Implementation of High Performance QDR II SRAM Interface with Xilinx FPGA

Ram Krishna Dewangan[1]
Asst. Professor
Department of Electronics and Communication Engg.
Institute of Technology, GGV, Bilaspur,India
ramkrishna.21@gmail.com

Prahlad Kumar Khandekar[2]
M. Tech. Scholar
Department of ET&T
Chouksey Engg. College, Bilaspur, India
prahladkh@gmail.com

***Abstract* - Memory devices are critical components of electronic systems. And with increasing complexity brought about by greater end market demands, next-generation systems require newer memory architectures. For networking infrastructure applications, the memory devices required are typically high-density, high performance, high bandwidth memory devices with a high degree of reliability. For very high speed data communication system Designer need faster processors, faster memory and high speed interfacing peripheral components. While the processors in these systems have improved in performance, static memories have been unable to keep up the pace. Newer SRAM architectures have evolved to support the higher throughput requirements of current systems and processors. This application note introduces QDR, which is an SRAM architecture designed to improve the SRAM interface bandwidth by more than four times that of the current solutions. This paper summarizes that for high performance as well as high bandwidth requirement of the networking application, QDR II SRAM has very efficient memory architecture and QDR II SRAM is used for interface with the Xilinx FPGA.**

***Keywords* – QDR II SRAM, QDR II Controller, User Interface, Physical interface, UART, Burst length, Bus Width.**

I. INTRODUCTION

In order to increase memory bandwidth significantly for future high-performance communication applications, Cypress Semiconductor, Integrated Device Technology, Inc. and Micron Technology have jointly defined and developed a new SRAM architecture referred to as the Quad Data Rate™ (QDR™) SRAM technology. This architecture is aimed at the next generation of switches and routers that operate at data rates above 200 MHz, and will serve as the main memory for lookup tables, linked lists, and controller buffer memory.

The basic QDR architecture has independent read and write datapaths for simultaneous operation. Both paths use Double Data Rate (DDR) transmission to deliver two words per clock cycle, one word on the rising clock edge and another on the falling edge.

QDR-II architecture has separate data inputs and data outputs to completely eliminate the need to "turn-around" the data bus required with common I/O devices. Access to each port is accomplished through a common address bus. Addresses for Read and Write addresses are latched on alternate rising edges of the input (K) clock. Accesses to the QDR-II Read and Write ports are completely independent of one another. In order to maximize data throughput, both Read and Write ports are equipped with Double Data Rate (DDR) interfaces.

Here for high speed and high performance, we have used QDR-II SRAM CY7C1315BV18 – 512K x 36 operating at 1.8V equipped with both a Read port and a Write port. The design is implemented in Virtex-5 FPGA Board.

II. BLOCK DIAGRAM

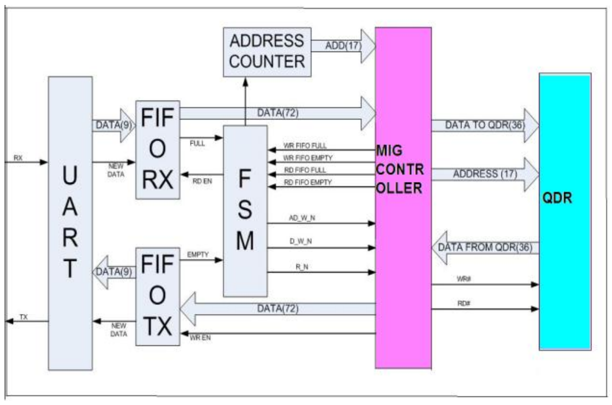

Fig. 1 Block Diagram of QDRII SRAM Interface

Figure 1 shows high-level block diagram of the QDR II reference design that shows both the external connections to the QDR II memory device and the internal FPGA fabric interface for initiating Read/Write commands.

This design uses MIG controller as main memory controller QDR II. Interface consist of User Interface, Physical Interface , Read/Write State Machine, Delay Calibration State Machine.

III. DESIGN OVERVIEW

The user interface uses a simple protocol based entirely on SDR signals to make Read/Write requests. This module is constructed primarily from FIFO16 primitives and is used to store the address and data values for Read/Write operations before and after execution. The Read/Write state machine is responsible for monitoring the status of the FIFOs within the user interface module, coordinating the flow of data between the user interface and physical interface, and initiating the actual Read/Write commands to the external memory device. It ensures execution of Read/Write operations with minimal latency in a concurrent manner as per the requirements of the QDR II memory specification.

