to provide 8:1 serialization and deserialization as opposed to the 4:1 ratio for the CPU.

The GPU PHY was designed by Cadence Design Systems, Inc. The largest challenge for the GPU PHY was the coordination of the design across a number of different design teams. These design teams existed in different companies and different physical locations. The GPU ASIC was designed by one company, the ASIC library by another, and the PLL by yet another. The system

requirement was a low latency, low jitter, 5.4 Gb/s serial link. To help with this coordination, a number of different types of models were generated and simulated but, as always, models alone are not sufficient to ensure success.

E. GPU PHY Transmit Interface

As in the CPU TX, latency is reduced by not using an elastic buffer for the parallel data Great care was needed in closing the timing on the transmit data bus going to the PHY, which was done by a 3rd Party. To aid in closing timing at the interface, timing models were developed for this interface. The creation of these timing models was challenging because of all the different circuits of the transmit path that needed to be analyzed. In addition to the analysis of the individual circuits, the worst case path and best case path had to be identified to accurately represent the timing path to the serializer FIFO.

Even with these timing models, closing timing on the interface was difficult because of a number of factors. First there was very little variation allowed between the data bits on the interface. Second, in a traditional digital environment, setups generally needed to be analyzed and fixed at the slow corner and the holds at the fast corner. The PHY is actually opposite of this because the clock alignment and data path inside of the PHY. The 3rd party received timing models of the PHY to aid in closing the timing. This seems rather straight forward at first, but in generating these timing models a number of parameters that are part of the design flow need to be understood for the timing models to be accurate in the 3rd party's design flow. The design challenges mentioned above are not captured by the timing models provided and required both extensive communication and a number of compromises in both designs to get a functioning interface.

F. GPU PHY High Speed Clock Interface

The PLL was designed by a different 3rd party. The output from the PLL was a 5.4 GHz clock. The clock has to be high frequency with very low jitter, meaning that transfer of the clock from the PLL to the PHY has to be seamless.

Once again a number of things had to be provided from each company to ensure that this worked as expected. Providing specifications at the interface was not sufficient to solve the problem. To lower the risk and maximize the opportunity for success, each company provided a model of what the other company would see. The PHY team received a model of the output stage including wiring of the PLL and the PLL team received a model of the input stage of the PHY including wiring. These models allowed the respective teams to do the analysis with the expected loads.

At these speeds and process geometries, device matching is an important variable in the system performance. A system that looks perfect in simulation could have significant performance degradation due to mismatches at the interface. This potential problem was minimized by using identical CML buffering on both sides of the interface allowing both teams to account for these mismatches in their respective designs and ensuring that there were no surprises when the silicon came back.

G. GPU PHY Development Issues

At times the technical issues mentioned appeared easy to solve in comparison to some of the logistical issues. The process chosen was being offered only to the GPU ASIC design company. Originally, this company was not allowed to provide the process IP needed to the PHY designers. Once those communications were established, the legal aspects of actually providing the IP still had to be considered. As with any aggressive schedule, the different pieces required to do the design was constantly changing. Revision control can be difficult, but add the problem of keeping track of not only the latest version, but what version is going to be used for a particular release of the mask set and it can become overwhelming. Multiply these problems by the number of companies providing design files and it can be seen that the logistical challenge was as difficult to solve as the technical issues.

In any system design, great care is needed to assure the startup sequence is robust. Accomplishing this task, however, took an extraordinary amount of communication and analysis. It required 4 different companies to analyze and evaluate all the different scenarios possible once the power was applied and the reset released. This included scenarios such as which end of the link, the CPU end or the GPU end, becomes active first. When the GPU end became active first, an error occurred during the training. Even though great care was taken and many timing diagrams were drawn, robust startup sequences invariably require thinking about the event or set of events that isn't expected, but is possible. In addition, with 4 companies involved there are a large number of dependencies that sometimes are not communicated. For example, in coming out of reset, the PLL team might not think it is important to communicate that when the PHY is released from reset, the clock may or may not be active. If the PHY team needs that clock to properly reset, then there is a scenario where the part will not start up. If any one detail is communicated to 3 of the teams, but not to the forth one, then problems can occur.

