Implementation of Memory Controller using Cadence Tools

T. Subha Sri Lakshmi
CVR College of Engineering/ ECE Department, Hyderabad, India
Email: rupashubha@gmail.com

***Abstract*—The Controller is designed which allows a high degree of programmability and interfaces many other memory devices which will be very flexible in nature to be programmed with other devices. To achieve throughput or latency multiple abstraction layers are present. The Controller provides the access of memory, executes in parallel form which leads to less logic utilization. This type of memory controllers are used in double data rate synchronous dynamic random access memory. The Memory Controller and Core Memory Controller are implemented using Cadence ASIC 45nm technology. Blocks are tested using ncvlog simulator, RTL schematics are generated using RTL Compiler and finally GDS-II file is obtained by using SoC Encounter. So the proposed Double data rate synchronous dynamic random access memory controller works with very less logic utilization, route and offset delay.**

***Index Terms*— Memory Controller, Core Memory Controller, Arbiter, ASIC Cadence- RTL Compiler, and SoC Encounter.**

I. INTRODUCTION

As with standard Synchronous dynamic random access memory [6] the design is pipelined and consists of various register banks which allows parallel operation to achieve higher bandwidth [1]. It performs read, write instructions by varying the length size i.e., 2, 4 or 8. A double data rate Synchronous Dynamic Random Access Memory consists of four register banks where as each register bank consists of multiple rows. Further, each row is divided into columns and consists of 32 bits of data. The number of rows and columns are reliant on the size of the on chip memory. The memory organization with four register banks will be executed with single command operation. Due to the size of the data bus, we cannot perform more than one write or read instruction. When one row is performing either read or writes operation, the other rows can be pre charge and activated to access the register bank. The schematic view of double data rate SDRAM is shown in the figure 1.

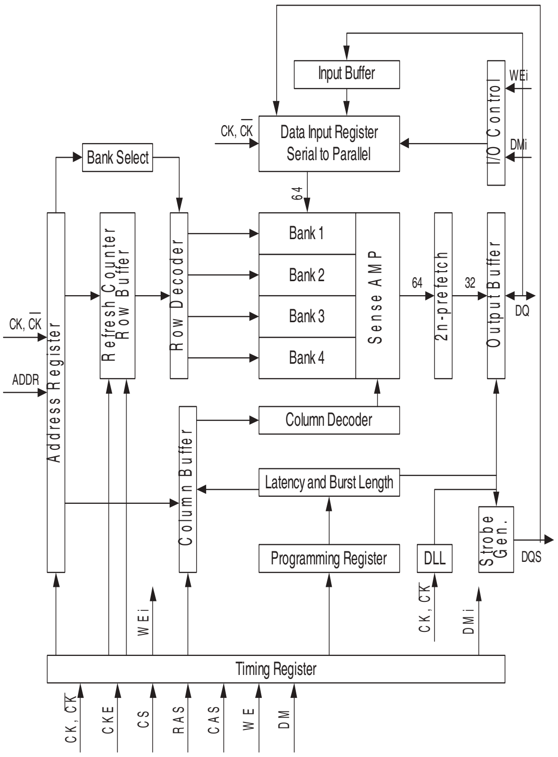
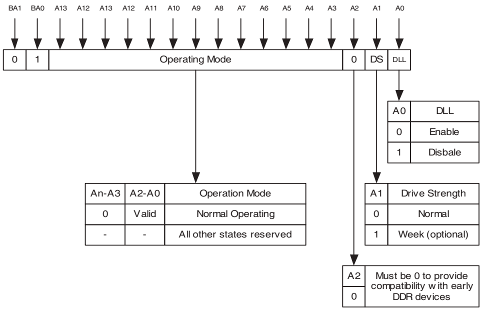


Fig 1: Schematic View of Double data rate

II. BLOCKS EXPLANATION

A. EXTENDED REGISTER SET

The Register set [4] and extended register set control word formats are used for writing data, which defines the specific mode of operation. The extended register set is shown in the below figure 2.



