

# External Memory Controller for Virtex II Pro

Blagomir Donchev  
Department of Microelectronics  
Technical University-Sofia  
8, Kliment Ohridski, Bl.2, 1000, Sofia, Bulgaria  
Email: donchev@ecad.tu-sofia.bg

Georgi Kuzmanov, Georgi N.Gaydadjiev  
Computer Engineering, EEMCS  
Delft University of Technology  
Mekelweg 4, 2628CD Delft, The Netherlands  
Email: {g.kuzmanov,g.n.gaydadjiev}@ewi.tudelft.nl

**Abstract**— An implementation of an On Chip Memory (OCM) based Dual Data Rate external memory controller (OCM2DDR) for Virtex II Pro is described. The proposed OCM2DDR controller comprises Data Side OCM (DSOCM) bus interface module, read and write control logic, halt read module and Xilinx DDR controller IP core. The presented design supports 16MB of external DDR memory and 32 to 64 bits data conversion for single read and write operations. Our implementation uses 1063 slices of Virtex2Pro FPGA and runs at 100 MHz. The major benefits of the proposed design are high bandwidth to external memory with reduced and more predictable access times compared to the Xilinx PLB DDR controller implementation. More specially, our read and write accesses are 2,44 and 4,25 times faster, than the PLB based solution respectively.

## I. INTRODUCTION

The PowerPC (PPC) hard cores embedded in the Virtex II Pro Field Programmable Gate Arrays (FPGA) have two bus interfaces that can be used for memory access: the Processor Local Bus (PLB) and the On-Chip Memory controller (OCM) Bus. The OCM bus supports interface to on-chip Block RAM (BRAM) only. This type of RAM has short and uniform access times, however it is limited by the size of a single chip memory only [1]. To access larger data volumes, dedicated interface to external RAMs is needed but is not currently supported. PLB is the only solution, provided by Xilinx, for connecting external memories to Virtex II Pro FPGA. Although PLB supports a variety of external memory types, such as SRAM, SDRAM, and DDR, and addresses larger storage capacities compared to OCM, it has one major drawback. This drawback is that PLB is not a dedicated memory interface but it is based on the shared bus concept. The latter concept implies that each PLB connected memory module has to compete for the bus resources with other peripheral modules attached, which potentially leads to performance degradation.

The goal of this paper is to propose a dedicated memory design solution that solves both the access time limitation of the PLB and the storage capacity limitation of the OCM. The proposed solution of the above design challenges is a memory controller hereafter referred to as OCM2DDR controller. For our design, we consider Double Data Rate (DDR) dynamic RAM due to its best performance/cost ratio compared to static memories (SRAMs) and other dynamic memory types (e.g., SDRAM). The OCM2DDR controller consists of a module for input and output 32/64 bits data conversion, a Xilinx DDR controller (v1.11) [2], an addressing module and a

control unit. The system is implemented on the Digilent XUP V2P development platform [3], which embeds a Virtex-II Pro XC2VP30 FPGA and 256 MB DDR RAM. The key features of the proposed controller are:

- Communication with external DDR memory through the Data Side OCM Controller (DSOCM);
- Run time adjustable read and write access times;
- 100 MHz operational frequency;
- Trivial resource utilization: 7.8% slices and 3.8 % flip-flops of the XC2VP30 device;
- 4,25 write speedup and 2,44 read speed up compared to Xilinx PLB DDR implementation.

The remainder of this paper is organized as follows: The motivation for this work is presented in Section II. Section III introduces the OCM2DDR controller organization and provides short discussion on its modules and on the specific clock generation strategy utilized. The implementation results of OCM2DDR controller are presented in Section IV. Finally, Section V summarizes the findings and presents the conclusions.

## II. MOTIVATION

The PowerPC cores in the Virtex2Pro are supported by two memory interfaces: the OCM and the PLB. The timing and the protocols of these interfaces are conceptually different. In this section, we briefly discuss the differences between these two interfaces. Based on their advantages and drawbacks, we motivate the need of a controller, combining some advantages of both the OCM and the PLB.

OCM provides a dedicated interface between the PowerPC core and the on chip BRAMs. Some key features of this interface are: separate Instruction Side OCM(ISOCM) and Data Side OCM(DSOCM); short and fixed access time to the BRAM memory.