IV. Channel

A. Introduction

Presented herein are several considerations relative to definition and analysis of the Front Side Bus (FSB) I/O channel. Model correlation was achieved using two different simulation engines each with different model formats. These results were further compared to simulation results using an end-to-end model that was directly extracted from the actual I/O channel hardware using a Vector Network Analyzer (VNA). Moreover, I/O link performance sensitivity results are discussed for various channel characteristics, such as impedance discontinuities,

bit pattern run length and frequency content, crosstalk, ESD diode capacitance, transmission line length variation and pre-emphasis levels

Designing high speed interface channels within the "low-cost, high-volume" consumer application environment presents unique challenges. Most notably, channel designers had to understand the variability of transmission line characteristics; construct features, and material sets. The use of multiple suppliers exacerbated this as process tolerances vary widely from supplier to supplier. This design had to accommodate the widest variety of chip, substrate, and Printed Circuit Board (PCB) tolerance variation to minimize scrap and thus save cost. Although the overall I/O channel is short, designing a robust channel proved to be a challenge due to characteristic impedance variability, discontinuities, and crosstalk [2].

B. Channel Description

The I/O channel (Figure 5) packaging technology consists of standard "build-up" laminate technology for the CPU and GPU chip carriers and 2S2P "ultra low cost" PCB technology. Chip carrier wiring consists of stripline transmission line structures in GX3 dielectric material employing standard BGA anti-pad design and core vias that optimize return loss and pair-to-pair isolation. System PCB wiring consists of microstrip transmission line structures with 4mil trace widths and 6mil spacing within a differential pair. Pair-to-pair spacing was adjusted to 26mils to meet the isolation requirement. The electrical characteristics for the laminate chip carrier and PCB wiring are shown in Table 1.

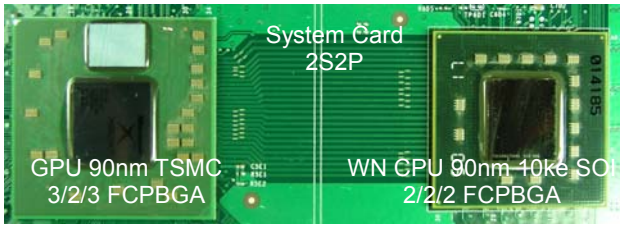

Fig. 5. FSB channel

Table 1. FSB design specifications

Electrical Characteristic	Chip Carrier Wiring	PCB Wiring
Differential Impedance	100 +/- 18 Ohms	100 +/- 15 Ohms
Differential Attenuation	-1.7dB @ $f_{Baud}/2$	-1.5dB @ $f_{Baud}/2$
Return Loss	-10dB @ $0.75*f_{Baud}$	N/A
Pair-to-pair Isolation	-30dB @ $0.75*f_{Baud}$	-30dB @ $0.75*f_{Baud}$
Maximum Length	18mm	50.8mm

C. Modeling and Simulation Approach

While considering the modeling and simulation technique of a 5.4Gbps FSB design, the validity of the standard simulation approach of combining lumped-element models for the channel interconnects and lossy, frequency-dependent transmission line models for the wiring constructs required reexamination. Figure 6 shows the various model elements associated with the FSB channel. Where, V(1-4) refer to FC-PBGA microvias, PTH refers to the FC-PBGA resin filled plated-through hole vias (RFP), W2 and W5 refer to the FC-PBGA wiring, SB refers to the FC-PBGA solder ball, VC refers to the PCB vias, and CW refers to the PCB wiring.