A0	DLL
0	Enable
1	Disbale

An-A3	A2-A0	Operation Mode
0	Valid	Normal Operating
-	-	All other states reserved

A1	Drive Strength
0	Normal
1	Week (optional)

A2	Must be 0 to provide compatibility with early DDR devices
0	

Fig 2: Extended Register Set

DOI: 10.32377/cvrjst1007

B. AHB INTERFACE

The Advanced High performance Bus Interface is a receiver interface as shown in figure3. The bus [3] acts as an input for the accessing port to the on chip memory and even for all the intellectual property cores which are present on the controller. The advanced high performance interface informs to the core memory controller about the starting location of the data when a read operation approaches the bus. The data is buffered and as soon as the interface has started to receive data from the DDR SDRAM it presents the data on the AHB bus. As soon as the write operation approaches the interface will store the data into the memory controller. Compared to random access memory, the advanced high performance bus works on lower frequencies for that purpose when the data appears the data will be registered before the write instruction is started.

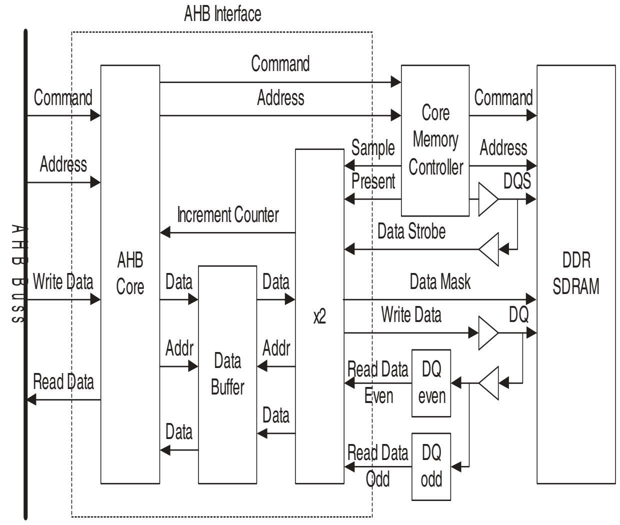


Fig 3: Schematic View of AHB Interface

C. MEMORY CONTROLLER

The Memory Controller [2] as shown in figure 4 is implemented to support a multiple advanced high performance bus receiver interfaces. For this purpose the two paths have been separated i.e., control path and data path. The core memory controller handles the control path and the data path which are inbuilt in each of the AHB interface of the core memory controller. The access of the data path, control path will be decided by the internal arbitrator. Advanced Peripheral Bus is used for the purpose of initialization. The memory controller is supported to handle multiple random access memory chips and other memory devices for the synthesis.