PLB is based on IBM's 64-bit CoreConnect technology and uses an arbitration policy to control the slave devices attached to the bus. Some key features of this bus are: 64 bits wide data bus; 32 bits wide address bus, and 8-word cache line transfers. Xilinx provides several PLB-based external memory solutions, including a DDR SDRAM controller, which is a soft IP core with the following features [2]:

- PLB interface;
- Auto-refresh cycles generation;

- Single-beat and burst memory transactions;
- 32 and 64 bits DDR data widths;
- Error correction code (ECC).

Despite all PLB advantages, there exist two essential drawbacks: 1) Low speed and 2) The non-deterministic memory access times. A short comparison between the PLB and the OCM is presented below (for more elaborated comparison one can refer to [4]):

*Operating frequency:* The PLB operating speed depends on the maximum operating frequency of the PLB arbiter and the FPGA IP blocks that are connected to it. On the other hand, the OCM speed depends only on the amount of on-chip memory that is connected to it.

*Shared vs. Dedicated:* The PLB is a shared bus, and allows up to sixteen masters and sixteen slaves. All devices connected to the PLB have to share the available bus bandwidth. There is no arbitration on the OCM bus because of its dedicated interface.

*Non-deterministic vs. Deterministic timing:* The fact that the PLB must share its bandwidth with many masters and slaves makes its access times unpredictable. Because the OCM is a dedicated interface, it has deterministic timing.

It can be concluded that one considerable drawback of the PLB is the speed limitation imposed by the bus arbitration. Another severe PLB drawback is that the bus bandwidth is shared among all attached devices, which results in non-deterministic latencies. A positive feature of the PLB is the support for large memory sizes. In contrast to PLB, the OCM bus speed depends only on the amount of the connected BRAMs. The OCM bus is dedicated and its timing is deterministic. Serious drawback of the OCM is that the supported memory capacity is limited to the available on-chip BRAMs. Moreover, Xilinx does not provide any dedicated interface to external memories similar to the one they provide to the internal ones through OCM. This causes severe problems when fast and uniform access to external memory is required. The above observations indicate the origin of serious design problems, which arise when fast external memory accesses are required.

The above design problems motivated our research towards finding a performance efficient interface solution between the Virtex2Pro embedded PPCs and external memories with large storage capacities. More specifically, we propose a design, which combines a high speed and deterministic OCM interface from one hand and the PLB advantages to support external memory on the other.

### III. OCM2DDR CONTROLLER ORGANIZATION

The block diagram of the OCM2DDR controller is shown on Figure 1. The OCM2DDR controller consists of the following modules:

*DSOCM interface:* DSOCM is a data memory controller, which is integrated in PPC. It is connected through accepts an address and associated control signals with the processor during a load instruction, and passes valid address to the OCM2DDR controller. For store instructions, a valid addresses

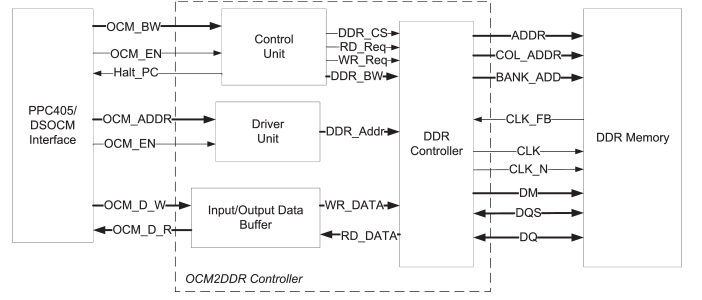


Fig. 1. Block diagram of OCM to DDR controller

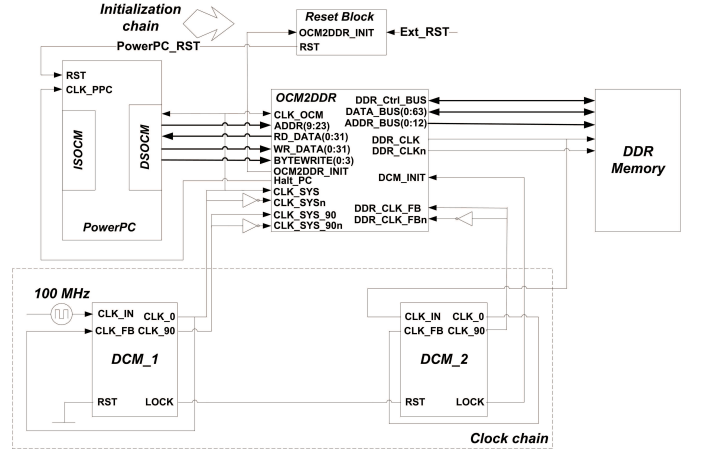


Fig. 2. Clock Architecture and Initialization chain

from the processor are accompanied by the data and by the associated control signals.