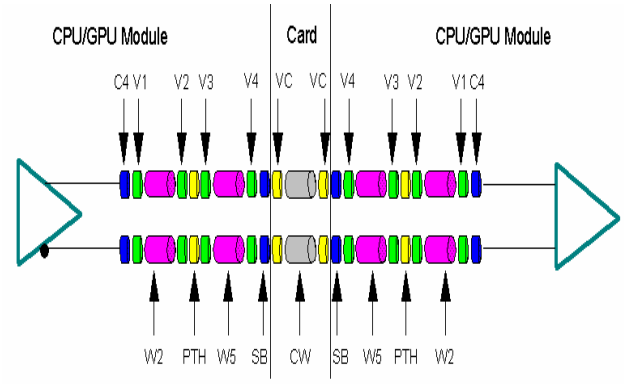

Fig. 6. FSB link model elements

In addition, acceptance of design criteria at specified link boundaries required the correlation of simulation results between the GPU and CPU PHY teams. This was a significant challenge as the various design teams historically use different simulators and model formats. Proper correlation required the establishment of several correlation points to ensure the quality of work by both design teams. First, the distributed, causal, lossy frequency dependent RLGC [3] and lumped models were correlated with scattering-parameters from full-wave model simulation to ensure that the distributed and lumped model approach would be valid for the frequencies and transition times associated with the FSB [4]. Second, traditional SPICE time-domain simulation approaches were verified across multiple model sets and circuit simulators to ensure each design team was simulating the same interconnect link and its effect on the signaling topology. Finally, hardware measurements were taken to validate the models used in the simulators. S-parameters from the CPU C4 to GPU C4 were extracted and correlated to the S-parameters from the modeled FSB link. Figure 7 shows the comparison of the hardware extracted versus modeled insertion loss (S21) and return loss (S11) of the FSB channel. Red and grey are hardware extracted and modeled return loss respectively. Blue and green are hardware extracted and modeled insertion loss respectively.

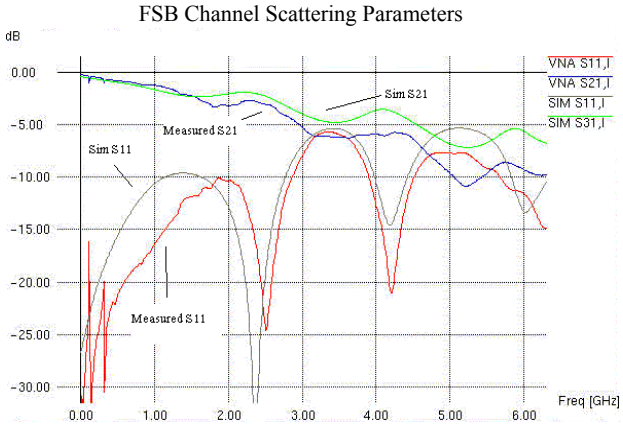

Fig. 7. FSB channel S21 and S11 correlation

The measured, full-link S-parameters were also used as link models in time domain simulations and the results were compared to the time domain simulations based on the pre-hardware models. Figure 8 shows that excellent correlation was achieved between Cadence Spectre$_{TM}$ with full-wave simulated S-parameter models, IBM PowerSPICE$_{TM}$ with frequency dependent RLGC models, and Cadence Spectre$_{TM}$ with hardware extracted S-parameter models as the 3 waveforms are nearly identical. The fourth curve, using a different SPICE simulator with the full-wave simulated S-parameter models does not show as accurate correlation as the other three waveforms although there is decent agreement in the rising and falling edges [4].

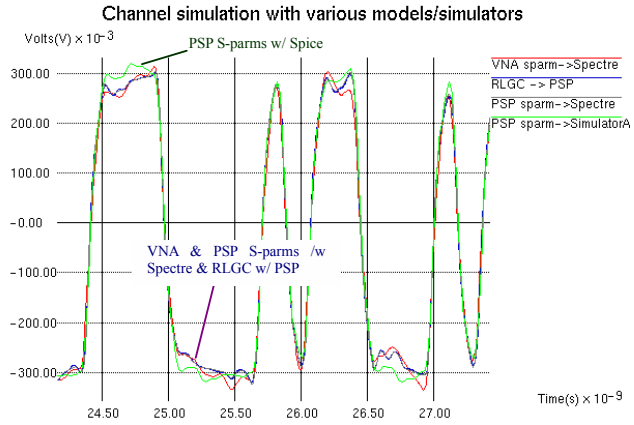

Fig. 8. channel model and simulator comparison

D. Channel Optimization Design Considerations

Once the channel models and simulation approaches were validated, end-to-end simulation examined the sensitivity of the design. Channel characteristics included in the sensitivity simulations were, bit patterns, impedance tolerance and discontinuities, capacitive loading due to ESD diodes, cross-talk, use of pre-emphasis, local odd-mode length mismatches at the package boundaries, transmission line length variations and power supply noise. This was accomplished through time-domain simulation with IBM PowerSPICE$_{TM}$ [5], using the validated RLGC wiring structures and lumped interconnect models.

