

# Selecting the Right Memory Controller for Real-Time Applications

You can use performance modeling to compare the MPMC memory controller and CoreConnect bus for high-bandwidth DSP applications.

by Deepak Shankar  
Founder and CEO  
Mirabilis Design Inc.  
[deepak.shankar@mirabilisdesign.com](mailto:deepak.shankar@mirabilisdesign.com)

Imaging and networking applications require large data movement that must be available to the application on a specific time schedule. The embedded processor and custom co-processors in the fabric of Xilinx® Virtex™-4 and Virtex-5 FPGAs provide significant processing capacity to handle data processing algorithms. This means that data moving to and from external DRAM has a significant impact on response time.

In this article, I'll show how you can use performance analysis to evaluate the impact on task response time using different memory access techniques on the Virtex-4 FPGA. We conducted an evaluation by constructing a virtual prototype in Mirabilis Design's VisualSim, a performance modeling and architecture exploration solution.

An add-on library to VisualSim called the VisualSim Xilinx FPGA Modeling Toolkit provides components that represent common hard and soft IP available for the Virtex-4 FPGA. You can simply drag components from this library, instantiate them in a graphical block diagram editor, and modify parameters to represent the specific implementation under study. To learn more about developing virtual prototypes, see the article in Third Quarter 2006 issue of *Xcell Journal*, "Accelerating Architecture Exploration for FPGA Selection and System Design."

We used a data-dependent DSP algorithm for the evaluation. Our exploration looked at processor stalls, cache hit ratios, I/O throughput, and latency per task. These virtual prototypes provided the flexibility to experiment with different architecture configurations and select the solution that best met our requirements. Figure 1 shows an example of a virtual prototype developed in VisualSim using components from the VisualSim Xilinx FPGA Modeling Toolkit. This example shows a model containing the e405, CoreConnect bus, SDRAM, DMA, and other interface cores.

## Project Overview

There are a number of techniques for I/O processing to minimize the cycle count for data and instruction access from external SDRAM. The Xilinx multi-port multi-channel (MPMC) memory controller is an extremely efficient technique for interfacing the processor and key high-speed devices to SDRAM. Another popular method is to use a Xilinx CoreConnect bus with multiple masters, including the processor instruction/data caches and key high-speed devices connected by a slave port to the SDRAM.

To investigate the similarities and differences between the two approaches, we constructed models of both configurations using the VisualSim Xilinx FPGA Modeling Toolkit. Figure 1 shows the virtual prototype and Figure 2 is the timing diagram for request, acknowledge, read, and write access on the CoreConnect bus.

### Design Considerations

We used the same configuration to explore the Xilinx MPMC memory controller and CoreConnect bus. The instruction and data channels of the PowerPC e405, Ethernet, and PCI interface were connected as masters; the SDRAM is a slave. The Ethernet and PCI data are burst to SDRAM using dedicated CDMAC engines. The masters and slaves were maintained at constant speed and size in both virtual prototypes. Table 1 shows the instruction and data cache statistics for the virtual prototype model in Figure 1.

The design considerations were:

- What is driving the design – overall performance or a combination of power and performance?
- Do I need one or two e405 PowerPCs for my core tasks?
- If the e405 PowerPC is running at 400 MHz, what might be the best MPMC or CoreConnect bus clock ratio?
- Does my design have equal read and write memory activity, multiple reads for one write, or multiple writes for one read?
- What is the effect of SDRAM speed on the processor/memory controller/bus clock ratio?
- What is the effect of SDRAM speed on added application access to memory?