The physical interface is responsible for generating the proper timing relationships and DDR signalling to communicate with the external memory device in a manner that conforms to its command protocol and timing requirements.

The delay calibration state machine is an integral component of the direct-clocking methodology used to achieve maximum performance while greatly simplifying the task of read data capture inside the FPGA.

IV.UART DESIGN

A UART is usually an individual (or part of an) integrated circuit used for serial communications over a computer or peripheral device serial port.

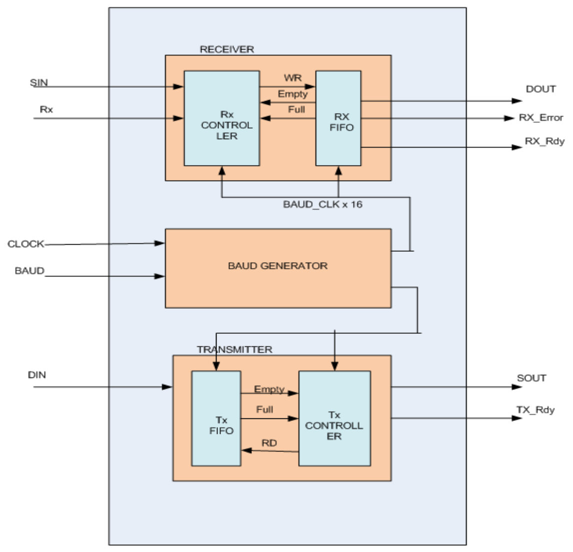

Fig. 2 Block diagram of UART

UARTs are now commonly included in microcontrollers. A UART is used to convert the transmitted information between its sequential and parallel form at each end of the link. Each UART contains a shift register which is the fundamental method of conversion between serial and parallel forms.

Figure 2 Block diagram of UART. All operations of the UART hardware are controlled by a clock signal which runs at a multiple (say, 16) of the data rate - each data bit is as long as 16 clock pulses. The receiver tests the state of the incoming signal on each clock pulse, looking for the beginning of the start bit. If the apparent start bit lasts at least one-half of the bit time, it is valid and signals the start of a new character. If not, the spurious pulse is ignored. After waiting a further bit time, the state of the line is again sampled and the resulting level clocked into a shift register. After the required number of bit periods for the character length (5 to 8 bits, typically) have elapsed, the contents of the shift register is made available (in parallel fashion) to the receiving system.

V. IMPLEMENTATION OF QDRII SRAM CONTROLLER

Figure 3 shows the timing protocol required to issue Read/Write requests to the user interface when using the 4-word burst reference design. As mentioned previously, the interface uses all SDR signals synchronized to the main FPGA design system clock (USER_CLK0).

A. User interface QDRII Write Cycle Timing Diagram

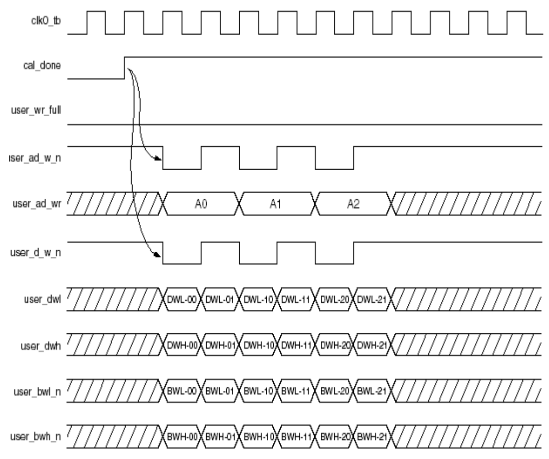

Figure 3: User write interface, ref [4]

Figure 3 shows User Read interface timing diagram. To write User address user asserts user_ad_w_n signal low. Write data bits are written into Write Data FIFO when user asserts user_d_w_n signal low. The first write data is written in Write DATA FIFO on the same clock when Write Data Address is given. Write burst data should be done on consecutive clock cycle as there can not be any break between the data burst.

