

# DesignCon 2007

## Packaging a Supercomputer in a PCI Express Form Factor

Mark Bailey, IBM Technology Collaboration Solutions  
[mjbailey@us.ibm.com](mailto:mjbailey@us.ibm.com)

Greg Edlund, IBM Technology Collaboration Solutions  
[gedlund@us.ibm.com](mailto:gedlund@us.ibm.com)

Bob Morse, Mercury Computer Systems  
[rmorse@mc.com](mailto:rmorse@mc.com)

Ankur Patel, IBM Technology Collaboration Solutions  
[ankurp@us.ibm.com](mailto:ankurp@us.ibm.com)

## Abstract

We will look at the trade-offs between power, cooling, and performance involved in packaging the Cell Broadband Engine, multi-core graphics processor, and associated bridge chip on a PCI Express™ card together with large amounts of memory. In particular, we will examine ac and dc power distribution to over twenty domains, PCI Express compliance, and timing specifications for a DDR2 interface. Close collaboration among members of a multi-disciplinary team enabled the successful co-development of the card and bridge chip.

## Author Biographies

Mark Bailey is an Advisory Engineer in IBM's Technology Collaboration Solutions division. He has been an RF engineer for seven years and has worked on 10 Gbps optical-electrical transceivers for IBM and JDSU. His work experience includes 3D electromagnetic field analysis. Mark got his Bachelor of Science in Electrical Engineering from University of North Dakota.

Greg Edlund is a Senior Engineer in IBM's Technology Collaboration Solutions division where he has responsibility for modeling and operating margin analysis. He has eighteen years of experience in signal integrity engineering and IC design on servers, enterprise systems, and supercomputers at IBM, Digital Equipment Corporation, Cray Research, and Supercomputer Systems. Greg earned a Bachelor of Science in physics from the University of Minnesota Institute of Technology. When he is not sitting at his workstation, he enjoys biking, flying, creative writing, and reading in front of the fireplace.

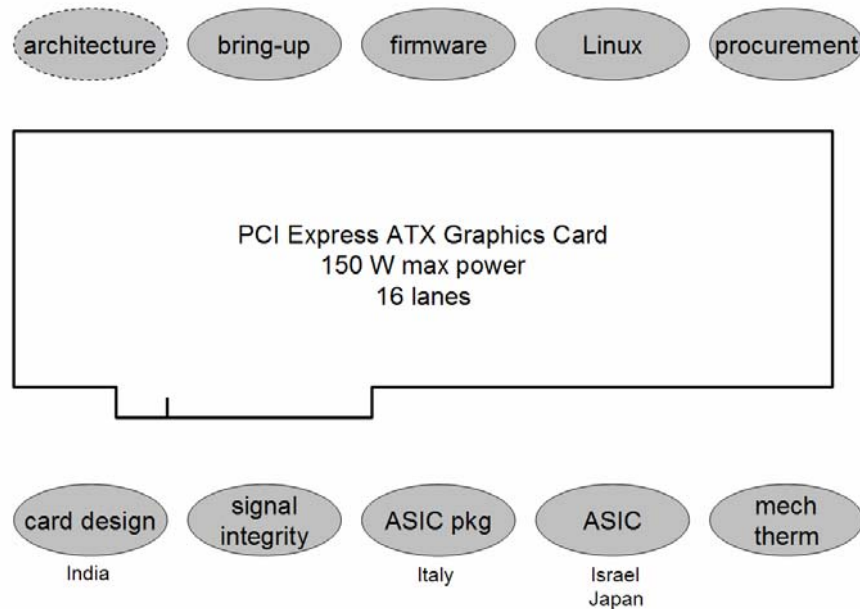
Bob Morse is a Principal Hardware Design Engineer at Mercury Computer Systems Inc, located in Chelmsford, MA. Bob has over twenty years of hardware design experience focused on the digital domain at Mercury and MIT Lincoln Lab. Bob holds a BA degree in Geography from University of Massachusetts, Amherst campus, and a BS degree in Electrical Engineering from University of Massachusetts, Lowell campus. A resident of Chelmsford, Bob enjoys time with his family, is on the town Planning Board, and is restoring a 1730's Colonial Farmhouse.

Ankur Patel is a Signal Integrity Engineer in IBM's Technology Collaboration Solutions division. He has three years of experience in signal integrity engineering on servers, flex circuits, and I/O cards. Ankur earned a Bachelor of Science in Electrical Engineering from Michigan State University in December 2003. Outside of work, Ankur enjoys sports, traveling and spending time with friends.

## Introduction

In these opening years of the 21<sup>st</sup> century we have observed the law of supply and demand operating on a global scale at an unprecedented level. Global competition has brought profound changes to engineering and manufacturing organizations across the world, and IBM is no exception. Along with these changes come new opportunities that would not have been accessible had events held to their previous trajectories. The Cell Broadband Engine (BE) microprocessor stands as an example of one of those opportunities that was the result of collaboration across corporate, cultural, and national boundaries. Industry leaders Sony, Toshiba, and IBM drew from their deep pools of engineering resources to assemble the broad spectrum of talent that was necessary to design the Cell BE, a high-performance graphics processor. Cell BE technology has, in turn, enabled a second generation of derivative products unrelated to the gaming industry.

In 2004 IBM Technology Collaboration Solutions met with a potential development partner, Mercury Computer Systems, to discuss a PCI Express<sup>TM</sup> (PCIe) add-in card targeted at imaging applications in the medical, semiconductor, seismic, and aerospace industries. The development team for the Mercury Cell Accelerator Board (CAB) comprised engineers from both companies, a wide range of disciplines, and five nations: India, Israel, Italy, Japan, and the United States of America. In the early months of 2005, engineers at Mercury and IBM engaged in a feasibility study intended to answer several key questions. Would all the components fit on a PCIe ATX graphics card? Would it be possible to get the power in and the heat out? Could we place and route the card? Would interfaces run at the desired speeds? How could we debug the firmware and operating system in time to meet the market window?