### Analysis

We ran the simulation on a 1.6-GHz Microsoft Windows XP (SP2 and Standard Edition) with 512 Mb of cache. We simulated 3.0 ms of real time and the VisualSim simulation took 28 seconds of wall clock time to finish. The model was constructed

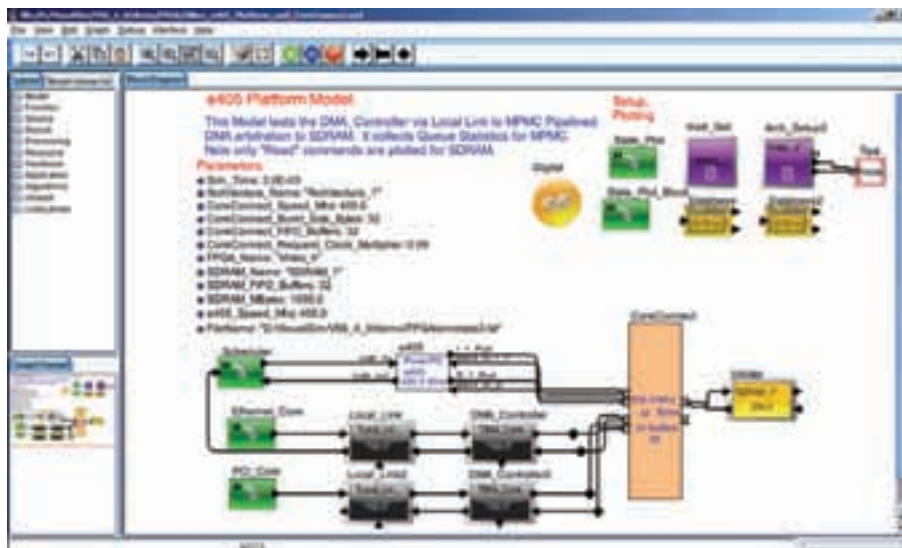


Figure 1 – VisualSim model of the Xilinx Virtex-4 platform: e405 with CoreConnect bus model

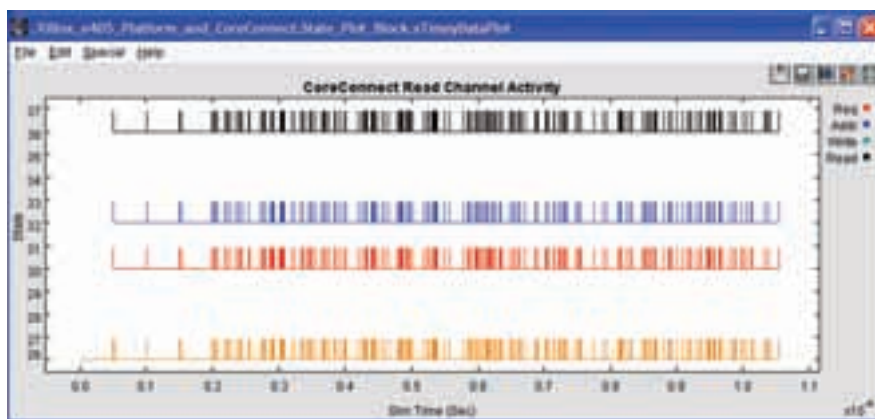


Figure 2 – CoreConnect bus activity with SDRAM at bottom

Statistic Name	CoreConnect		MPMC	
	Instruction Cache	Data Cache	Instruction Cache	Data Cache
Hit_Ratio_Max	95.05	96.23	94.98	97.37
Hit_Ratio_Mean	90.51	90.62	94.98	97.37
Throughput_MIPs_Max	74.45	1.69	7.44	0.16
Throughput_MIPs_Mean	7.44	0.17	7.44	0.16
Utilization_Pct_Max	18.61	0.42	1.86	0.04
Utilization_Pct_Mean	1.86	0.04	1.86	0.04

Table 1 – Instruction and data cache statistics

## Mirabilis Design provides a performance and architecture analysis solution for the Xilinx Virtex FPGA family.

and validated in four days using standard elements in the VisualSim library.

Using the 200-MHz MPMC, the end-to-end latency for the execution of the application benchmark was 87.190  $\mu$ s, while the 400-MHz CoreConnect was 86.42  $\mu$ s. Both matched our real-time threshold.