*Control unit:* Consists of logic for read/write requests generation to the DDR, chip select and read/write signals to the DDR, and halt logic driving PPC. Read and write operation are determined by OCM\_EN and OCM\_BW signals. During the read operation the PPC has to be halt for the time until DDR provides valid data.

*Driver unit:* Provides address conversion from DSOCM format to format required by the DDR controller.

*Input/Output Data Buffer:* This buffer is responsible to convert data between the 32 bits-OCM data bus format and 64 bits-DDR data bus format and is managed by the Control unit.

The main function of the OCM2DDR controller is to provide data communication between the PowerPC Core (PPC) and external DDR memory through DSOCM. In case of writing the data to the memory, PPC provides the data, the address and a write request through DSOCM to the OCM2DDR controller. The OCM2DDR controller generates all required signals with the regarded timing, for writing the data to the DDR memory. In case of memory read, PPC provides the address and a read request through the DSOCM, generates read request to the DDR.

**Design considerations:** The DSOCM's controller is implemented in a setup with a single PPC. In our design, both the data and the instruction side are used: the instruction side is

used to store the instruction segment of the program and the data side is connected to the OCM2DDR controller.

*Clock Architecture:* There are two clock schemes that are recommended by Xilinx application notes for Virtex II Pro DDR[5],[6]. In our design implementation, DCM circuits with local inversion [7] are used as illustrated in Figure 2.

The first DCM starts automatically at power on. When the first DCM is initialized, the second DCM starts. Additional DCM cores are linked together in this fashion to ensure that all clock signals are stable before the system boots up. By inserting the OCM2DDR controller into this chain, the system boot can be delayed until the DDR has been initialized.

*Signal Translation:* The OCM2DDR controller has to translate the signals provided by the OCM controller into the corresponding Intellectual Properties Interconnect Format (IPIF) signals (supported by the Xilinx DDR controller) [8] and vice versa. This leads to the signal translation diagram as shown in Figure 3. The IPIF has an address width of 32 bits, the DSOCM has only 22 bits address bus. Since the IPIF addresses are byte aligned and the DSOCM is 32-bit aligned, the two least significant bits of the IPIF address will be set to zero and the 22 bits of DSOCM address will be placed behind that. The remaining 8 bits will be constantly set to zero. More precisely, this means that every address of the DSOCM address space is mapped to a respective address of the DDR controller. The IPIF protocol uses a scheme called "Byte Steering". This means that the peripheral can address the memory space byte aligned, but the data must be provided, in compliance with the base bit alignment of the bus. This means that the address is given as a byte address, but the byte mask and data are aligned to the width of the data bus (64 bits). The address generated from the DSOCM is always aligned to 32 bits. This conversion holds for both the incoming, and outgoing data, and the data mask has to be shifted accordingly. Both masks of the IPIF and of the DSOCM hide the data on a byte level. The byte mask of the IPIF specifies the bytes that contain valid data. The DSOCM mask determines the bytes to be written to the BRAM. For write operations, this means that the byte mask can be simply copied. However, for read operations, the DSOCM byte mask is kept empty, while all the data bits on the bus are expected to be valid. The IPIF bus has separate read/write indicator signals, and the byte mask validates the data for both, read and write operations. This means that in the case of a read operation, the DSOCM byte mask is empty, but the translated IPIF byte mask should be completely asserted.

Because of read/write timing differences between the DSOCM and DDR, it is necessary to halt the processor during the read operation for the time, required for DDR memory to provide valid data. A special logic circuit is developed to implement this feature.

