White Paper - Kintex 7 FPGA family: High Performance DDR3 memory throughput achieved by optimization of the memory controller

Florent Kermarrec
Sébastien Bouché

Taking full advantage of the performances of the 7 Series FPGAs can be done by using the high speed memory controller from Barco Silex, which has been reworked to achieve the highest frequency and efficiency.

This White Paper focuses on the measurement methodology put in place to validate and evaluate various scenarios, as well as the flexibility of the controller configuration to optimize timing, logic, user ports and therefore the overall performances.

Introduction

The increasing use of FPGAs as System On Chip solutions entails that the memory controller performances have never been so critical. A lot of modules independently request more and more bandwidth (processor(s), HW acceleration, data compression, audio and video buffering...), which is usually still supported by a single memory controller.
Thanks to the new 7 Series FPGAs, it is now an easier task to meet timing with the quarter rate phy (ie the fabric runs at a quarter of the frequency of the physical memory interface). However, running in quarter rate makes it even more challenging to achieve high efficiency!

This White Paper describes the new constraints introduced by the quarter-rate phy and the new methodology used to stress a memory controller with various scenarios and ensure that it always achieves the best performance compromises to serve all your applications!

Controller Performances

The memory controller of Barco Silex is an assembly of modules: SDRAM controller core, user ports and physical interface. A top-level is automatically generated with all the necessary modules, according to the user application.

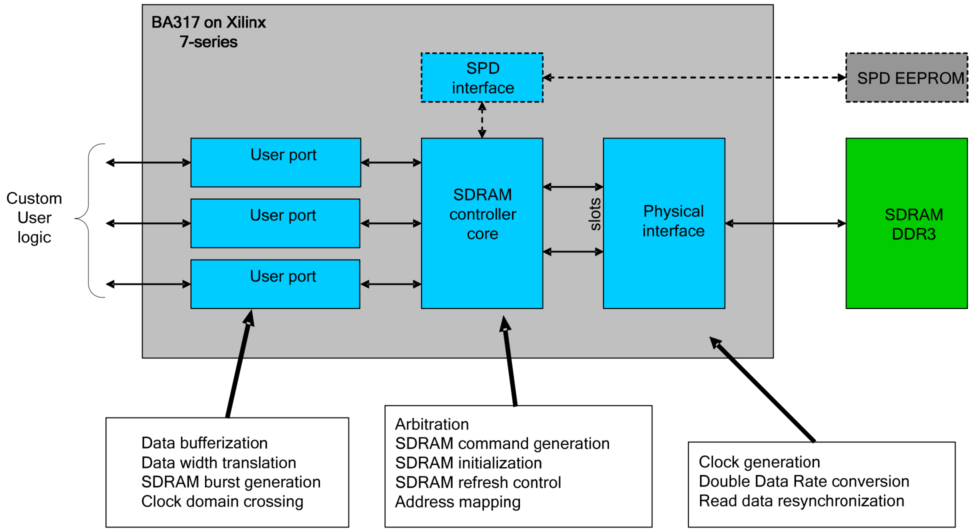

Figure 1: Simplified Barco Silex BA317 memory controller architecture

The SDRAM controller core includes a multi-port arbiter and a command sequencer. It is optimized to achieve high bandwidth, even on random accesses, by mixing the accesses to the different banks of the SDRAM, intelligently reordering bursts and other innovative strategies.

The user ports include specific FIFOs for both data and addresses. Each port can have a different data bus width (larger or smaller than the SDRAM bus width) according to the application's needs. They provide access to individual data and manage the generation of the SDRAM bursts, further hiding the rigidity that comes with DDR technology from the user and allowing her or him to concentrate the application.

While porting the BA317 to Xilinx's series 7 quarter rate phy with its data rates of up to 1.866 Gb/s the architecture has been completely revisited to include improvements based on Silex's extensive experience with DDR.

7 Series Quarter Rate design constraints:

While the previous version of the BA317 memory controller was very efficient in Full Rate and Half Rate Modes, and despite being designed in a very flexible way, early Quarter Rate Mode performance tests gave poor results.

Iteration	**Random Address WR (BL=128b)**	**Random Address RD (BL=128b)**	**Random Address WR/RD (BL=128b)**	**Random Address WR (BL=256b)**	**Random Address RD (BL=256b)**	**Random Address WR/RD (BL=256b)**	**Random Address WR (BL=512b)**	**Random Address RD (BL=512b)**	**Random Address WR/RD (BL=512b)**
0	*43.0%*	*47.0%*	*44.0%*	*44.0%*	*48.0%*	*42.0%*	*46.0%*	*48.0%*	*44.0%*
1	*44.0%*	*47.0%*	*44.0%*	*45.0%*	*47.0%*	*44.0%*	*45.0%*	*45.0%*	*44.0%*
2	*43.0%*	*47.0%*	*44.0%*	*44.0%*	*46.0%*	*44.0%*	*47.0%*	*47.0%*	*44.0%*
3	*42.0%*	*47.0%*	*42.0%*	*44.0%*	*47.0%*	*42.0%*	*46.0%*	*47.0%*	*44.0%*
4	*42.0%*	*45.0%*	*44.0%*	*44.0%*	*47.0%*	*44.0%*	*47.0%*	*48.0%*	*44.0%*
Avg	**42.8%**	**46.6%**	**43.6%**	**44.2%**	**47.0%**	**43.2%**	**46.2%**	**47.0%**	**44.0%**

Table 1: Previous Barco Silex BA317 memory controller architecture with Xilinx 7 Series Quarter Rate Phy Results

These random access results are in the range of expectation but could be improved by better predictability, using a new feature in the 7 series architecture. Therefore the controller does not draw the best advantage out of this new architecture.