We found that the MPMC offered uniform latency cycles for cache to SDRAM accesses, whereas CoreConnect was more variable in cache to SDRAM access times. The lowest cycle count was achieved with the CoreConnect bus, but its average count was significantly higher. At the lower clock rate, CoreConnect performance deteriorated significantly. Except for eight out of the 33 tasks, the MPMC Memory Controller typically finished its tasks faster than the CoreConnect bus.

Table 3 shows the e405 processor stalls statistics. The CoreConnect caused a peak processor stall of 19.64%; this caused excessive latency for certain tasks. When the CoreConnect was stalled, the impact on the application latency was higher than with the MPMC. This also confirmed our finding that the CoreConnect bus had a higher overall latency and processor stalling percentage. Increasing network traffic did not cause any noticeable increase in MPMC latency, although Ethernet traffic increases did cause the overall CoreConnect latency to increase.

The MPMC memory controller had a significantly better average hit ratio (94% versus 90.51%) for the I-cache (instruction). At certain times during the simulation, the CoreConnect bus did get to a 94% hit ratio, but the duration was very short. This indicated that the MPMC arbitrated SDRAM requests better than the CoreConnect bus.

Task Name	Cycles in Processor	
	MPMC Memory Controller	CoreConnect Bus
DFT	108	109
DFT	735	742
DFT	716	712
CS_Weighting	448	523
IR	850	881
Q_Taylor_Weighting	439	523
CS_Weighting	437	523
IR	842	934
Q_Taylor_Weighting	463	492
CS_Weighting	460	549
IR	832	879
Q_Taylor_Weighting	459	515
CS_Weighting	466	523
IR	851	871
Q_Taylor_Weighting	486	502
CS_Weighting	521	513
IR	834	879
Q_Taylor_Weighting	533	482
CS_Weighting	518	518
IR	1,565	1,649
DFT	763	743
DFT	2,671	2,671
DFT	762	748

Table 2 – Task latency in cycles for key tasks in the application benchmark software

Statistic Name	CoreConnect	MPMC
Stall_Time_Pct_Max	19.64	1.99
Stall_Time_Pct_Mean	1.96	1.99
Task_Delay_Max	6.823E-06	6.750E-06
Task_Delay_Mean	2.372E-06	2.374E-06

Table 3 – Latency and stall activity statistics

Looking at the number of threads (33) and the kilobytes per thread (1.8), we could see that having a 64-KB cache for the PowerPC could reduce SDRAM latency and achieve a 100% hit ratio for the I-cache. The analysis could be extended to include the impact of multi-line prefetch and duration of loops in the application code. Table 2 shows the latency to execute the individual tasks within the application benchmark for both architectures.

### Conclusion

Virtual prototypes and performance modeling allowed us to explore a number of scenarios and clock configurations. The statistics generated provided full visibility into the operation of the FPGA design. Moreover, the fully validated VisualSim FPGA Modeling Toolkit components allowed us to save time without compromising overall accuracy. The toolkit has accurate models and extensive design statistics for Xilinx FPGAs, thus making relative comparisons between different configurations possible.

Mirabilis Design provides a performance and architecture analysis solution for the Xilinx Virtex FPGA family. Our VisualSim graphical environment contains a series of building blocks that emulate the performance characteristics of common Xilinx IP and cores. Using this library of building blocks, you can construct a specification-level simulation model of a proposed system containing multiple FPGA and on-chip resources such as memories, sensors, and buses. Model construction is a process of connecting icons that represent the IP in a graphical editor.

For details on Mirabilis and VisualSim, visit [www.mirabilisdesign.com](http://www.mirabilisdesign.com). To experience virtual prototyping, try running pre-built simulation models at [www.mirabilisdesign.com/WebPages/mdi\\_demonstration/mdi\\_demonstration.htm](http://www.mirabilisdesign.com/WebPages/mdi_demonstration/mdi_demonstration.htm). 