#### IV. VIRTEX II PRO MAPPING

The proposed design has been implemented using Xilinx Platform Studio 7.1i [9]. Initially, the design has been simulated with ModelSim 6.0 SE using a reference functional

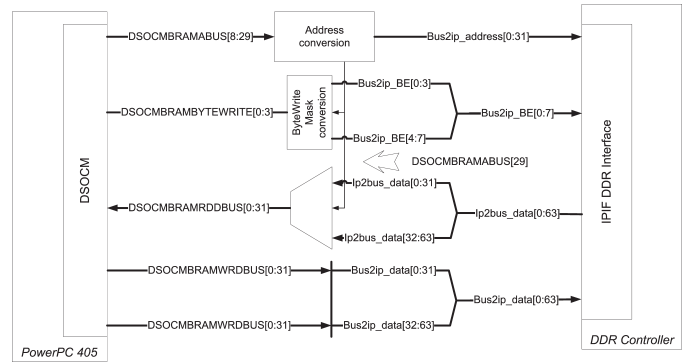


Fig. 3. Signal translation conception

timing model of Micron DDR 256 MB memory, provided by the vendor.

Implementation results of OCM2DDR controller, presented in Table I suggest that the hardware costs are trivial with respect to the available reconfigurable resources (8%). The reported delays suggest a maximum speed of 159,9 MHz. After implementing in XC2VP30-7 (Digilent's XUP V2Pro board)[3], the design was tested at 100 MHz with two syntectic applications that write and read into the DSOCM address space. One of them consists of single word (32 bits) write and read operations and the second one consists of loops of memory initializations and linear write/read operations for 20 32 bits words. Figure 4 and Figure 5 depict the simulation results of the OCM2DDR with the DSOCM in a single cycle mode. Position 1 on both figures clearly indicates that the DDR access is completed within the OCM bus assertion. The DDR memory used and its simulation timing model have a CAS latency of two clock cycles. Because of the necessity to keep DDR\_CS signal for longer time than the DSOCM Enable signal, an internal counter was used, indicated by position 2 on Figure 5. Because of the difference between the times required for read/write operations by the PPC and the DDR, it is necessary to halt PPC during the read operation. The halt lasts for the time required by the DDR memory to provide the data, depicted by position 3 on Figure 4. This feature is implemented using simple logic based on a clock multiplexor primitive (BUFGMUX) [10]. The proposed solution follows recommended technics for clock synchronization given by Xilinx [11]. A severe concern is the fact that the DDR access time can vary greatly. To solve this problem, a run time adjustable circuit for read and write operations was developed. The execution time for both operation is calculate with generation of acknowledge by the internal DDR controller. Its behavior is indicated by position 4 on Figure 4

For debugging purpose, an Input/Output interface based on the Xilinx OPB UART Light IP core [12] is designed. Its parameters are the following: 115200 kbits/s, 8 bits data, no parity check and no hardware/software corrections. In this implementation the CPU and the OCM2DDR controller are running at 100 Mhz with additional fixed phase shifting of 60 degrees in the second DCM. It is done to compensated

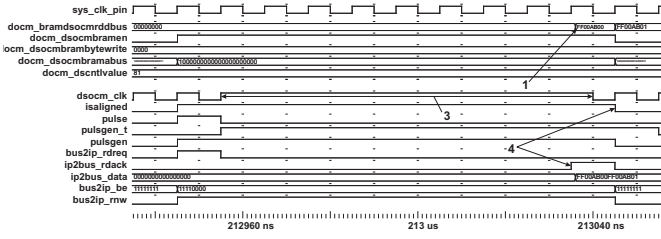


Fig. 4. Read data from DDR

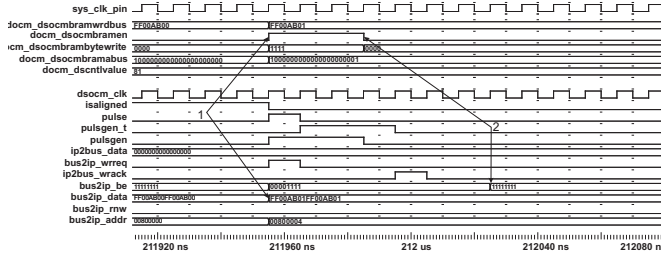


Fig. 5. Write data to DDR

the external wire's delay of the clock path. More details about technics on how to calculate the proper phase shifting are given in [13].

Table I presents the synthesis results for the proposed memory controller and provides comparison to the Xilinx PLB DDR controller. Synthesis results indicate substantial savings of design resources in the range of 17%-30%. The reason of that is lower complexity of OCM interface vs. PLB. The last row of Table I suggest that our design exhibits 30% shorter critical path, therefore it can be run at approximately 1.6 times higher frequency, then the PLB.