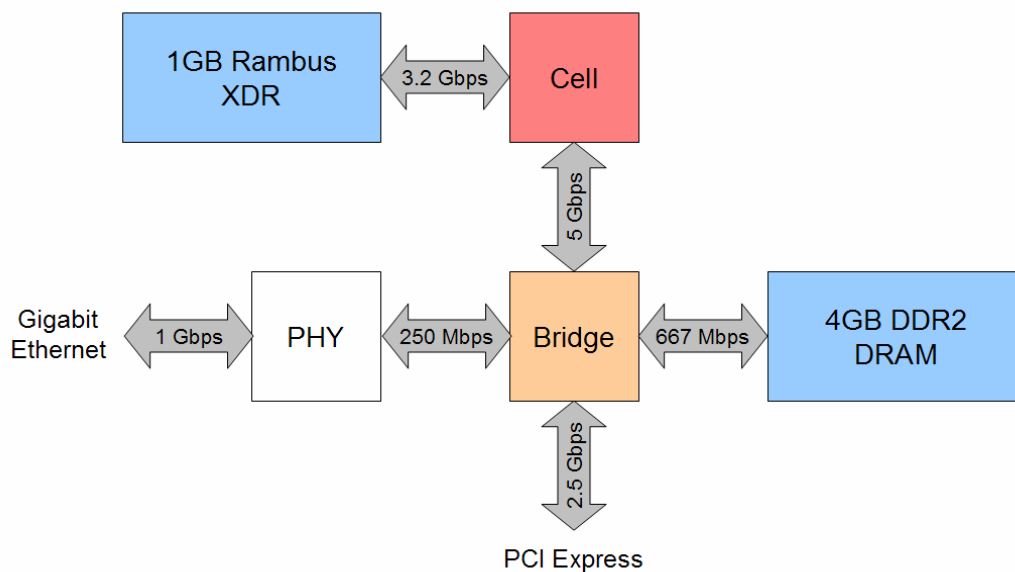
**Figure 1: Mercury CAB Engineering Team**

During the three-month feasibility phase of the project, representatives from many disciplines sat at the table deliberating these questions. The disciplines included firmware, operating system, procurement, card design, signal integrity, ASIC, ASIC packaging, mechanical, and thermal. We held frequent meetings to identify potential road blocks. Each member of the team had the chance to voice his or her concerns, explore how their resolution affected others in the room, and perform the analysis required to make a data-driven decision. The final conclusion was a go.

This paper tells the story of three critical areas of intense analysis and trade-off: power delivery, PCI Express™, and DDR2 memory. Each took place within the context of a demanding multi-tasking environment.

## Feasibility Study

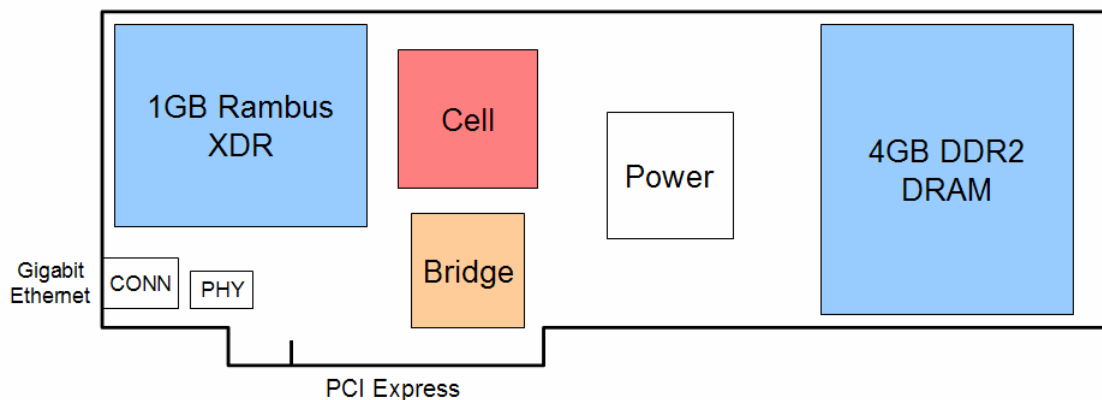
The functional block diagram in Figure 2 depicts the packaging challenge Mercury Computer Systems posed to the team: double the Rambus XDR memory from previous Cell BE implementations and concurrently design a bridge ASIC that would link Cell BE with 4 GB of DDR2 memory, PCIe, and Gigabit Ethernet. The eight parallel graphics processors within the Cell BE chip required large amounts of high-quality power delivered to a small space – a technical challenge common to supercomputer design. After the electrons had done their job, the cooling solution would need to sweep the heat away from the silicon. Another defining feature of a supercomputer is high-bandwidth access to memory and IO from the processor complex; the bridge chip satisfied this requirement. All this hardware had to fit within the three-dimensional confines of the PCIe ATX graphics card without the benefits of solid copper bus bars, liquid impingement cooling, or 200 mil thick circuit boards.



**Figure 2: CAB Functional Block Diagram**

During the early part of the feasibility study the engineering team faced the task of mapping the functional block diagram into a component placement that was consistent with the physical constraints of the card. At first we questioned whether there were even enough square inches on the card to accommodate the major chips, associated voltage regulators, and hundreds of smaller components! After settling this question, the next issue we wrestled with arose from thermal considerations. Cooling the Cell BE required a large volume of air moving across the card from right to left, and this implied a rather large blower located on the right side of the card. Since the clearance between the bottom of the blower and the top of the card was low, we placed the low-profile components under the blower: DDR2, flash memory, and H8 service processor.

Any components with heat sinks had to be located to the left of the blower, and this presented another conflict: the bridge chip wanted to be near the PCIe edge fingers *and* the DDR2 memory. However, if it were located beneath the Cell BE its heat sink would collide with those of the Cell BE and XDR memory. Moving the bridge chip near the DDR2 memory would stretch the length of the PCIe wires beyond the recommended limits. An in-depth analysis of the PCIe bus was outside the scope of the feasibility study. This led us to the next phase of the study, which involved defining the card stack-up in the context of a first-order PCIe loss budget.



**Figure 3: First Pass Placement**

PCIe design add-in guidelines recommend 5 mil lines at a maximum length of 3.5 in. [2] The card physical design team estimated it would take at least six signal layers (top and bottom not included) to route DDR2 memory – the most dense region of the card. Two 5 mil layers would have pushed the card thickness beyond the specified 62 mils. A spreadsheet PCIe loss budget satisfied us that 4 mil lines would indeed meet the PCIe specification in our application (see discussion of PCIe interface for further details). The bridge chip was another mitigating factor: the designers of the high-speed serial circuits worked for IBM’s Microelectronics division, and they characterized the silicon thoroughly in the lab. Furthermore we controlled the ASIC package wiring.

