Architecture of An AHB Complaint SDRAM Memory Controller

S. Lakshma Reddy
Metch student, Department of Electronics and Communication Engineering
CVSR College of Engineering, Hyderabad, Andhra Pradesh, India

A .Krishna Kumari
Professor, Department of Electronics and Communication Engineering
CVSR College of Engineering, Hyderabad, Andhra Pradesh, India

Abstract- -- Microprocessor performance has improved rapidly these years. In contrast, memory latencies and bandwidths have improved little. The result is that the memory access time has been a bottleneck which limits the system performance. As the speed of fetching data from memories is not able to match up with speed of processors. So there is the need for a fast memory controller. The responsibility of the controller is to match the speeds of the processor on one side and memory on the other so that the communication can take place seamlessly. Here we have built a memory controller which is specifically targeted for SDRAM. Certain features were included in the design which could increase the overall efficiency of the controller, such as, searching the internal memory of the controller for the requested data for the most recently used data, instead of going to the Memory to fetch it. The memory controller is designed which compatible with Advanced High-performance Bus (AHB) which is a new generation of AMBA bus. The AHB is for high-performance, high clock frequency system modules. The AHB acts as the high-performance system backbone bus.

Index Terms- SDRAM, Memory controller, AMBA, FPGA, Xilinx,

I. INTRODUCTION

In order to enhance overall performance, SDRAMs offer features including multiple internal banks, burst mode access, and pipelining of operation executions. Accessing one bank while pre charging other banks is enabled by the feature of multiple internal banks. By using burst mode access in a memory row, current SDRAM architectures can reduce the overhead due to access latency. The pipelining feature permits the controller to send commands to other banks while data is delivered to or from the currently active bank, so that idle time during access latency can be eliminated. This technique is also called interleaving. The design two way cable networked SoC, that is SDRAM controller connected by AMBA The AMBA AHB is for high-performance, high clock frequency system modules. The AHB acts as the high-performance system backbone bus.AHB supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripherals. It has their own bandwidth requirements and responding speed requirements for SDRAM. By analyzing the multiple accesses from the modules and the SDRAM Specifications such as its accessing delay, we take both side 1 and side 2 into consideration respectively. On side 1, we use bank closing control. On side2, the controller employs two data write FIFO to reduce the data access awaiting time, and uses 2 read FIFO to decrease the CAS delay time when reading data from SDRAM.

II. AMBA

The Advanced Microcontroller Bus Architecture (AMBA) specification defines an on chip communications standard for designing high-performance embedded microcontrollers. AMBA (Advanced Microcontroller Bus Architecture), is a bus standard devised by ARM with aim to support efficient on-chip communications among ARM processor cores. Now a days, AMBA is one of the leading on-chip busing systems used in high performance Soc design. AMBA is hierarchically organized into two bus segments, system- and peripheral-bus, mutually connected via bridge that FIFO data operations between them. Standard bus protocols for connecting on-chip.

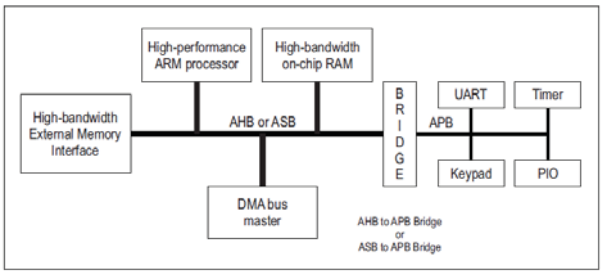

Fig.2.1. A Typical AMBA System

2.1. Advanced High-performance Bus (AHB)

AHB is a new generation of AMBA bus which is intended to address the requirements of high-performance synthesizable designs. It is a high-performance system bus that supports multiple bus masters and provides high-bandwidth operation.AHB is for high clock frequency system modules. It acts as the high performance system backbone bus. AHB supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripherals .Bridging between this higher level of bus and the current ASB/APB can be done efficiently to ensure that any existing designs can be easily integrated. An AMBA AHB design may contain one or more bus masters, typically a system would contain at least the processor and test interface.

2.2 Advanced System Bus (ASB)

The Advanced System Bus (ASB) specification defines a high-performance bus that can be used in the design of high performance 16 and 32-bit embedded microcontrollers. AMBA ASB supports the efficient connection of processors, on-chip memories and off chip external memory interfaces with low-power peripherals. An AMBA-ASB based microcontroller typically consists of a high-performance system backbone bus, able to sustain the external memory bandwidth, on which the CPU and other Direct Memory Access (DMA) devices reside.

2.3. Advanced Peripheral Bus (APB)

The Advanced Peripheral Bus (APB) is part of the Advanced Microcontroller Bus Architecture (AMBA)hierarchy of buses and is optimized for minimal power consumption and reduced interface complexity. The AMBAAPB should be used to interface to any peripherals which are low bandwidth and do not require the high performance of a pipelined bus interface. The latest revision of the APB ensures that all signal transitions are only related to the rising edge of the clock.