In Full Rate Mode, the memory controller needs to provide 2 data per memory clock cycle to the SDRAM, 4 in Half Rate Mode, and … 8 in Quarter Rate Mode.
Memory controllers generally manage bursts of 8, meaning that in quarter rate one Burst command must be issued at each controller clock cycle. Even with a highly pipelined controller, architecture bottlenecks can appear:

- When using reordering algorithms, misplaced buffers could prevent long (efficient) reordered sequences.
- Less clock cycles are available to determine whether or not the currently opened rows should be closed.
- If bank management is done independently for each bank, special attention must be paid to the arbitration design to provide enough information to bank machines.
- …

The design of a memory controller is more and more challenging and it has become really difficult to predict how architecture changes will impact the global performances.
Those performances will strongly depend on the controller architecture, but also on the type of accesses.
While most memory controllers will achieve high efficiency on simple requests (consecutive bursts, non-conflicting requests…), their performance will be very different for complex accesses (conflicting

requests, random accesses, frequent read/write alternations...). Simple or unsuitable memory controllers can rapidly see their efficiency dropping below 10%...

Memory controller performance measurement environment

With the innovations introduced by the new architecture of the BA317 controller, it was necessary to improve the measurement tools to assess the impact of the architecture on memory transfer efficiency. As the controller's efficiency is not measured on the execution of one specific scenario, it was mandatory to design an environment that would not only build scenarios but could also easily replay them in simulation and on board.

In order to evaluate the efficiency and explore different design architectures on complex test cases, we identified 2 solutions:

- Creation of a high level model of the controller (Matlab, Python...);
- Evaluation of the performances directly on hardware.

The second solution was chosen by Barco Silex. Therefore, several features were added to bring flexibility, ease of use, and tools to investigate and reproduce test cases.

The first step was to give the possibility to the designer to create scenarios based either on user created reference files or on a reference simulation which creates these files through the user ports (a newly added feature). These files can then be the basis for the development of alternative scenarios.

To allow dynamic generation of scenarios, in simulation and on board, a simple soft core processor is used. It is intended to generate commands for each user interface, sequence their execution and then retrieve the results. Chipscope modules are mandatory for communication and monitoring of performance.

Taking account of these features, we came up with the following test environment:

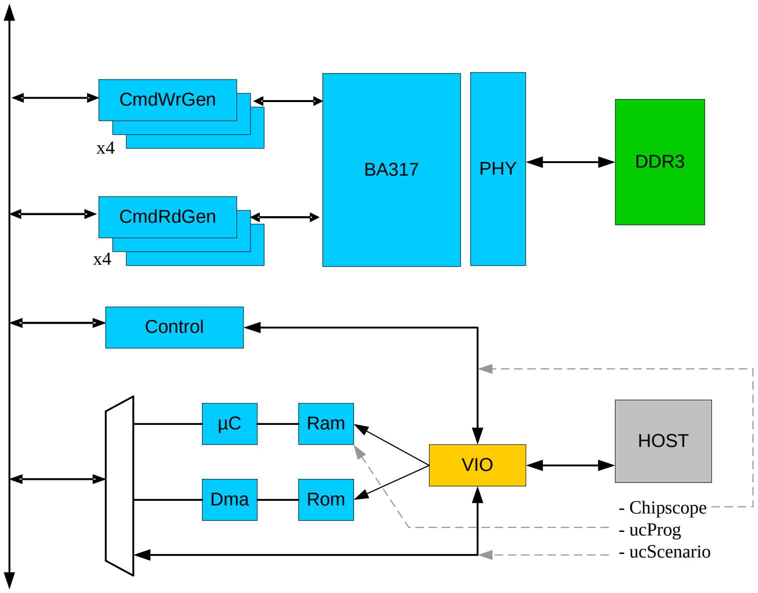

Figure 2: Performances measurement architecture

Xilinx CSE[1] is used to control Chipscope VIO[2] from a Python script, and the example script is modified to serve as VIO control over TCP/IP. The Python script can also simply connect to the TCP/IP server and control VIO from the host or a distant machine.

[1] Xilinx CSE is a set of Tcl script enabling access to JTAG, FPGA and VIO core functions.

[2] VIO (Virtual Input/Output core from Xilinx) enables the driving and monitoring of internal FPGA circuitry in real time through the JTAG connexion.