Along with PCIe considerations, ac and dc power distribution strongly influenced card stack-up. Knowing the Cell BE would draw somewhere between 80 A and 100 A of current, we used 1 oz. copper for the innermost two layers. This card utilized over 20

unique power domains, not including analog supplies. Obviously there were not enough layers in the card for each voltage to have its own plane, so we split some of the power planes among multiple domains and allowed the remaining low-current domains to overflow onto signal layers. Four dual stripline layers provided a home for the low-current power shapes and the extra wires needed to complete the high-density wiring between the bridge chip and the DDR2 DRAMs.

Signal referencing was another important consideration in defining the stack-up. The PCIe nets wanted ground-ground referencing. The Rambus XDR nets used either ground-ground or ground-VDD, and the Rambus Redwood nets used ground-VDD. These referencing requirements drove plane splits on certain layers of the card.

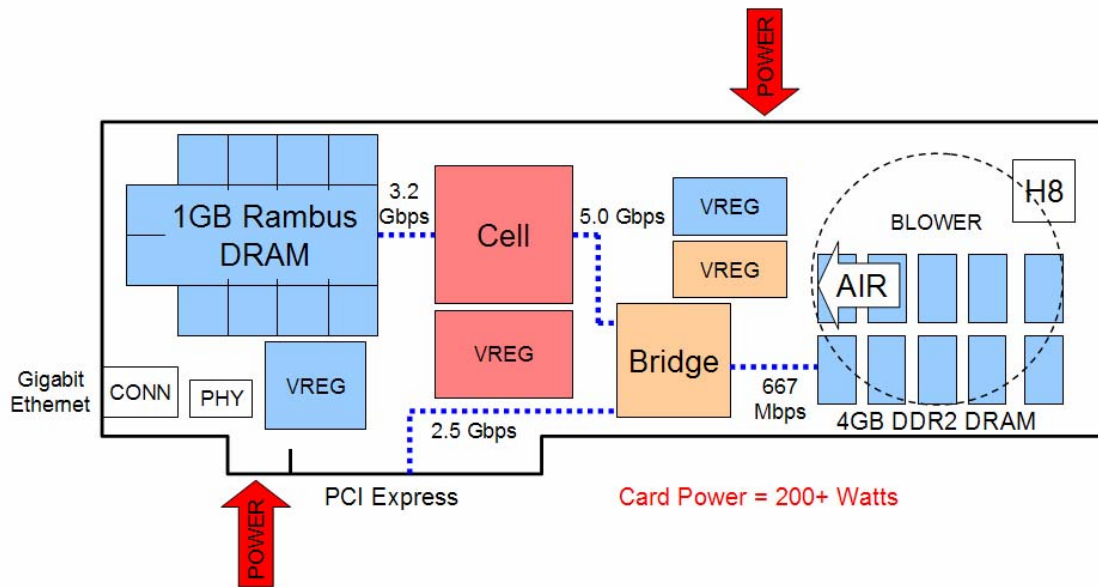
After considering all these factors together, we settled on an 8S6P (8 signal, 6 power) stack-up with two 4 mil stripline layers and four 3 mil dual stripline layers. The 4 mil stripline layers were 55  $\Omega$  single-ended, 100  $\Omega$  differential. The 3 mil dual stripline layers were 52  $\Omega$  single-ended, 96  $\Omega$  differential. Some of the XDR wires were wider to match the lower impedance of a forked net topology with heavy loading. We used the top and bottom layers for shapes and escape wiring only.

LAYER		T
TOP		1.9
	Fill	3.6
V2		0.6
	Core	4.5
S3		0.6
	Fill	5.0
V4		0.6
	Core	3.0
S5		0.6
	Fill	4.0
S6		0.6
	Core	3.0
V7		1.2
	Fill	4.5
V8		1.2
	Core	3.0
S9		0.6
	Fill	4.0
S10		0.6
	Core	3.0
V11		0.6
	Fill	5.0
S12		0.6
	Core	4.5
V13		0.6
	Fill	3.6
BOT		1.9
8S6P		62.9

**Figure 4: Card Stack-up**

With the PCIe, power, and stack-up issues resolved at a level appropriate for exiting the feasibility study, the design team finalized component placement and began wiring the card. The bridge chip moved slightly to the east of the Cell BE. We placed voltage

regulators adjacent to the chips they serviced. While both XDR and DDR2 DRAMs needed +1.8 V, we decided to use two local independent regulators to prevent contention in the region between the Cell BE and the bridge. The actual power demand for the card exceeded the 150 W allowed by the PCIe ATX specification. In addition to the power that enters through the card edge connector, we used a 10-pin +12 V connector to supply the extra power from a location northeast of the bridge chip.



**Figure 5: Final Component Placement**

In retrospect, two areas of the card design stand out as high risks during the feasibility study. The Rambus XDR memory was entirely new to the team. Our unfamiliarity with the bus and the complications with a forked net topology gave us good cause to contract this work out to Rambus. In the end this turned out to be a sound decision as it significantly reduced the risk on an already risky program. Rambus was flexible and a pleasure to work with. The DDR2 memory, while slower, was also a high risk. We had experience designing DDR2 interfaces, but never with such severe wiring limitations. As it turned out, we were able to make all the connections after a month of manual routing, but it was touch and go. It would have been prudent to do a DDR2 wireability analysis during the feasibility study to further reduce risk.

## Power

In the grand scheme of electrical packaging, delivering reliable power to circuits is of primary importance. To guarantee that these circuits will function as advertised, ac and dc power fluctuations must remain within the power supply tolerances specified by the component datasheets. IBM's approach to this problem divides the frequency response of a power distribution network into three bands: low-frequency (dc to 100 MHz), mid-frequency (100 to 500 MHz), and high-frequency (500 MHz and above). While these

numbers may vary by application, they represent a noise containment strategy: the IC should contain the high-frequency noise and the IC package should contain the mid-frequency noise. Low-frequency noise and dc drops are present everywhere.

