

AXI Compliant DDR3 Controller

Vikky Lakhmani, M.Tech(Sequential) Student
 Department of Electrical & Electronics Engineering,
 Uttar Pradesh Technical University,
 Lucknow (Uttarpradesh), INDIA
 Email: vikky lakhmani@gmail.com

Nusrat Ali
 Project Manager (VLSI Division),
 HCL Technologies, Noida, INDIA
 Email: nusrata@hcl.in

Dr. Vijay Shankar Tripathi
 Department of Electronics & Communication
 Motilal Nehru National Institute of Technology,
 (MNNIT), Allahabad (U.P), INDIA
 Email: vijay_s_t@yahoo.com

Abstract—This paper describes the implementation of AXI compliant DDR3 memory controller. It discusses the overall architecture of the DDR3 controller along with the detailed design and operation of its individual sub blocks, the pipelining implemented in the design to increase the design throughput. It also discusses the advantage of DDR3 memories over DDR2 memories and the AXI protocol operation

Keywords- DDR3 memory, AXI Interface, AXI access Manager, DDR2 memories, AXI protocol operation

I. INTRODUCTION

The AXI compliant DDR3 Controller permits access of DDR3 memory through AXI Bus interface[1-4]. The DDR3 controller works as an important bridge between the AXI host and DDR3 memory. It takes care of the DDR3 initialization and various timing requirements of the DDR3 memory. The DDR3 controller performs multiple schemes to increase the effective memory throughput[3]. These schemes include combining and reordering the Read/Write commands. For attaining the maximum throughput from the memory, it operates all the memory banks in parallel and minimizes the effect of precharge/refresh and other DDR3 internal operations [2].

AXI Compliant DDR3 Controller Architecture

The architecture of the design is shown in the fig.1. The design consists of following blocks-

- AXI interface
- AXI access Manager
- DDR3 Controller

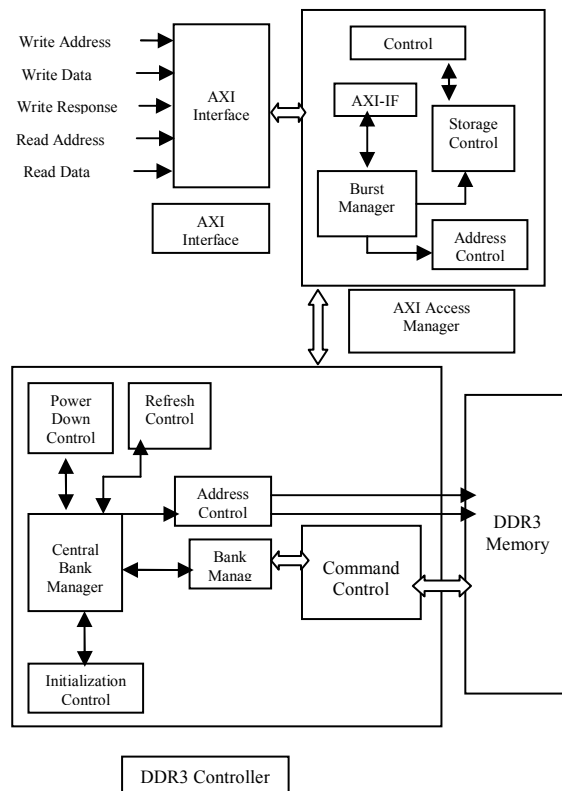


Fig.1 AXI compliant DDR3 controller

A. AXI Interface:

AXI-Interface block interacts with HOST processor and AXI access manager. It is responsible for accepting and interpreting the AXI commands issued by the processor and responding to Read / Write requests in AXI protocol as requested by processor. It also maintains an arbiter block which is responsible for arbitrating between Read/Write commands. Arbitration is required between the commands

because of parallel/independent Read/Write command received at the AXI interface.

It maintains asynchronous FIFO's to store the command and the data. The Read command gets stored in (Read Command Block), Write command gets stored in (Write Command block), Read data gets stored in (Read Data block), and Write data gets stored in (Write Data Block). The stored commands are supplied to the AXI access manager whenever AXI access manager is free. Now since the storage can have both Read and Write commands in the respective block hence it maintains an arbiter which arbitrates between Read/Write command and whenever Burst Manager is free one of the present command is supplied to the Burst Manager.