III. SDRAM

Random access memory (RAM) is a form of computer data storage. A random access device allows stored data to be accessed in very nearly the same amount of time for any storage location, so data can be accessed quickly in any random order. In contrast, other data storage media such as hard, CDs, DVDs and magnetic tape read and write.

3.1. SRAM

Static random-access memory (SRAM) is a type Of semiconductor memory that uses bistable latching circuitry to store each bit. SRAM exhibits data remanence, [1] but is still volatile in the conventional sense that data is eventually lost when the memory is not powered. Each bit in an SRAM is stored on four transistors (M1, M2, M3, and M4)that form two cross-coupled inverters.

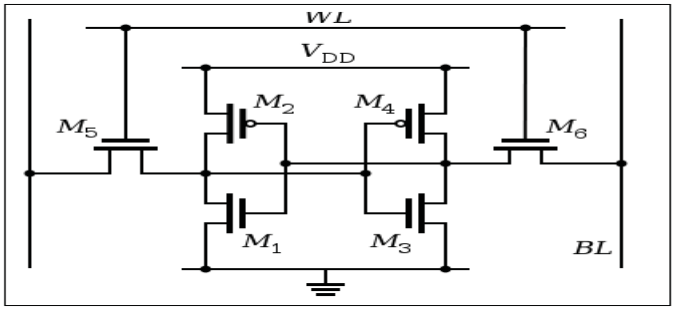

Fig.3.1. A Basic SRAM cell.

3.2. DRAM

Dynamic random-access memory (DRAM) is a type of random-access memory that stores each bit of data in a separate capacitor within an integrated. The capacitor can be either charged or discharged; these two states are taken to represent the two values of a bit, conventionally called 0 and1. Since capacitors leak charge, the information eventually fades unless the capacitor charge is refreshed periodically. Because of this refresh requirement, it is a dynamic memory as opposed to SRAM and other static memory. Unlike flash memory, DRAM is volatile.

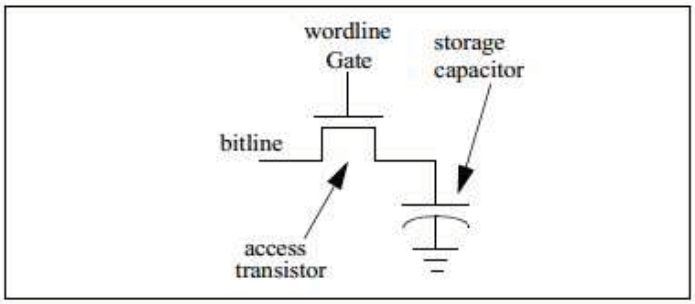

Fig.3.2. A Basic DRAM Cell

3.3. SDRAM

Synchronous dynamic random access memory (SDRAM)is DRAM that is synchronized with the system bus. Classic DRAM has an asynchronous interface, which means that it responds as quickly as possible to changes in control inputs. SDRAM has a synchronous interface, meaning that it waits for a clock signal before responding to control inputs and is there fore synchronized with the computer's system bus enabling higher speed. The SDRAM controller is capable of either 16-bit or 32-bit data path, and supports byte, half-word and word access. Bursts can be used for both write and read access.

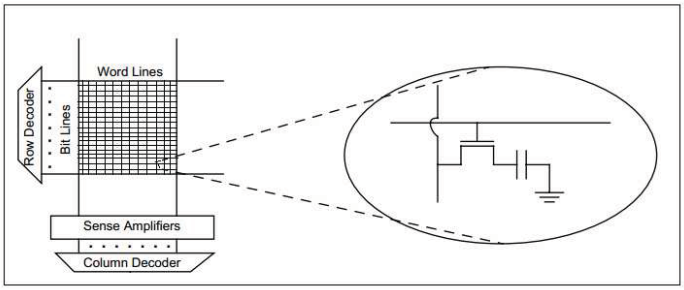

Fig.3.3. SDRAM Array structure

IV. MEMORY CONTROLLER

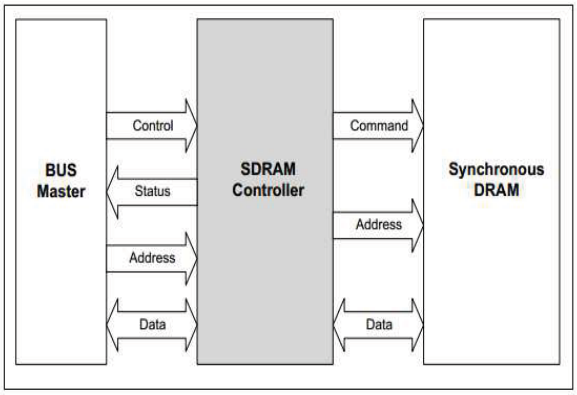

Fig.4.1. SDRAM Controller System

The controller is expected to synchronize data transfer between the processor and memory. To achieve this, the controller has to accept the requests from the processor side and convert them to a form suitable to the memory and execute the requests. Since the processor is faster than the memory, it is illogical to make the processor wait till each command is executed for it to give the next command. So the controller has to have some kind of storage so that it can buffer multiple requests while the processor continues with
other work the interface at the processor side of the controller
has to synchronize to the speed of the processor where as the
memory side interface.