By the time our team began working on this new application of the Cell BE, the Sony-Toshiba-IBM design team had already done a thorough job of optimizing the high- and mid-frequency bands of the Cell BE power distribution network. They characterized the IC package in the lab and clearly specified the current vs. time demand and allowable voltage tolerances. This level of detail is not commonly available for other ICs on the market, but it is essential for a high-performance processor running at low voltage. We optimized the low-frequency and dc regions of the power distribution network using the information provided by our colleagues at the Sony-Toshiba-IBM Design Center.

Before the calculation of impedance profiles and dc drops began, we went through an exercise of mapping the sources and sinks in the power distribution network. Some of this occurred during the feasibility study, but the majority of it took place after placement and before routing. If we had waited until after the physical design team started drawing artwork, we would have found the degrees of freedom to be significantly fewer. Yet this exercise could not occur in a vacuum either. Signal referencing was an equally high priority, especially for clocks, so we carried out a similar mapping exercise for clock sources and sinks. The space between the Cell BE and the bridge chip was a region of high contention between power and clocks for limited copper resources. This meant that the signal integrity engineers responsible for power and clocks had to collaborate closely with each other and with the physical designers.

The power mapping exercise addressed the following questions:

1. Where were the primary points of entry for power?
2. What were the optimum locations of voltage regulators relative to their loads?
3. How much copper was required to carry the current?
4. Would the copper reside on a power or signal layer?
5. Could we concurrently satisfy the requirements of power and signal referencing?

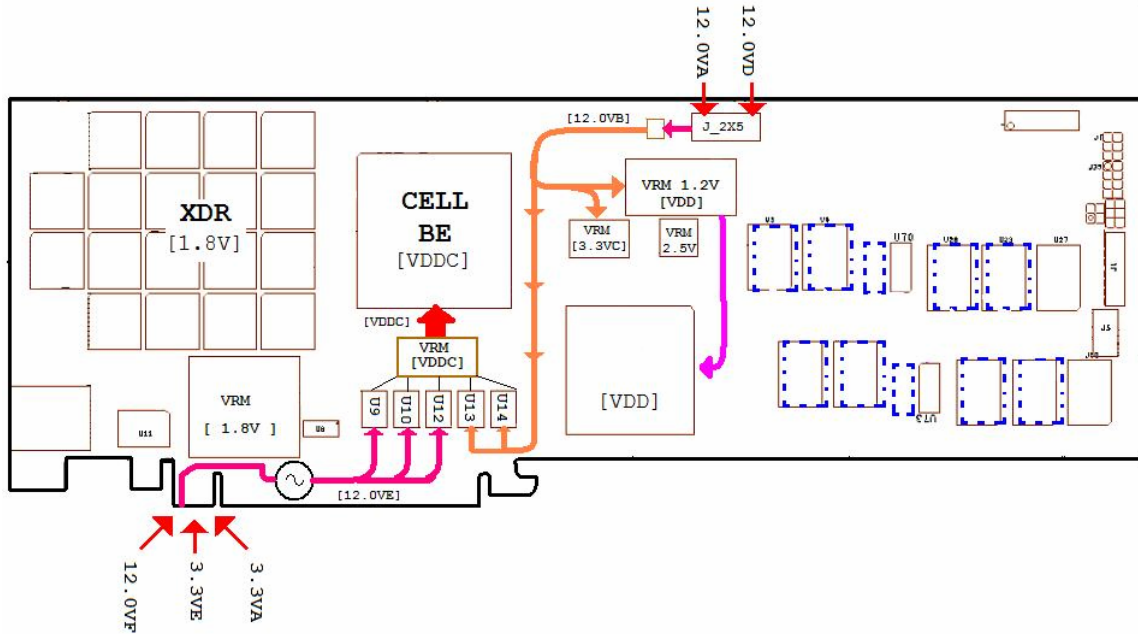
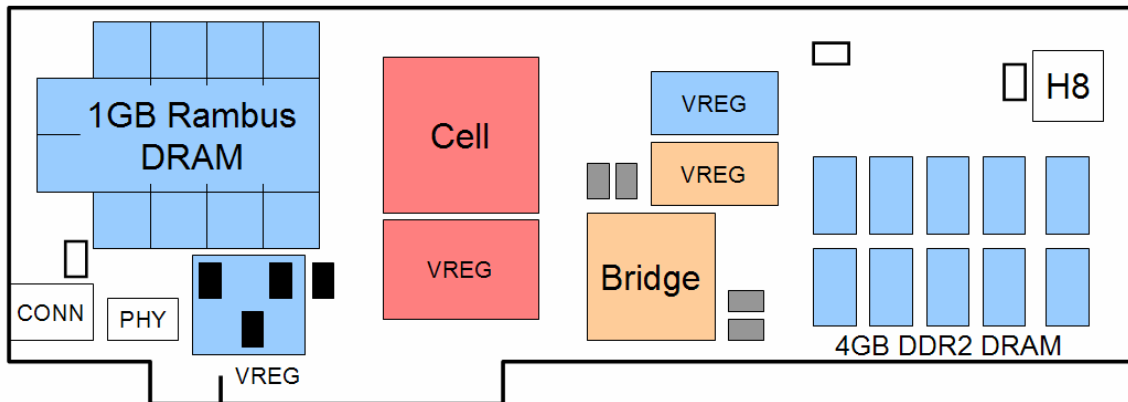


Figure 6: Power map



KEY: ■ = Rambus & Cell clocks      ■ = Bridge clocks      □ = Misc. clocks

Figure 7: Clock map

Once we finished our first pass of the power and clock maps, it was time to get down to the business of dc and low-frequency analysis and physical implementation, which was an interactive process. For the majority of our dc drop analysis we used a spreadsheet estimator for that was based on resistivity, geometry, and prevalence of antipads. For more demanding supplies, the PowerDC tool from Sigrity, Inc. allowed us to calculate voltage drops and current densities to a greater degree of accuracy.

PowerDC provided us with several advantages. We were already familiar with the user interface from using Speed, and we found that we were able to quickly import shapes

from our card layout database. Calculations took only a handful of minutes. This quick turn-around enabled us to work interactively with the card physical designers and arrive at a solution that satisfied power, referencing, and real estate. (We were able to validate the accuracy of the tool when the assembly house inadvertently rotated an EEPROM, thereby creating a power-ground short that dropped some 250 mV – a quantity that was easy to measure accurately!) One important benefit was the capability to calculate via currents, which we were unable to do with our spreadsheet. PowerDC identified a few vias that exceeded our current limits, and it was not immediately obvious why. After confirming the results using hand calculations and SPICE, we were able to understand the pattern of current distribution and formulate new ground rules for via placement.