Typical high-speed links are plagued by inter-symbol interference due to frequency loss characteristics of the I/O channel. The longer the transmission line, the more it is affected by attenuation. In these systems, the impact of impedance discontinuities is somewhat minimized by the fact that the reflected energy is attenuated below the noise floor by the losses associated with the longer transmission lines. Due to the short length of the FSB channel, reflections are more significant as discontinuities can produce destructive interference depending on the propagation delay and bit pattern dependence. Figure 9 illustrates eye opening variability across 32 impedance corners of the FSB channel for GPU/CPU carrier substrate wiring length of 18mm and system card wiring length varied from 0.5 – 4.0 inches. High (H) and low (L) impedance corners were considered for the CPU transceiver, GPU transceiver, CPU carrier wiring, GPU carrier wiring, and system card wiring.

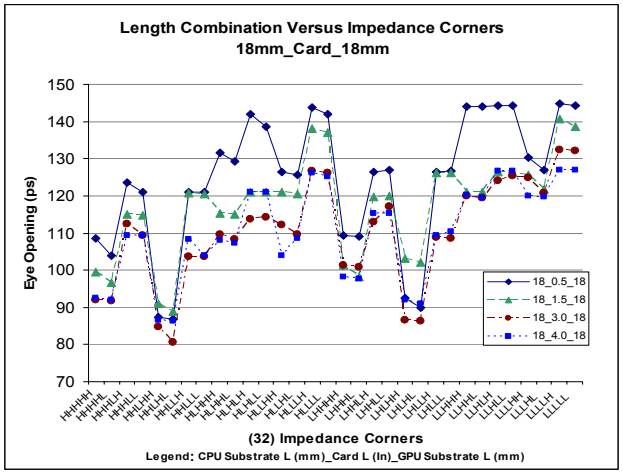

Fig. 9 Eye opening variability with impedance and length variation.

These effects were quantified in simulation by sweeping transmission line lengths of both the chip carrier and PCB around nominal values. Destructive effects were minimized by running exhaustive bit patterns and simulation in an IBM internally developed statistical communication theory based simulator, similar to StatEye$_{TM}$ [6] to identify the optimal channel delays. Traditional time domain simulation with frequency dependent RLGC models is not practical for bit patterns of more than a few hundred data bits.

The FSB channel has four obvious physical boundaries which could exhibit impedance discontinuities. These boundaries are: the driver and receiver termination to the laminate chip carrier, and the laminate chip carrier to the PCB at both ends of the link (at C4 and SB-VC respectively in Figure 6). In addition to possible loss in transmit amplitude; a ±15% manufacturing tolerance to the target impedance at one of these boundaries could result in a 15%

reflection of the signal. Even though dynamic termination was employed in the driver and receiver circuitry to reduce the impedance variability, the remainder of the FSB channel can still exhibit significant impedance discontinuities due the variations detailed in Table 2. All values listed are differential impedance with the exception of driver termination.

Table 2. Impedance variation

	Impedance [Ω]		
	Min	Nominal	Max
Driver termination	44.9	50.6	58.0
Driver package	85.4	93.5	109.7
PCB	85.0	100.0	115.0
Receiver package	85.4	93.5	109.7
Receiver termination	89.6	100.8	107.6

The capacitance associated with ESD structures in SOI technology can be significant. The tradeoff between ESD protection and capacitive loading was examined. Capacitive loading on a net causes edge degradation and decreases slew rates. At 5.4Gbps, increases in capacitance, as little as 100fF, produces decreased slew rates resulting in reduced signal swing leading to additional jitter and eye closure. The limit on ESD capacitance associated with the ESD device was determined to be 0.75pF to satisfy both the electrical design requirements and to provide adequate ESD protection.