B. User Interface QDRII Read Cycle Timing Diagram

Figure 4 shows User Read interface timing diagram. User Read Address bits are written into User Read Address FIFO when user asserts user_r_n signal low. Controller then reads the memory address from

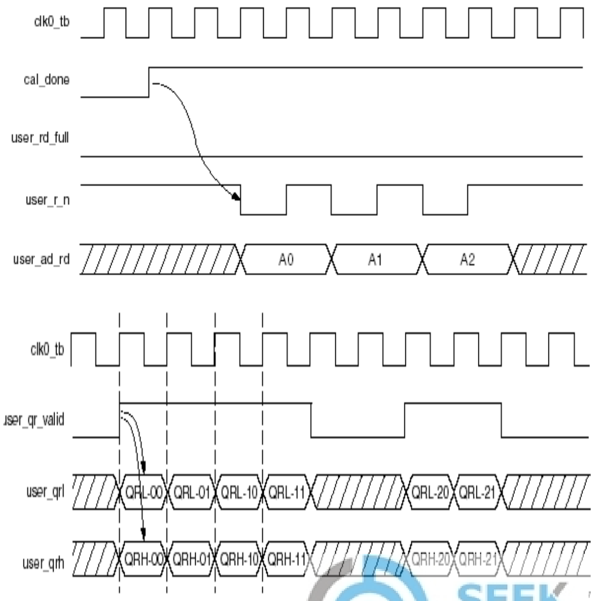

Figure 4: User Read interface, ref [4]

FIFO and places it onto memory address bus. When User Read Address FIFO is full then user_rd_full signal is asserted high by the controller. This indicates user to stop writing into User Read Address FIFO.

C. *FSM for user write interface*

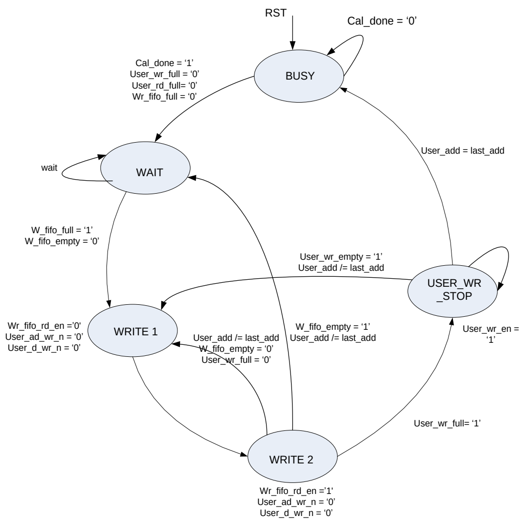

Figure 5: FSM for user write interface

Figure 5 shows FSM for user write interface. The write cycle timing diagram shows when the calibration is complete memory write address is written to Write Address FIFO.

Write requests are made via an active-Low USER_W_n signal during the rising edge of USER_CLK0. The Write address (USER_AD_WR) must be presented on this same clock edge. The first and second 36-bit data words to be written to the memory are also presented at
this time to the 36-bit USER_DWL and USER_DWH input buses, respectively. The third and fourth words of the four-word burst are presented to USER_DWL and USER_DWH, respectively, on the next rising edge of USER_CLK0.

D. *FSM for user read interface*

Figure 5 shows FSM for user write interface. Read requests are made via an active-Low USER_R_n signal during the rising edge of USER_CLK0. The 18-bit Read address (USER_AD_RD) must be presented on this same clock edge. After the execution of the Read command, the 4-word burst values are stored in the Read data FIFOs.