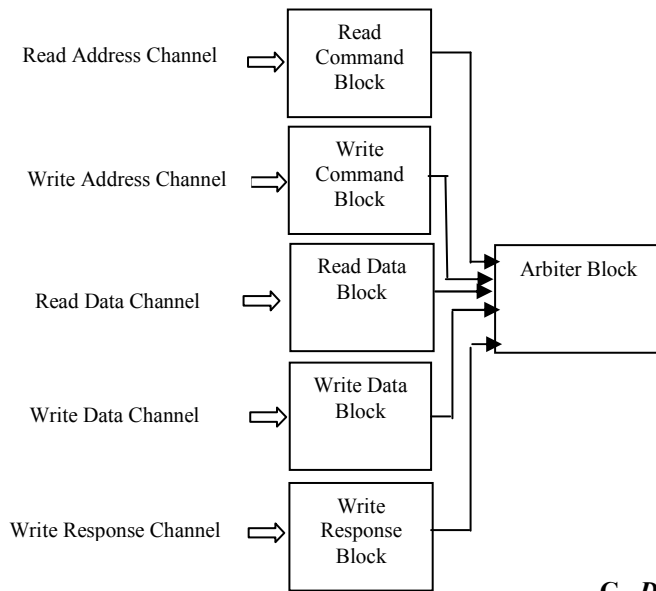


Fig.2 AXI Interface Block

B. AXI Access Manager:

The main function of the AXI-Access Manager is to convert the AXI commands into memory access commands for maximum utilization of the DDR3 Band Width. The DDR3 memory takes command only in burst 4 or 8 mode whereas the AXI command could be a smaller or longer

burst. The AXI access manager combines the command wherever possible to improve the performance and passes the final command to DDR3 controller. To insure maximum throughput it pre-fetches the commands from AXI interface converts into memory transactions and maintains locally. These stored commands are supplied to the DDR3 Controller whenever DDR3 Controller is not busy in the very next clock.

The AXI-IF block interacts with the AXI interface block and receives commands. The received commands are stored in the Storage control block. The Burst Manager converts the AXI command into DDR3 burst. The address control block is responsible for generation of address. The overall operation of various blocks in the AXI access manager is controlled by the Control Logic. The fig.3 shows the AXI Access Manager Block-

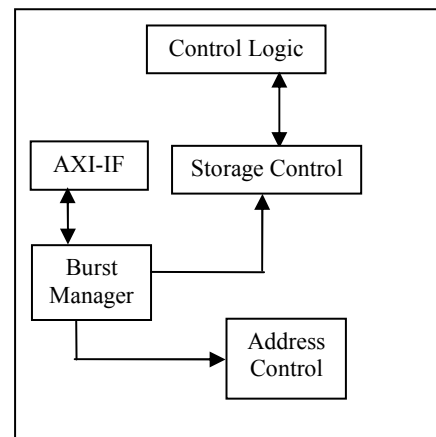


Fig.3 AXI Access Manager Block

C. DDR3 Controller :

The main function of DDR3 controller is to interact with the DDR3 memory. This is the heart of the AXI compliant DDR3 controller and responsible for understanding the DDR3 protocol and communicating with the DDR3 memory [4]. DDR3 Controller also issues Refresh, Power down, Self-refresh command along with the read or write command as per the user configuration [1]. The internal blocks of DDR3 controller are shown in the fig.4-

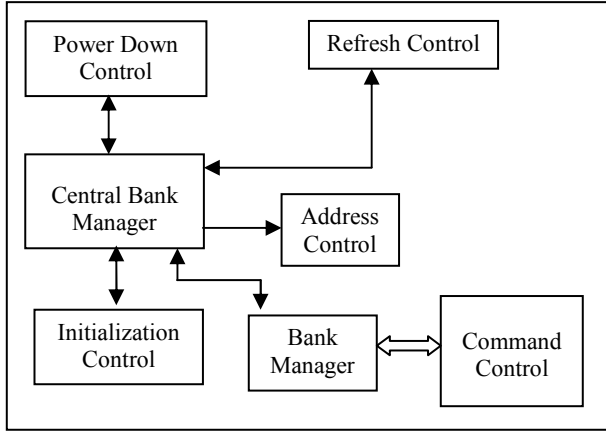


Fig.4 DDR3 Controller Block

The Power Down Control Block takes care of generating the power down command to the DDR3 memory whenever the host commands it to go to Power saving mode. The refresh control block takes care of refreshing the DDR3 memory as per the user supplied configuration. The initialization control blocks take care of initializing the DDR3 memory after reset. To control the timings of individual DDR3 banks it contains Bank manager which track the timing requirements for the individual DDR3 banks. The address control block is responsible for generating the address to the DDR3 memory. The Command control block interacts with the DDR3 memory and based on the Bank manager inputs drives the DDR3 bus. It is also responsible for receiving the data during the Read operation. The Central Bank manager coordinates between the individual Bank Managers and maintains the overall timing requirement of DDR3 memory.