Lab measurements showed a dc drop of 25 mV across the Cell BE pin field. A sense point in the pin field zeroed out the drop between the regulator and the Cell BE. The combination of dc drop and low-frequency ac noise was 69 mV – within the specified tolerance for the Cell BE core supply (see Table 1). Two of our ac noise measurements fell outside of the limits, and we had to decide how to proceed. Noise was 20 mV high on +1.2V, which is an IO rail for the two Rambus interfaces. Since our eye openings on these two interfaces were in the range of 70% – 80 % we decided not to change the design. However, 20 mV of extra noise on an analog VDD was another story. This warranted a closer look. At the time of this writing, the final power measurements were still in progress.

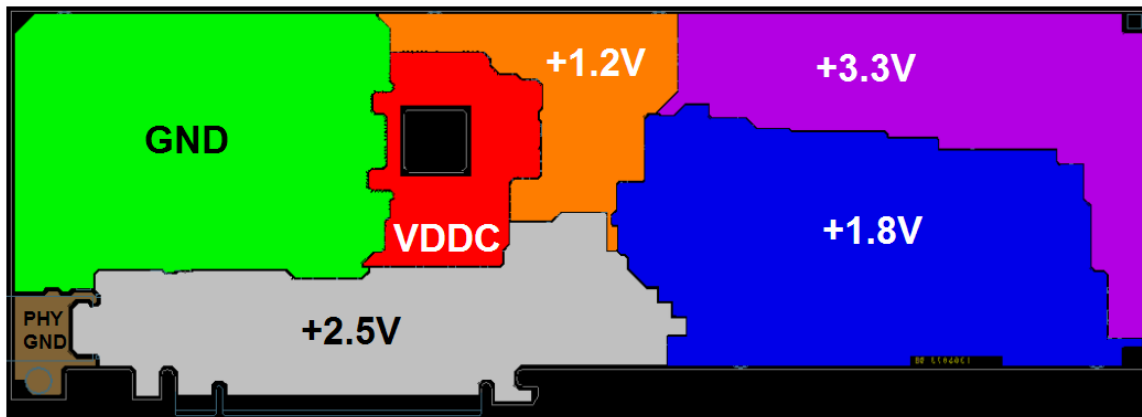


Figure 8: Power plane splits on V08

**Table 1: Power domains & ac noise**

No.	Net Name	Usage	Tolerance (%)	Tolerance (mV)	Meas. (mV)
1	VDDC	Cell BE core & RRAC digital (1.125 V)	8%	90	44
2	VDD	Bridge core & RRAC digital (1.2 V)	5%	60	64
3	+1.2V	Rambus XIO & RRAC	3%	36	56
4	+1.2VB	+1.2V filtered XDR bank 0 term	5%	60	48
5	+1.2VC	+1.2V filtered XDR bank 1 term	5%	60	
6	+1.5V	Cell BE VDDA, Bridge PCIe HSS	3%	45	64
7	+1.6V	Cell BE VDDA	3%	50	
8	+1.8V	XDR DRAM	5%	90	64
9	+1.8VA	Bridge and DDR2 DRAM	6%	100	
10	+2.5V	Cell BE, Rambus clocks, PHY	5%	125	
11	+3.3VA	+3.3V aux from PCIe edge conn	5%	165	
12	+3.3VC	Clocks, Flash, RTC, NVRAM, VREGs	5%	165	
13	+3.3VD	VREG	5%	165	
14	+3.3VE	+3.3V from PCIe edge conn	5%	165	
15	+5.0VC	H8, LEDs	5%	250	
16	+12.0VA	+12V from ATX conn	2%	250	
17	+12.0VB	VREGs	2%	250	
18	+12.0VC	Blower, VREGs	2%	250	
19	+12.0VD	+12V from ATX conn	2%	250	
20	+12.0VE	+12V from PCIe edge conn	2%	250	
21	+12.0VF	+12V from PCIe edge conn	2%	250	

## PCI Express

As mentioned in the discussion of the feasibility study, the primary trade-off for the PCIe bus involved interconnect losses and cooling. The Cell BE-XDR memory complex had to be on the left, and the blower had to be on the right. Furthermore, the number of layers required to route DDR2 and the component density on the card surface forced us to use 4 mil striplines rather than 5 mil microstrip. During the feasibility study we estimated that the PCIe interface would tolerate the extra losses and still run reliably at 2.5 Gbps. During the early part of the design phase, we performed a more accurate simulation-based analysis. When the hardware came back, we measured jitter and eye opening using the compliance base board. The analysis and testing confirmed our original estimates.

The viability of the end product rested on the question of compliance with the PCIe Card Electromechanical Specification version 1.1. Cooling, power, stack-up, placement, and performance were all interdependent on one another. We knew we would be unable to meet the conventional PCIe design guidelines, but the feasibility study did not include financing an analysis of how much margin there was in the guidelines. The situation called for a first-order estimate of the most significant contributors to loss. Since we were not funded to construct the models necessary for a simulation, we made the assumption that loss was the most fundamental parameter in the specification and that jitter would follow loss.

**Table 2: First-order PCIe loss budget @ 1.25 GHz**

<b>Component</b>	<b>RX (dB)</b>	<b>TX (dB)</b>
Card wire	1.4	1.4
Impedance discontinuity at package pins	0.6	0.6
AC coupling capacitor	N/A	1.2
Card crosstalk	0.3	0.3
Via crosstalk	0.0	0.0
<b>Total</b>	2.30	3.50
<b>Budget</b>	2.65	3.84
<b>Margin</b>	0.35	0.34

How closely can a simple sum of independent terms approximate the actual losses of a PCIe add-in card? Mathematically, a linear sum is not the correct way to combine terms in the frequency domain, especially when energy storage mechanisms are involved.

However, our approximation bounded the losses derived from a more accurate frequency-domain analysis and resulted in an add-in card with low jitter.

The card wire loss figure originated in a 2D field solver model. Then we ran a script that calculated loss from the RLG table. The size of the loss from the ac coupling capacitor and associated vias was unknown. Reasoning that someone on the PCIe committee must have done this analysis, we simply assumed it would be no larger than the difference between the TX and RX loss budgets. From previous analysis, we estimated the via crosstalk in an 0.062 in. card to be negligible.