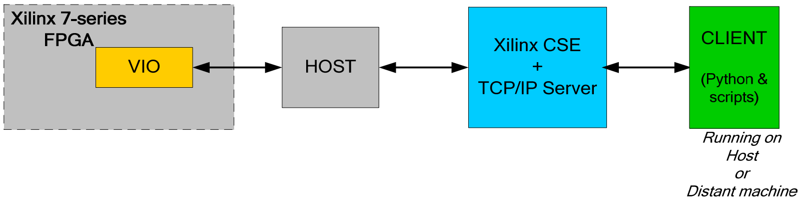

Figure 3: Python interface to VIO

Elementary modules generate read / write access to the controller ports. The access list for the test sequence is stored in a RAM inside the module, which is programmed before the start of each sequence.

This programming can be done through different interfaces:

- by direct dynamic programming from the soft core processor;
- by DMA from a ROM containing the orders of all modules;
- by the VIO / Python on the Host / Distant PC.

During the test sequence, each elementary module logs the data throughput (data accepted for a write or provided for a read by the user port) and the total sequence time. This information is retrieved at the end of the test by the processor or the VIO.

We developed various scenarios to test the efficiency:

- RANDOM ADDRESS WRITE Test
- RANDOM ADDRESS READ Test
- RANDOM ADDRESS WRITE/READ Test

Each of these patterns (easily written in Python) are run 1 million times for accuracy on a 16 bit DDR interface.

Iteration	**Random Address WR (BL=128b)**	**Random Address RD (BL=128b)**	**Random Address WR/RD (BL=128b)**	**Random Address WR (BL=256b)**	**Random Address RD (BL=256b)**	**Random Address WR/RD (BL=256b)**	**Random Address WR (BL=512b)**	**Random Address RD (BL=512b)**	**Random Address WR/RD (BL=512b)**
0	*45.0%*	*48.0%*	*42.0%*	*53.0%*	*57.0%*	*51.0%*	*62.0%*	*64.0%*	*59.0%*
1	*48.0%*	*46.0%*	*44.0%*	*55.0%*	*57.0%*	*48.0%*	*64.0%*	*63.0%*	*57.0%*
2	*45.0%*	*46.0%*	*41.0%*	*55.0%*	*58.0%*	*49.0%*	*63.0%*	*64.0%*	*57.0%*
3	*46.0%*	*48.0%*	*43.0%*	*54.0%*	*59.0%*	*50.0%*	*61.0%*	*66.0%*	*58.0%*
4	*46.0%*	*48.0%*	*42.0%*	*51.0%*	*56.0%*	*51.0%*	*61.0%*	*66.0%*	*59.0%*
Avg	**46.0%**	**47.2%**	**42.4%**	**53.6%**	**57.4%**	**49.8%**	**62.2%**	**64.6%**	**58.0%**

Table 2: The new BA317 memory controller architecture with Xilinx 7 Series Quarter Rate Phy Results

The above table shows that the random access results have significantly improved compared to those in table 1, now that the controller takes advantages of our newly implemented reordering process, grouping read/write mechanism and bank management policy.

Flexibility

Combining performance and low silicon footprint is often not an easy task in the development of an IP and even more so for a memory controller. Therefore emphasis was also placed on the configuration so that it supports multiple types of memory devices with the same set of RTL sources.
Different user ports are also available to allow easy interfacing with various industry standards (AXI, OPB / PLB IPIF, DCI for specific video projects, raw or buffered ports).

The following parameters can be configured through a configuration script managed by Barco Silex:

- The SDRAM device (DDR, DDR2, DDR3, speed grade, size…).
- The operating characteristics (CAS latency, frequency…)

- The target technology (Xilinx, ASIC…).
- The number and the type of user ports.

The following files can be generated:

- The top level of the controller. This is the file that will be used in the user design.
- A synthesizable master. This file can generate stimuli for the controller and can be optionally used to check the controller (in simulation or on silicon).
- A template for a top-level including both the controller and the synthesizable master. This file is only a starting point. It must be modified to become functional (for example, DCM or PLL might need to be instantiated).

Conclusion

The powerful combination of the Xilinx Kintex-7 devices and the expertise of Barco Silex in memory controllers delivers a state of the art solution for your next FPGA design.

Even though the BA317 allows some abstraction of the DDR constraints designers should bear in mind that the overall architecture of their design is still very important. This paper and the results of the performance studies show that the throughput can drop dramatically if the memory accesses are not correctly planned.

The tools developed by Barco Silex to measure performances can now be reused to help teams and SoC architects to increase the DDR performance by optimizing the accesses at top-level and by tuning the parameters of the controller.

To learn more about our portfolio, visit our web site at http://www.barco-silex.com/ip-cores/memorycontrollers

BarcoSilex

Barco Silex is a leader in contract engineering services, custom hardware and software development, as well as Intellectual Property (IP). Thanks to its continued stream of aggressive innovations, Barco Silex stays ahead of the competition. Barco Silex's history as a custom electronic design house (ASIC, FPGA, DSP, Board) specialized in video coding, cryptography, security and memory controllers goes back to 1991. We offer the best guarantee for continuous support throughout the complete lifecycle of products.

For more information about Barco Silex' portfolio of advanced IP cores, please contact barco-silex@barco.com or visit www.barco-silex.com.