The forward coupling coefficient associated with PCB microstrip structures shows that far-end cross-talk (FEXT) will accumulate until the cross-talk levels saturate due to channel attenuation. Based on other constraints in the design, physical isolation was the only means available to mitigate far-end cross-talk. A victim-aggressor spacing of 26mils was required to reach acceptable percentages of cross-talk (less than 3%).

Another design consideration related to the physical layout of the FSB channel was the 1mm offset between the positive and negative phases within a differential pair on the PCB because of the relative BGA I/O assignments of the GPU and CPU modules. This brief loss of odd-mode differential coupling affects each phase of the differential pair as they become less symmetrical to each other and more susceptible to other effects such as discontinuities and coupled energy. The effect of this offset was observed to be minimal in both edge degradation and eye opening reduction.

Once the channel's individual design sensitivities were examined and understood, realistic variances were combined to generate a realistic worst-case eye diagram. Given the current design environment, combining absolute worst-case design elements would result in an overly pessimistic eye diagram and perhaps un-necessary cost increases. Therefore, it is critical to understand realistic variations in all examined design considerations. For example: manufacturing supplier's tolerances and capabilities determines impedance values to be used for packaging. Knowing how any driver or receiver enhancements in the silicon (such as pre-compensation or dynamic termination) are implemented can improve the worst-case eye. Driving the isolation between traces in the physical layout and the assignment and signaling of the I/O helped determine how to mitigate the cross-talk effects. Figure 10 shows a simulated eye with nominal channel parameters (dark) compared to a worst case eye (light) generated using realistic worst-case effects, as well as the eye criteria (receiver mask was 90ps @ +/- 50mV) for the link.

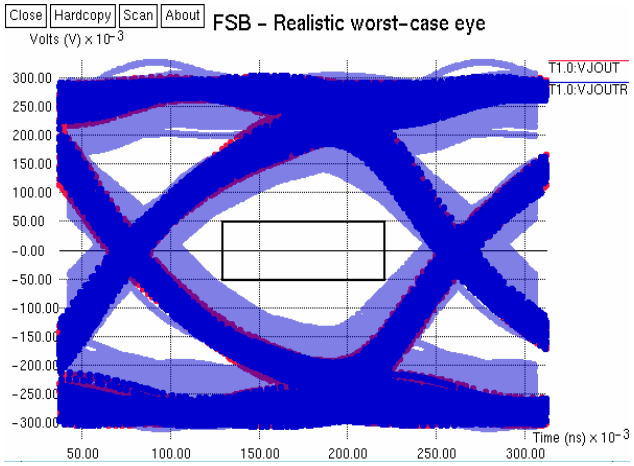

Fig.10. Realistic worst-case eye simulation

V. Bus Performance Measurements

This section summarizes the overall link measurements confirming that the end-to-end solution achieved sufficient margin. Figure 11 shows the transmitted eye diagram at 5.4 Gb/s. This was measured on a test board that brings all the signals to SMP connectors with high bandwidth cables. This measured eye compares very well with the simulated eye in Figure 10.

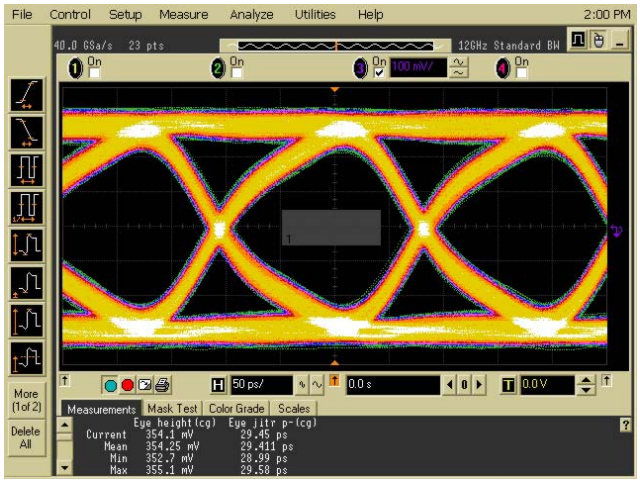

Fig. 11. Measured Transmitter Eye Diagram

The ultimate measure of link performance and margin is Packet Error Rate (PER) vs. clock position. This measurement encompasses on-chip transmitters and receivers, on-chip noise, channel impedance discontinuities, and channel cross talk. PER is similar to bit error rate except that it quantifies the error rate impact to a data bus more clearly. When a data packet is sent on the bus all that is important is the likelihood that the packet will be received correctly. A bad packet is just as bad with one error as it is with 10. In this test the FSB link layer (see Figure 1) creates 128 byte packets with correct headers, trailers, and random data payloads. The receiver FSB link layer checks the packets for errors. This is analogous to what happens when functional data is transferred on the link.