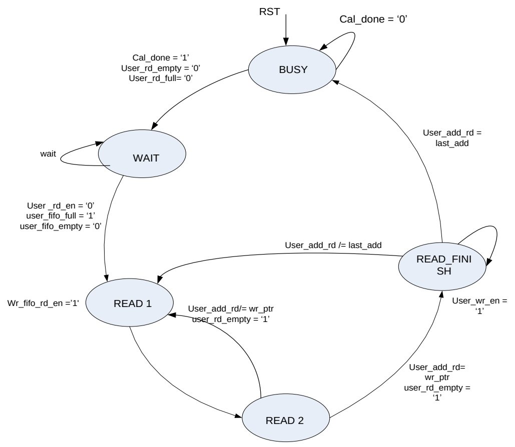

Figure 6: FSM for user read interface

VI. SIMULATION RESULTS

A. *QDRII User Write Simulation result*

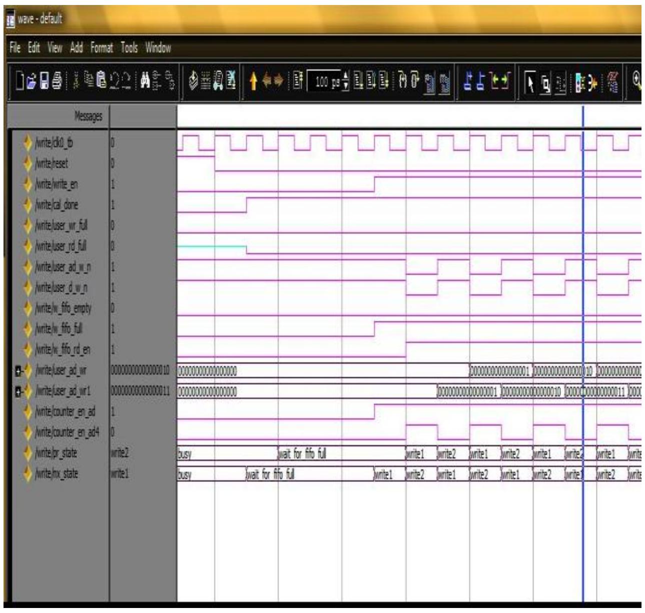

Figure 7: QDRII User Write Simulation result

Figure 7 shows simulation results for QDRII User Write Interface. When reset signal is high then FSM stays in Busy State. When Cal_done signal is high then the User Address is written in address command FIFO when user asserts User_ad_w_n signal low and User Data is written into Write Data FIFO when User asserts User_wr_d_n signal low.

B. *QDRII User Read Simulation result*

Figure 8 shows simulation results for QDRII User Read Interface. When reset signal is high then FSM stays in Busy State. When Cal_done signal is high then the User Read Address is written in User Read Address FIFO when user asserts User_r_n signal low. Controller reads this memory address from FIFO and places onto memory address bus. When User_qr_valid signal becomes high then read data is available to user .

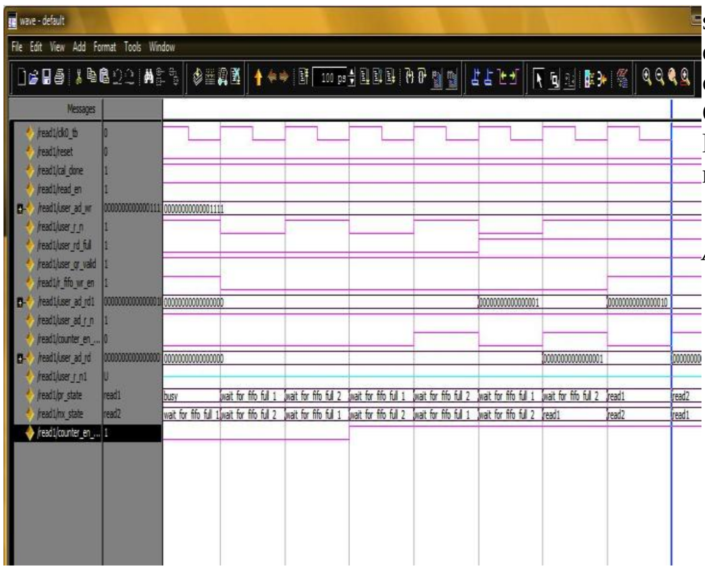

Figure 8: QDRII User Read Simulation result

C. *QDRII simultaneous Read/Write Simulation result*

Figure 9 shows simulation results for QDRII read/write operation for 4-word Burst . when User_ad_w_en signal is low then USER Write Address is written into User Address FIFO and 4 word burst data is written into Write Data FIFO using User_dwl and User_dwh signals.