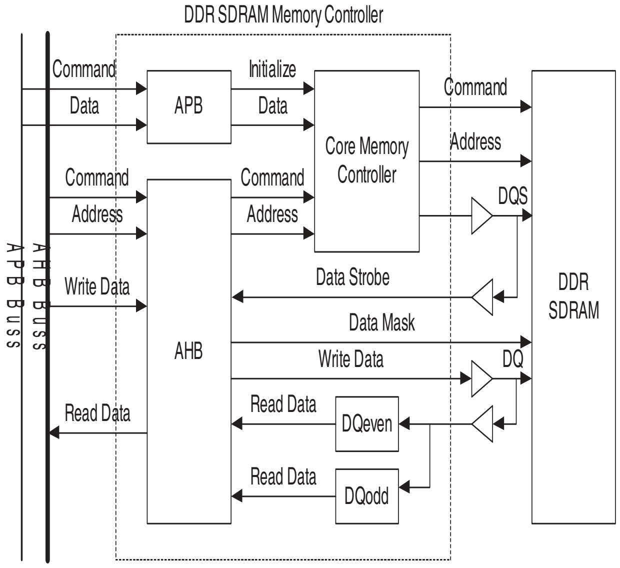


Fig 4: Schematic view of Memory Controller with AHB interface

D. CORE MEMORY CONTROLLER

The main purpose for the implementation of on chip core memory controller is to control all the command of instructions in multiple timings and all the activated rows will be present in the form queue. The main feature of arbitrator is that it will inform the current operation process and the next operation which is going to be executed because the activation of the rows will take huge amount of time. This function can be possible to activate the row ahead when the next instruction is not using the same register bank as present instruction. The on chip core memory controller divides the burst into multiple instructions which are useful to perform the entire burst operation as shown in the figure 5. The core memory controller also handles the activation of the next rows which are going to be executed.

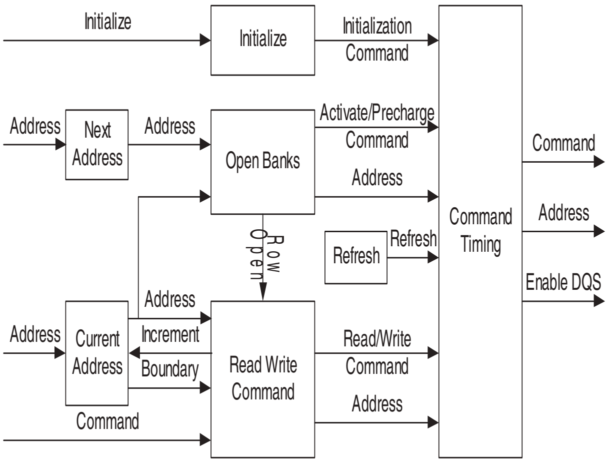


Fig 5: Schematic View of Core Memory Controller

DOI: 10.32377/cvrjst1007

E. ARBITER

As shown in figure 6 it is more advantageous to have multiple advanced high performance bus interfaces compared to having single bus interface [3], by adding more bus interfaces it leads to increase the transfer data rate of the device. The arbitrator is implemented by using round robin protocol or round robin with priority. The arbitrator will decide how many bus interfaces are required to implement the memory controller. By using the implementation of arbitrator with round robin fashion, it supports two advanced high performance bus interfaces. This makes the construction of arbitrator very simple because any additional functionality block is not necessary to decide which device of advanced high performance bus interface should grant the access for the current instruction if any instructions are in the form of queue.

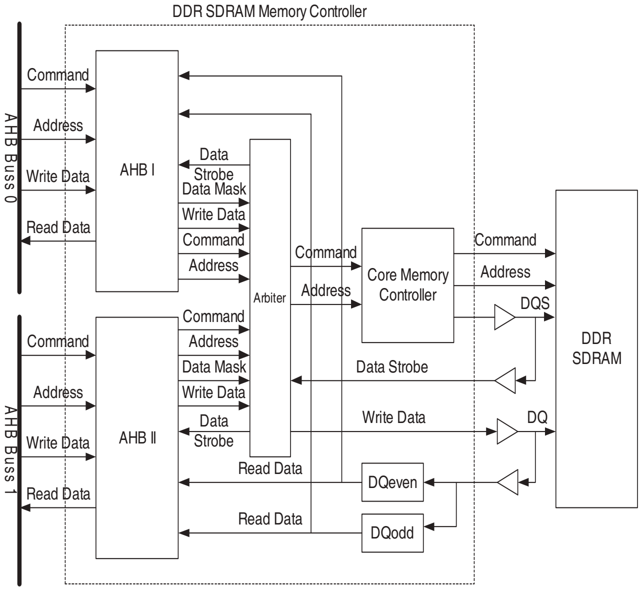


Fig 6: Schematic View of Memory Controller with 2 AHB Devices

III. BACKEND PROCESS

A. PLACEMENT

The placing and routing were performed by using SoC Encounter tools [5] for the 45nm technology TSMC libraries to obtain the layout structure for the device and it takes 70% of core utilization. The pads (GND, VDD) will be surrounded by the core in all the four directions (top, bottom, left, right) with subsequent width and length. The placement of memory controller is shown as is figure 7.