Figure 12 is a plot of PER vs. receive clock position. Each receiver for the 16 data bits has a Phase Rotator, as described in section III-B, that dynamically adjusts the clocks to line up with the data. To make the PER measurement the Phase Rotator is manually adjusted through 32 steps for one 185 ps bit interval. When the clock is too far from the center, packet errors are detected. For clarity in the printed document this graph shows the PER curves for 5 of the 16 bit lanes. This measurement can be made on any system and doesn't need a special test board.

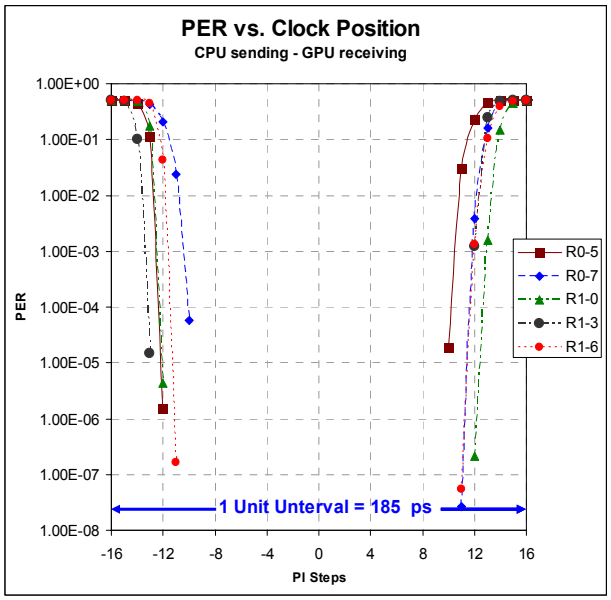

Fig. 12. Packet Error Rate (PER) vs. Clock Position

VI. Summary and Conclusions

By designing the FSB from a system perspective a high performance bus was designed with less resource and a shorter schedule than traditional approaches. Designing in a collaborative environment with multiple design teams was a necessary yet significant challenge. A testament to the capability of the effort is the fact the first pass CPU part was communicating with the GPU at speed 3 days after being powered up. Producing the Xbox360$_{TM}$ in high-volume only 18 months after project start highlights the success of the FSB channel design and verification process.

Acknowledgements

The authors would like to thank Andy Maki for his tireless characterization work, Ben Fox for his channel simulation and correlation work, Thomas Liang for end to end link simulations, Charles Geer for digital design, Dan Young for test ownership, Darryl Becker for package design, and Michael Connell creative for team leadership.

References

[1] Brown, J., "Application customized CPU design for Microsoft Xbox360$_{TM}$," *Fall Processor Forum Presented by In-Stat,* Oct. 2005.

[2] Maxson, M., et al "Xbox360$_{TM}$ Front Side Bus Development" *Signal Propagation on Interconnects 2006, IEEE 10th Workshop on*. May 9-12.

[3] Deutsch, A., *et al*, "Extraction of $\varepsilon_r(f)$ and tan $\delta(f)$ for Printed Circuit Board Insulators Up to 30 GHz Using the Short-Pulse Propagation Technique," *IEEE Transactions On Advance Packaging*, Vol. 28, No. 1 (Feb. 2005), pp. 4-12.

[4] Timpane, T., *et al*, "Multi-Gb/s I/O Link Model to Hardware Correlation," *Electrical Performance of Electrical Packaging 2005, IEEE 14th Topical Meeting on.* Oct. 24-26, pp. 155-158.

[5] PowerSPICE User's Guide, Version 1.5, IBM Corp., 1997

[6] www.stateye.org