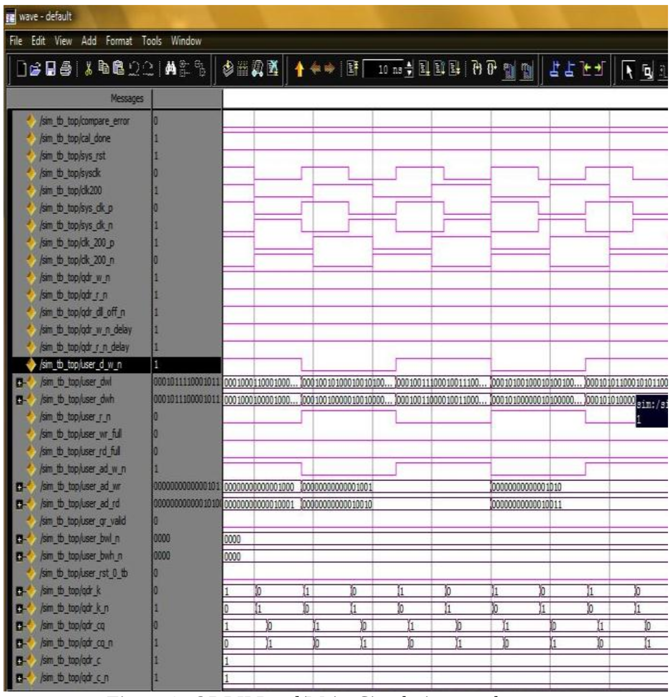
wave - default
File Edit View Add Format Tools Window
Messages
/sim_tb_top/compare_error 0
/sim_tb_top/cal_done 1
/sim_tb_top/sys_rst 1
/sim_tb_top/sysclk 0
/sim_tb_top/clk200 1
/sim_tb_top/sys_clk_p 0
/sim_tb_top/sys_clk_n 1
/sim_tb_top/clk_200_p 1
/sim_tb_top/clk_200_n 0
/sim_tb_top/qdr_w_n 1
/sim_tb_top/qdr_r_n 1
/sim_tb_top/qdr_dll_off_n 1
/sim_tb_top/qdr_w_n_delay 1
/sim_tb_top/qdr_r_n_delay 1
/sim_tb_top/user_d_w_n 1
/sim_tb_top/user_dwl 0001011110001011
/sim_tb_top/user_dwh 0001011100001011
/sim_tb_top/user_r_n 0
/sim_tb_top/user_wr_full 0
/sim_tb_top/user_rd_full 0
/sim_tb_top/user_ad_w_n 1
/sim_tb_top/user_ad_wr 000000000000101
/sim_tb_top/user_ad_rd 00000000000010100
/sim_tb_top/user_qr_valid 0
/sim_tb_top/user_bwl_n 0000
/sim_tb_top/user_bwh_n 0000
/sim_tb_top/user_rst_0_tb 0
/sim_tb_top/qdr_k 0
/sim_tb_top/qdr_k_n 1
/sim_tb_top/qdr_cq 0
/sim_tb_top/qdr_cq_n 1
/sim_tb_top/qdr_c 1
/sim_tb_top/qdr_c_n 1

Figure 9: QDRII Read/Write Simulation result

VII. Conclusion

QDR SRAMs are designed to greatly increase memory bandwidth compared to existing SRAM solutions in applications such as switches and routers. These SRAMs will serve as the main memory for look-up tables, linked lists and controller buffer memory to enhance bandwidth in future data communication systems. A family of high-performance QDR SRAMs is defined to ensure customers have the security of consistent multiple supplier roadmaps. In DDR and QDR memories the double-frequency clock is phase locked to the system clock. All QDR signals are registered in the I/O buffers and use HSTL buffers.

A. *Comparison of performance of QDRII vs. DDR2 SDRAM from Synthesis report*

Features	QDRII SRAM	DDR2 SDRAM
FPGA logic resource used 1. Slice Reg. 2. Slice LUTs 3.Flip Flop 4. IOs 5.Shift Register 6.Clock Buffer 7.IO Buffer	 1389 944 1409 120 2 4 118	 3192 2669 3295 205 23 4 193
Max Operating frequency	230.62 MHz	197.82 MHz
Minimum input arrival time before clock	1.663 ns	0.341 ns
Maximum output required time after clock	5.241 ns	3.877 ns
Maximum combinational path delay	0.818 ns	0.818 ns
Operation	2, 4 Word Burst	4, 8 word Burst
Bus width	36 Bit Write/ 36 Bit Read	36 Bit Write/ 36 Bit Read
IO Standard	HSTL (1.8 v)	SSTL (2.5 v)
Target Memory Dev.	CY7C1315B V18	MT47H32M 16XX-3

Table 1: Comparison from Synthesis report

References

[1] QDR SRAM Consortium – **www.qdrsram.com**

[2] Implementing High Performance Memory Interface by Adrian Cosoroaba, **Embedded Computing Design** Annual Product Directory / September 2005.

[3] QDR II SRAM Interface for Virtex-4 Devices, XAPP 703, July 9, 2008 **www.xilinx.com**.

[4] Xilinx Memory Interface Generator User Guide, UG086, Sept 18, 2007, **www.xilinx.com**.

[5] QDRII SRAM Clocking Scheme, Cypress Semiconductor Corporation, Feb 16, 2005

[6] Get Control of High Speed Design, By Jerry A. Long, Xcell Journal, ***www.timingdesigner.com***.

[7] QDR II SRAM Local Clocking Interface for Virtex-II Pro Devices, XAPP 250 (V1.0), May 24, 2004, **www.xilinx.com**.

[8] Cypress Semiconductor Corporation, 9-Mb Pipelined SRAM with QDR Architecture CY7C1302V25, March 28, 2000, **http://www.cypress.com**

[9] Cypress Semiconductor Corporation at **http://www.cypress.com**

[10] HITACHI, Ltd. at **http://semiconductor.hitachi.com/memory.html**

[11] Integrated Device Technology, Inc. at **http://www.idt.com**