II DDR FEATURES COMPARRISON:

DDR3 offers a substantial performance improvement over previous DDR2 & DDR memory systems. New DDR3 features, all transparently implemented in the memory controller, improve the signal integrity characteristics of DDR3 designs so that higher performance is achieved without an undue burden for the system designer. If proper consideration is given to any new DDR2 memory design, it can be a relatively easy upgrade to support DDR3 in the next generation design.

Feature	DDR	DDR2	DDR3
Data Rate	200-400 Mbps	400-800 Mbps	800-1600 Mbps
Burst Length	BL=2,4,8	BL=4,8	BL=4,8
No. of Bank	4 banks	512Mb : 4 Banks 1 Gb : 8 Banks	512Mb/1 Gb : 8 Banks
Prefetch	2 bits	4 bits	8 bits
CL/tRCD/tRP	15/15/15 ns	15/15/15 ns	12/12/12 ns
Source sync.	Bi-directional DQS (single ended Default)	Bi-directional DQS (Single/Diff. Default)	Bi-directional DQS (Differential Default)
Vdd/Vddq	2.5+/- 0.2 V	1.8+/- 0.1 V	1.5+/- 0.075 V
Reset	No	No	Yes
ODT	No	Yes	Yes

Table 1: DDR3 feature Comparison

III SIMULATION, TESTING & VERIFICATION

Verification is important part of complete design process. It takes almost 60% of design process flow so to minimize time to market in complete design we do verification process in parallel with design process. The verification of AXI compliant DDR3 controller is accomplished by sending the AXI transaction through the AXI Bus Functional Model. The response from the AXI compliant DDR3 controller is received by the AXI Bus Functional Model and is passed to the Checker. The checker also picks the expected response from the Local memory and compares it based on the comparison the environment prints passed and failed messages and the associated data mismatch if any. The verification environment for AXI compliant DDR3 controller is shown in fig.5.

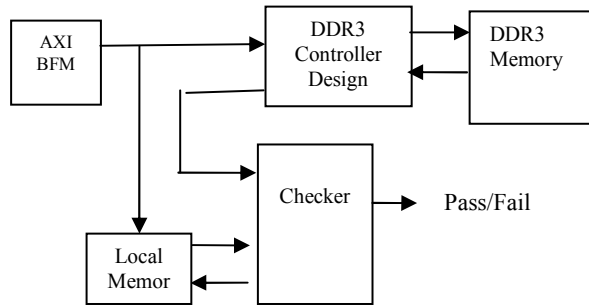


Fig.5 Verification of DDR3 Controller

The AXI Bus Functional Model consists of generator, driver, monitor and scoreboard. The generator generates AXI transactions. Driver is used to drive the DUT (Device under Test) here DUT is AXI interface. The AXI transaction is also passed to the local memory which prepares the expected response which is passed to the checker whenever the checker asks for it. Upon receiving the response from the DDR3 memory checker commands the local memory to give back the stored expected response and compares the DDR3 controller response with the expected response.

IV: SIMULATION RESULTS:

The below figures are showing snapshot of the DDR3 controller operation in various modes. The design has been coded in “Verilog HDL” language. The functional simulation tool used in IUS6.1 from Cadence.