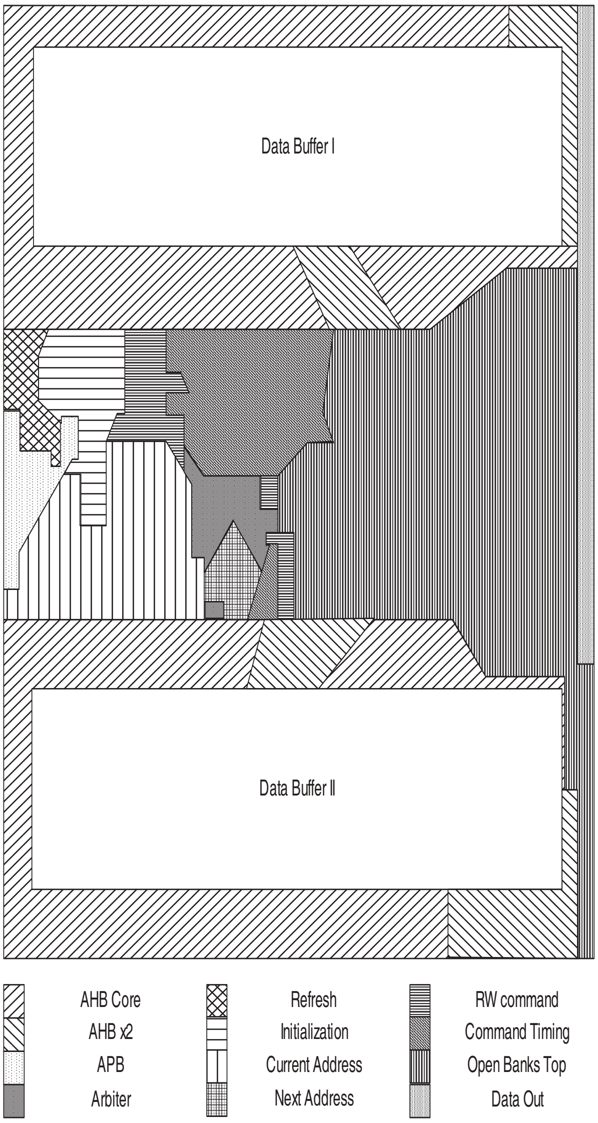


Fig 7: Placement

IV. IMPLEMENTATION RESULTS

All the blocks are implemented by using ASIC Cadence SoC Encounter tool with 45 nm technology libraries. Figure 8 shows RTL Schematic of Memory Controller, Figure 9 shows RTL Schematic of Core Memory Controller, Table 1 gives the pre, post clock tree synthesis report when it performs routing and finally figure 10 shows the IC chip fabrication layout structure which is named as GDS II file.