The traditional equation for the reflection coefficient at the boundary between two ideal transmission lines predicted the loss incurred by an ASIC package with 15% impedance tolerance and a card with 10% tolerance. This calculation assumed several things. First, it assumed the signal was nearly sinusoidal and that attenuation affected its amplitude. In real life, the discontinuity would behave differently at different frequencies, thereby introducing an undulation to s21. Second, it assumed nothing about the solder ball and the card and package vias on either side of it. Third, since the PCIe specification did not indicate how much of this loss belonged to the card and how much belonged to the transmit and receive devices, we assumed an equitable distribution of half to each.

$$\text{Loss} = 20 \cdot \log \left[ 1 - \frac{Z_2 - Z_1}{Z_2 + Z_1} \right] = 20 \cdot \log \left[ 1 - \frac{55\Omega - 42.5\Omega}{55\Omega + 42.5\Omega} \right] = -1.2\text{dB}$$

Our field solver told us the crosstalk between two 4 mil differential striplines at 11 mil pair-to-pair spacing was 7 mV. Again, we made a sinusoidal assumption to arrive at this first-order approximation of loss due to crosstalk. While these calculations were not rigorous, they did capture the first principles and lead us toward a sound decision about the viability of our placement and stack-up. In general our reflection and crosstalk loss calculations erred on the high side since the energy in a signal is really spread out over band of frequencies rather than a single frequency.

$$\text{Loss} = 20 \cdot \log \left[ \frac{500\text{mV} - 14\text{mV}}{500\text{mV}} \right] = -0.25\text{dB}$$

After the production contract was signed we had the resources to sharpen our pencils. Our strategy was to select the tools in our toolbox that would allow us to obtain a level of accuracy consistent with the needs of the project. We used our own internal 2D field solver to generate lossy transmission line models, and we used an external 3D field solver to model the solder balls, vias, and ac coupling capacitors. A third tool combined the s-parameters from each of these models in preparation for frequency-domain simulation. The s-parameters appeared linear around 1.25 GHz and began to show some shallow undulation out past 5 GHz.

The first- and second-order analyses both showed that card losses clearly dominated the budget. However, the second-order analysis showed the effects of the package-card

discontinuity and the ac coupling capacitors were not nearly as strong as the first-order analysis predicted. The actual losses in the capacitor were probably higher than those predicted by our 3D model, but not by much. The size of the crosstalk did not warrant coupled simulations. The driver would absorb most of the reverse crosstalk, and the forward crosstalk was near zero due to stripline construction. Frequency-domain simulations estimated 50 ps total jitter from chip and interconnect, of which only 6 ps was due to interconnect.

**Table 3: Second-order PCIe loss and jitter budgets**

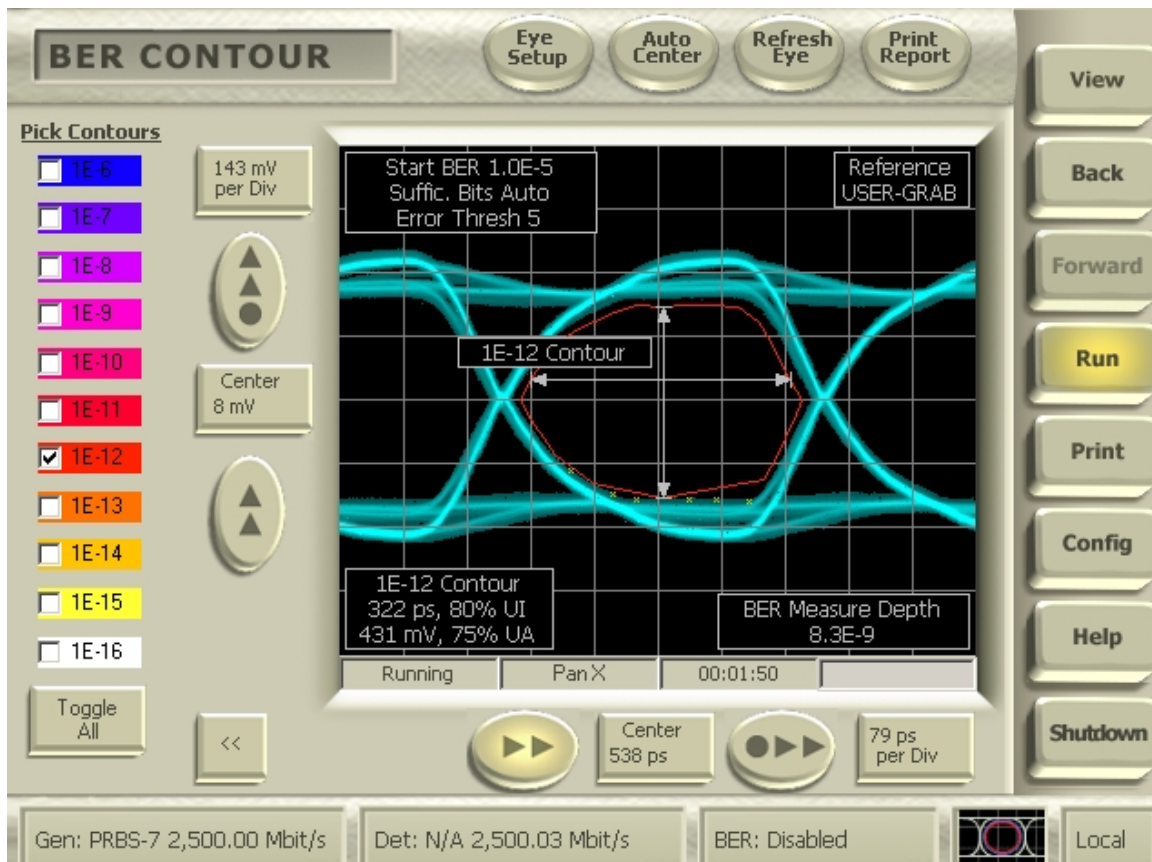
Component	TX Loss (dB)	TX Jitter (ps)
Card wire	1.4	
Impedance discontinuity at package pins	0.1	
AC coupling capacitor	0.1	
Card crosstalk	0.0	
<b>Total</b>	1.60	6
<b>Budget</b>	2.65	26
<b>Margin</b>	1.05	20

Lab measurements using the compliance base board compared our add-in card with five PCIe specifications – all passed with a healthy amount of margin. This confirmed our analysis and simulation. How healthy is healthy enough? That would be the topic of another paper.