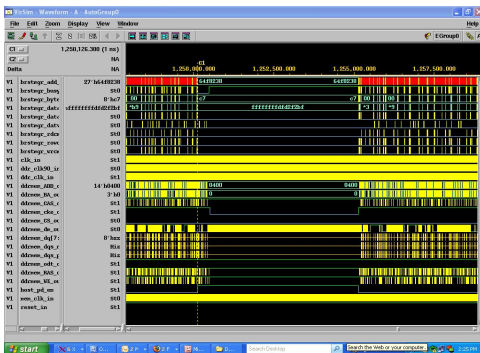


FIG6.1- AL0,BL4

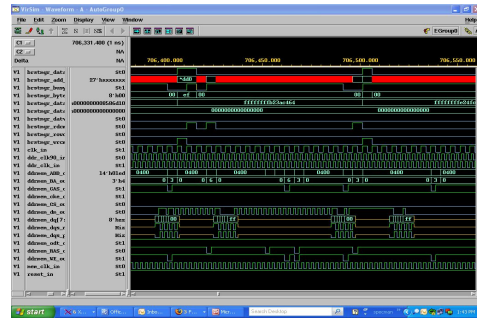


FIG.2- BL8, RANDOM READ/WRITE

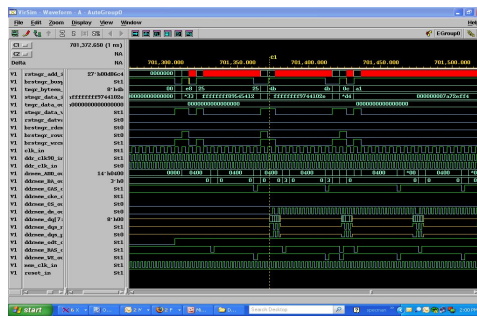


Fig6.3- AL0 , BL4

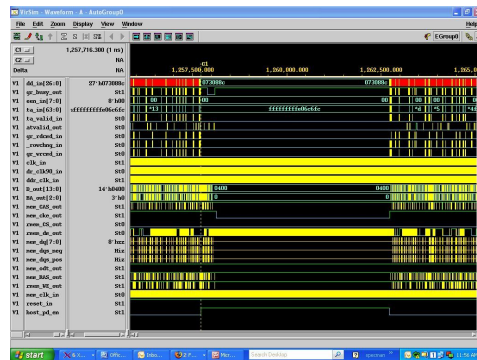


Fig6.4- BL4 , AL0

V: CONCLUSION

The design has been verified by the exhaustive functional verification. We examined the performance of the design by generating different type of AXI commands and noting down the time taken by the DDR3 controller in finishing them. In most of the scenario the throughput of the design is close to the theoretical max. The latency of the design is between 10 to 35 clocks based on the command generated and the internal DDR state.

VI: FUTURE IMPROVEMENTS

Future improvements in AXI interface block is to add more features like fixed address mode, address wrapping mode and write response signal generation other than OKEY response. In fixed burst, the address remain all the same for every transfer in the burst. This burst type is for every repeated accesses to the same location such as when loading or emptying a peripheral FIFO and wrapping burst is similar to an incrementing burst, in that the address for each transfer in the burst is an increment of the previous transfer address. However in wrapping burst the address wraps around to a lower address when a wrap boundary is reached. The write response signal other then OKEY are EXOKAY SLVERR and DECERR. Future Improvement in DDR3 Controller is to add Reorder block in between AXI Access Manager and DDR3 Controller block. The Reorder block will enhance the performance of complete DDR3 Controller because it sends same row address command first then sends other row address commands. When we switch the transaction from Row address X to other Row address Y first we have to close Bank corresponding to that Row address X means precharge that Bank and it take tRP time to precharge the particular bank. So to achieve high performance we have to order the same row address command with coming data from AXI Interface block. But at this time we are not implementing this block because this is applicable when we are firing random row address command means that depend on open the customer requirements because this reorder block increase cost and size of the chip.

REFERENCES

- [1] Churoo (Chul-Woo) Park, HoeJu Chung, Yun-Sang Lee, Jun-Ho Shin, Jin-Hyung Cho, Seunghoon Lee, Ki-Whan Song, Kyu-Hyoun Kim, Jung-Bae Lee, Changhyun Kim, *Senior Member, IEEE*, and Soo-In Cho. "A 512-Mb DDR3 SDRAM Prototype and Self-Calibration Techniques" Proc. IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 41,NO.4, APRIL 2006
- [2] K. Kim et al, "A 1.4 Gb/s DLL using 2nd order charge-pump scheme with low phase/duty error for high-speed DRAM application," in IEEE Int. Solid-state Circuits Conf. (ISSCC) Dig. Tech. Papers, 2004, pp. 212-523.
- [3] S.Lee et al, "A. 1.6 Gbs/pin double data rate SDRAM with wavepipelined CAS latency control," in IEEE Int. Solid-State Circuits conf. (ISSCC) Dig. Tech. Papers, 2004, pp.210-213.
- [4] H. Song et al, "A 1.2 Gbs/pin double data rate SDRAM with on die-termination," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, 2003, pp. 314-496.