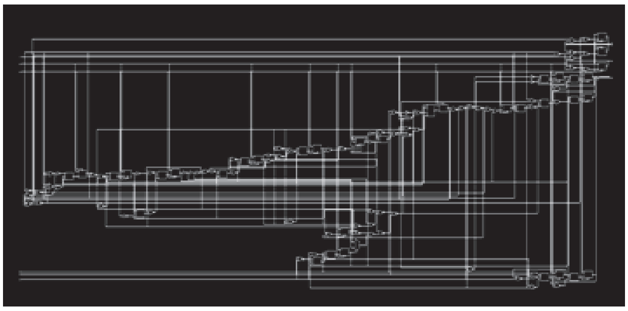


Fig 8: RTL Schematic of Memory Controller

DOI: 10.32377/cvrjst1007

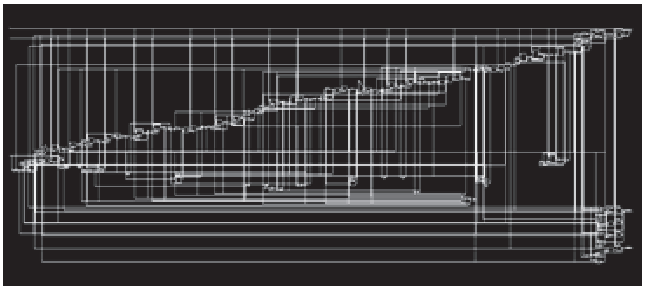
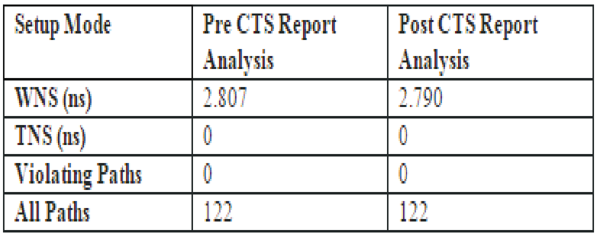


Fig 9: RTL Schematic of Core Memory Controller



Setup Mode	Pre CTS Report Analysis	Post CTS Report Analysis
WNS (ns)	2.807	2.790
TNS (ns)	0	0
Violating Paths	0	0
All Paths	122	122

Table 1: Pre and Post CTS Report Analysis Report

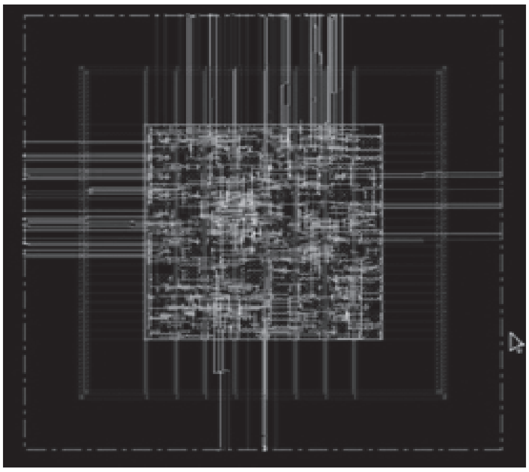


Fig 10 : GDS II File Core Memory Controller

V. Conclusions

All the blocks are verified by Ncvlog simulator and synthesized by using RTL Compiler and finally implemented on SOC Encounter and obtained IC chip layout i.e., GDS II file. The main advantage of memory controller is high through put or high latency. This type of memory controllers are used in random access memory devices. It provides layers of abstraction to achieve maximum throughput. It can be applied in personal computers and storage devices. As future work in place of two AHB bus interfaces more number of devices can be implemented to improve the performance of controller.

Refrences

[1] Jan M. Rabaey, “Digital Integrated Ciruits a Design Perspective”, 1996, Prentice Hall Inc, ISBN 0-13-394271-6.

[2] Samsung Electronics, “Key Points for Controller Design”, 1998 , DDr SDRAM/ SGRAM Application Note Samsung Electronics MPP- JLEE – Q4-98.

[3] ARM, “ AMBA Specification” (Rev 2.0), 1999, ARM IHI 0011 A.

[4] Sonali Ashokkumar Patel, Santhoshkrishna M, “Designof DDR SDRAM Controller For Embedded System”, Journal of Information, Knowledge and Research in Electronics and Communication Engineering, ISBN 0975 – 6779, Vol 02, Issue 02, Page 684 – 689.

[5]Http://lyle.smu.edu/~manikas/CAD_Tools/CPR/lib/CPR_Tutorial_2015Mar.pdf

[6] Tegze P.Haraszti, “CMOS Memory Circuits”, 2000, Kluwer Academic Publishers, ISBN 0 - 7923 – 7950 – 0.

DOI: 10.32377/cvrjst1007