**Table 4: PCIe Electrical compliance test results**

Specification	Range	Measured
Add-in Card Tx, Unit Interval	[399.88 ps to 400.12 ps]	399.99 ps
Add-in Card Tx, Template Tests	Zero Mask Failures	0
Add-in Card Tx, Median to Max Jitter	$\leq 56.5$ ps	19.89 ps
Add-in Card Tx, Eye-Width	$\geq 287$ ps	370.62 ps
Add-in Card Tx, Peak Differential Output voltage	[0.360 V to 1.200 V]	0.8374 V

Bit error rate (BER) measurements come in two flavors: compliance test mode and loopback mode. We measured BER using the compliance pattern. Loopback testing is significantly more complicated, and these measurements are still in progress.



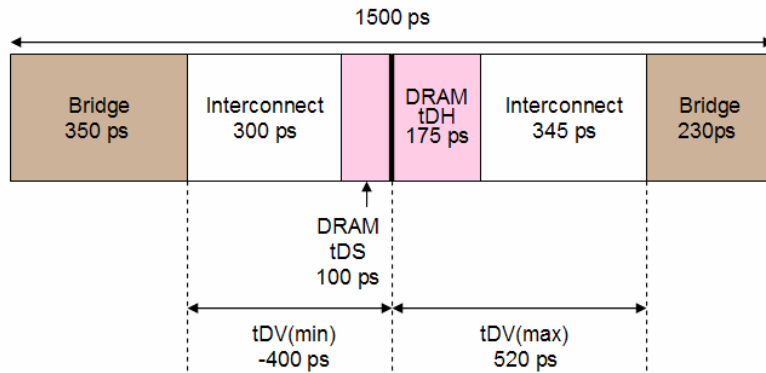
## DDR2 DRAM

If one were to consider raw data rate alone, a 667 Mbps interface should not be as challenging to design as a 5 Gbps interface. Nevertheless, the source-synchronous DDR2 interface required more signal integrity effort than all the other interfaces due to the complexity of the timing limitations imposed by the silicon. This was a chip-package-card co-design problem whose solution benefited from our ability to bring to bear expertise from across the globe: an ASIC engineering team in Haifa, static timing experts from Yasu and Rochester, DDR2 core macro designers from Austin, card physical designers from Rochester, and packaging experts from Burlington and Rochester. Careful attention to timing yielded a memory interface that came up two weeks after power-on.

Our DDR2 interface with registered address buffers had six general categories of timing constraints:

1. Strobe (DQS) to data (DQ/DQM) for write transaction
2. Strobe (DQS) to data (DQ/DQM) for read transaction
3. Skew between nine copies of strobe (DQS) for read transaction
4. Strobe (DQS) to clock (CK) for read and write transactions
5. Clock (CK) to address buffer input for read and write transactions
6. Clock (CK) to DRAM address input for read and write transactions

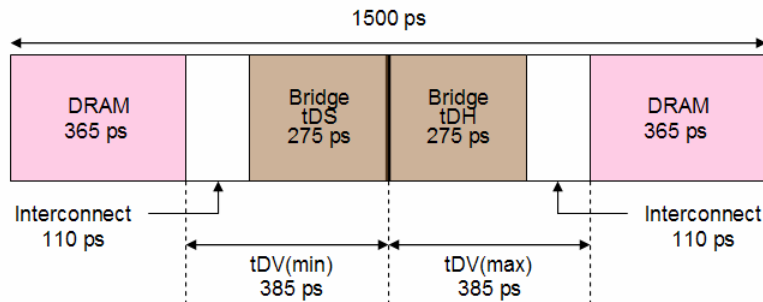
The DDR2 DRAM setup and hold specifications left a large portion of the unit interval for the bridge chip and the interconnect. IBM's DDR2 core simply phase shifted DQS 90 degrees from CK and placed it in the center of the unit interval. There was no need for fine tuning DQS-DQ timing since it was easy to achieve sufficient timing margins without it.



**Figure 9: Write timing unit interval allocation**

Read path timing was more difficult to meet than write timing for two reasons. One, the DRAM output data valid specs consumed 50% of the unit interval. Two, the bridge chip had to align nine byte lanes, each with its own strobe, and resynchronize them to the core clock. In our DDR2 core, only one delay chain controlled DQS – DQ timing for the entire nine byte lanes. This implied a maximum allowable static skew between DQS wires on the card, which became a critical factor in determining the viability of our DDR2 interface.

The factor of three difference between the read and write interconnect delays was due to the definition of switching thresholds in the DDR2 DRAM specification. We knew the bridge chip switching thresholds to be narrower, which proved to be a significant timing advantage.



**Figure 10: Read timing unit interval allocation**

The Signal Integrity team translated the DDR2 DRAM specification, merged the DRAM and interconnect timing, and generated a set of IO timing requirements for the bridge

chip. However, we did not simply throw a PDF file over the wall and hope things would turn out. Early in the project the Signal Integrity team set up a weekly meeting with members of the ASIC, package, and card development teams to insure clear communication channels. One of the earliest and most critical tasks to come out of this meeting was the assignment of C4s and package pins. At this time we also defined the package design rules, such as maximum skew between DQ bits in the same byte lane (100 mils).

At two key points in the project it became obvious that face-to-face communication was necessary. In one instance, the Signal Integrity team needed to understand the maximum DQS-DQS skew that the DDR2 core could tolerate and still capture read data in the core clock domain. Two Signal Integrity engineers traveled from Rochester, Minnesota to Austin, Texas to spend a day crawling through the resynchronization logic with the core designers. We concluded that our DRAM placement and the corresponding 3 in. skew between copies of DQS would yield sufficient operating margins at 667 Mbps. We tuned each DQ bit within 30 mils of the others in the same byte lane. On another occasion DDR2 was in the critical path of the ASIC design, and management brought engineers from Haifa, Israel and Yasu, Japan to Rochester to discuss the results of static timing analysis and compare them to the DDR2 IO timing requirements.