Regarding performance, experimental results suggest that our OCM DDR controller takes 4 clock cycles for the single write operation and 9 clock cycles for single read operation. In comparison, the PLB DDR controller takes 17 clock cycles for a write operation and 22 clock cycles for a read operation. Compared timing results between both implementations are reported in Table II. Note that we consider the worst case scenario, when no bus arbitration takes place and only one PLB DDR controller is attached to the PLB bus. If bus arbitration is considered, the PLB latencies are expected to increase dramatically.

TABLE I  
IMPLEMENTATION RESULTS

Used resources	OCM2DDR	PLB DDR	Differences
Number of Slices	1063	1246	17 % less
Number of Slice Flip Flops	1052	1367	30 % less
Number of 4 input LUTs	803	971	21 % less
Minimum clock period	6.254 ns	9.968 ns	60 % faster

TABLE II  
TIMING RESULTS AFTER SYNTHESIS

Timing parameters	OCM2DDR	PLB DDR	Speed up
Duration of write operation	4 Cycles	17 Cycles	4.25
Duration of read operation	9 Cycles	22 Cycles	2.44

## V. CONCLUSION AND FUTURE WORK

In this paper we proposed a design of a controller, which provides a dedicated interface to external DDR memory connected to the PowerPC cores of the Xilinx Virtex II Pro FPGAs. More specifically, we proposed a high speed access to large external storage capacity trough the dedicated DSOCM bus of Virtex2Pro. Compared to the traditional shared-bus approach (provided by the chip vendor) for connecting external memories our dedicated controller performs in the worst case 2.44 times faster for read and 4.25 times faster for write operations. Synthesis results suggest trivial hardware cost, measured with 8 % of XC2VP30. The proposed solution can be extended in future with a cache module implementation, running as L2 caching subsystem of reconfigurable processors such as MOLEN [14], [15]. The performance can be improved further by implementing a burst access to the external memory and ECC functionality also. The OCM2DDR controller can be also considered as a universal solution to connect IPIF compatible external memories ( static and dynamic).

## ACKNOWLEDGMENT

This research has been partially supported by the National Science Fund, Bulgarian Ministry of Education and Science, Project MU-X-02/29.07.2005.

## REFERENCES

- [1] "Virtex II Pro and virtex II Pro X platform FPGAs: Introduction and overview," Xilinx Corporation, DS083, Oct. 2005.
- [2] "PLB Double Data Rate (DDR) Synchronous DRAM (SDRAM) Controller," in *Product Specification*, Xilinx Corporation, DS425, Aug. 2004.
- [3] "Xilinx University Program Virtex-II Pro Development System," Hardware Reference Manual, UG069, Mar. 2005.
- [4] K. Lund, "PLB vs. OCM Comparison Using the Packet Processor Software," Xilinx Corporation, XAPP644, Oct. 2004.
- [5] H. Winkler, "Clocking Strategy for a Virtex II Pro DDR SDRAM Controller," in *Array Electronics*, <http://www.array-electronics.de/doc>.
- [6] C. Cain, "Reference system: Mch opb ddr sdram with opb central dma," Xilinx Corporation, XAPP912, Nov. 2005.
- [7] "High-Speed Clock Architecture for DDR Designs Using Local Inversion," Xilinx Corporation, XAPP685, Apr. 2004.
- [8] "PLB IPIF (v2.02a)," Xilinx Corporation, DS448, Apr. 2005.
- [9] "Embedded system tools reference manual," Xilinx Corporation, UG111, Feb. 2005.
- [10] "Libraries guide," Xilinx Corporation, ISE 6.3, Sept. 2005.
- [11] "Powerpc 405 processor block reference guide," Xilinx Corporation, UG018, July 2005.
- [12] "Opb uart lite (v1.00b)," Xilinx Corporation, DS422, May 2005.
- [13] "Determining the Optimal DCM Phase Shift for the DDR Feedback Clock," Xilinx Corporation, XAPP806, May 2005.
- [14] S. Vassiliadis, S. Wong, G. N. Gaydadjiev, K. Bertels, G. Kuzmanov, and E. M. Panainte, "The Molen Polymorphic Processor," *IEEE Transactions on Computers*, vol. 53, no. 11, pp. 1363–1375, Nov 2004.
- [15] S. Vassiliadis, S. Wong, and S. D. Cotofana, "The molen  $\mu$ -coded processor," in *11th International Conference on Field-Programmable Logic and Applications (FPL)*, Springer-Verlag Lecture Notes in Computer Science (LNCS) Vol. 2147, August 2001, pp. 275–285.