In order to meet the timing requirements of the DDR2 interface, we needed to pay close attention to all sources of jitter. Given the wiring density of the card and the four dual stripline layers, it was necessary to determine out worst case crosstalk and translate that into picoseconds of jitter. To calculate this, we used PCB SI 630 (Cadence Design Systems, Inc.) for the crosstalk analysis. We also compared the results from the PCB SI 630 analysis to that of an internal coupling calculator to ensure robustness. The methodology was as follows:

1. Configure a fully routed board file with appropriate IBIS models.
2. Run the crosstalk analysis within PCB SI 630 using the “All Drivers/Receivers” setting in order to identify which victim lines exhibited the highest levels of crosstalk
3. For nets which exhibited crosstalk above desired levels, run PCB SI 630 using the “Each Neighbor” setting to determine mV of crosstalk on the worst aggressors.
4. Make appropriate routing changes and repeat the process.

Using a nominal edge rate of 3 V/ns, we set our crosstalk clip level at 150 mV, i.e. we fixed every net with crosstalk greater than 150 mV. This corresponds to roughly 50 ps of jitter, which is nearly half the interconnect’s slice of the unit interval for read timing. In order for 150 mV of crosstalk to cause 50 ps of jitter, it would have to reflect off the near-end driver and reach the receiver at the same time as the input was switching. We did not analyze the timing to this level of detail, so we took at calculated risk. Lab measurements confirmed satisfactory operating margins.

$$\text{jitter} = \frac{\text{xtalk}}{\text{dV/dt}} = \frac{150\text{mV}}{3\text{V/ns}} = 50\text{ps}$$

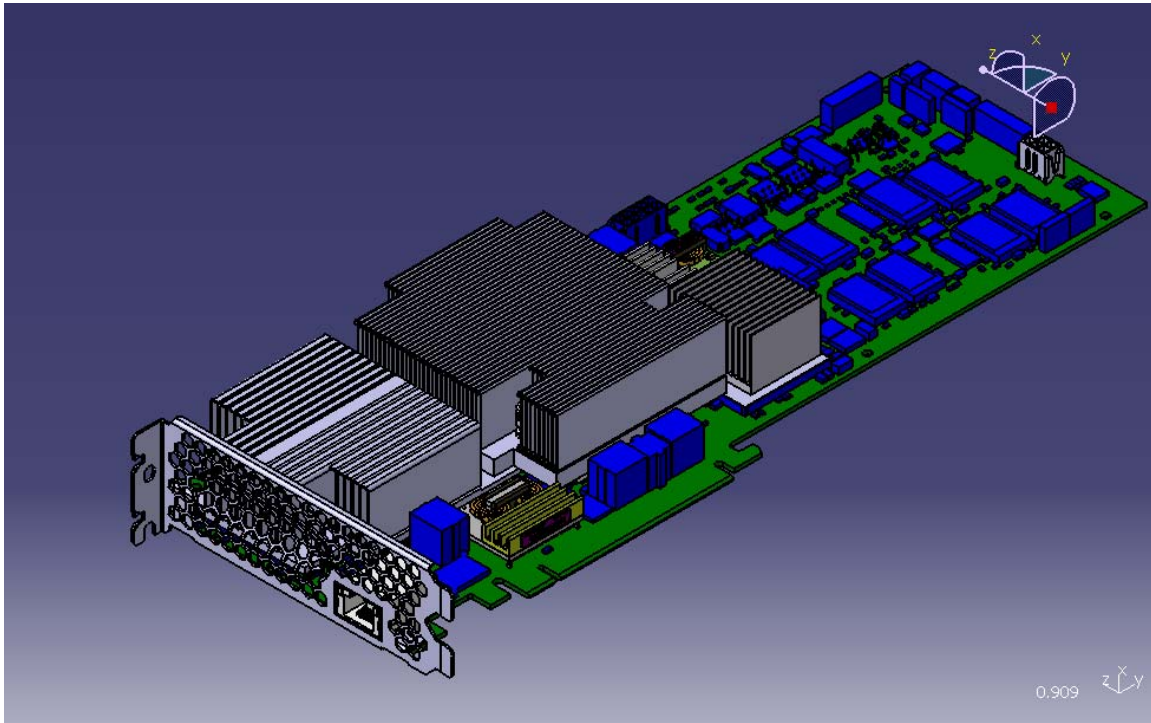
Table 5 shows the results of crosstalk simulation prior to our last iteration. HS and LS indicate the state of the victim net, high or low, and even/odd indicate the switching state of the aggressor, in phase or out of phase with the victim.

**Table 5: DDR2 crosstalk estimates**

<b>Victim Net Name</b>	<b>Max Jitter (ps)</b>	<b>HSOdd Xtalk (mV)</b>	<b>HSEven Xtalk (mV)</b>	<b>LSOdd Xtalk (mV)</b>	<b>LSEven Xtalk (mV)</b>
SDRAM0_DATA24	<b>72</b>	186.6	93.99	<b>215.4</b>	83.64
SDRAM0_DATA35	<b>55</b>	144.5	117.1	<b>165.7</b>	102.1
SDRAM1_DATA56	<b>67</b>	187.4	106.7	<b>201.8</b>	82.89
SDRAM1_DQS8_P	<b>69</b>	281.1	80.08	<b>208.3</b>	118

During our first round of testing we ran the DDR2 interface at 533 Mbps, and we saw a minimum eye opening of 66% and a maximum of 81% over six different cards. At the writing of this paper, 667 Mbps testing was still in progress.

## Conclusion



**Figure 11: Mercury Cell Accelerator Board**

Successful bring-up and verification testing of the Mercury Cell Accelerator Board were the result of collaboration between corporations, between lead engineers from various disciplines and locations, and between individual team members working on multiple concurrent projects. Ultimately, the quality of the end product benefited from the ability of the project to draw from a broad spectrum of deep resources across several companies and countries.

## References

- [1] Goto, Y., Hosomi, E., Miura, M., Harvey, P., Audet, J., Kawasaki, K., Noma, H., Mori, H., and Nishio, T., Takiguchi, I.: “Electrical Design Optimization and Characterization in Cell Broadband Engine Package,” proceedings of the 55<sup>th</sup> Electronic Components & Technology Conference, 2005.
- [2] Lee, C.: “Board Design Guidelines for PCI Express<sup>TM</sup> Architecture,” PCI-SIG, 2004.