4.1. Operation of SDRAM controller

Only one open row in an active bank can be accessed. An ACTIVE command can open a row and make active the particular row in the memory. A PRECHARGE command issued to memory can set the SDRAM to idle state, i.e. closing the open row in this memory. If every time after accessing a row AUTOPRECHARGE command is performed, and the next access is actually involving the same row of the same bank, an ACTIVE command needs to be applied again in order to access last open row. The memory controller basically consist of three main subparts, they are

a) 2 Read and 2 write FIFOs
b) Command generator
c) Command scheduler

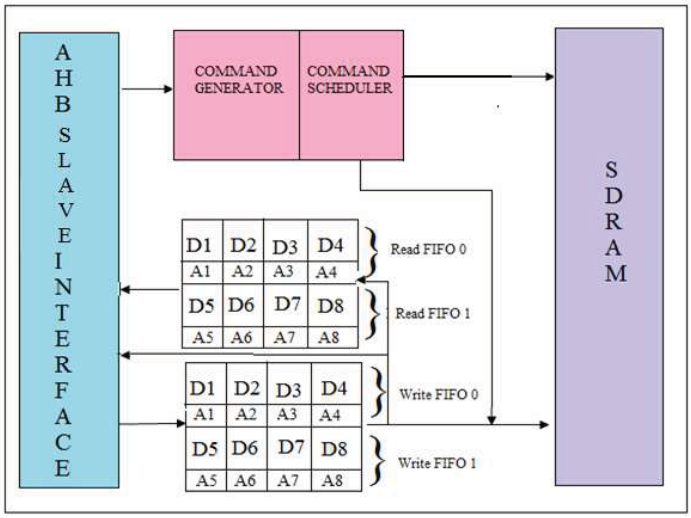

Fig.4.2. Architecture of SDRAM Controller

4.2. Read FIFO

After a preset number of clock cycles, the data is available on the output latches of the SDRAM for reading, and data is delivered to AHB bus and written to one of the read FIFO at the same time. In this way, we implemented pre fetching . According to the local principles of programs, the next read access is possibly a successive address to the current one. So the next data can be read from read FIFO, which can fasten the responding speed. In order to reduce the complexity of implementing read FIFO, data from SDRAM are stored in read FIFO 0 and read FIFO1in turn.

4.3. Writing or ping-ponging

As it is described about the AHB, in order to reduce the responding time to write access, write FIFO are used to pack and align the data when the AHB bus data transfer size does not match the SDRAM data bus width. Based on our previous research on FPGA designs, we didn't use only one write FIFO, instead, we use 2 FIFO

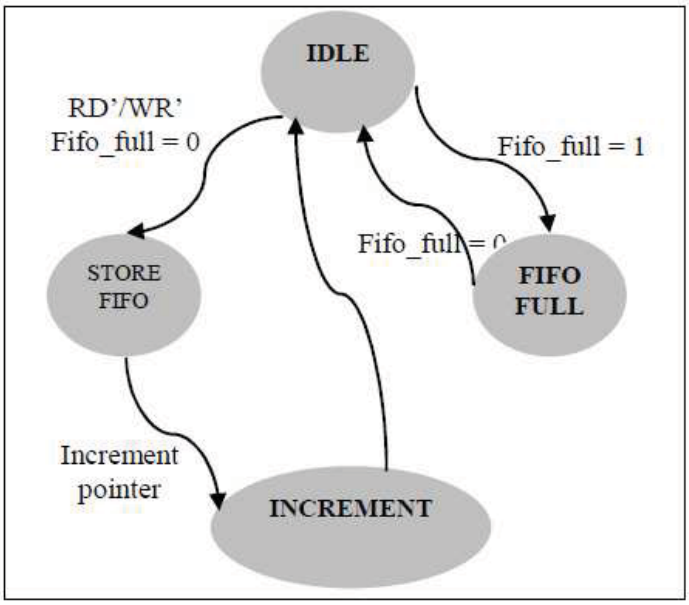

Fig.4.3. FIFO state machine

In order to keep the data consistency among the read FIFO, write FIFO and the SDRAM, whenever the data is written to write FIFO, match the write address with the addresses of data in the read FIFO. If the address matches, data will be written to read FIFO at the same time it is written to write FIFO. When reading SDRAM, match the read address with the read FIFO address and the write FIFO address first to see if the data can be read from those FIFO. This technique is called as Read-Update-Logic.

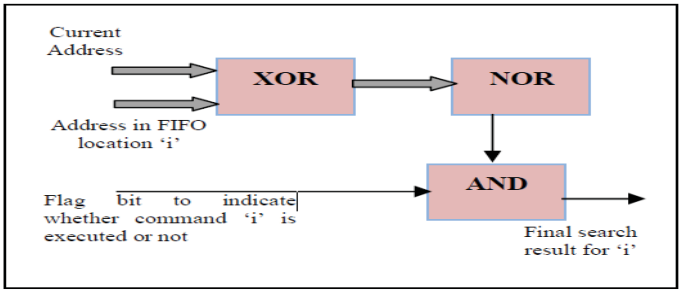

Fig.4.4. Search Logic

4.4. Command Generator

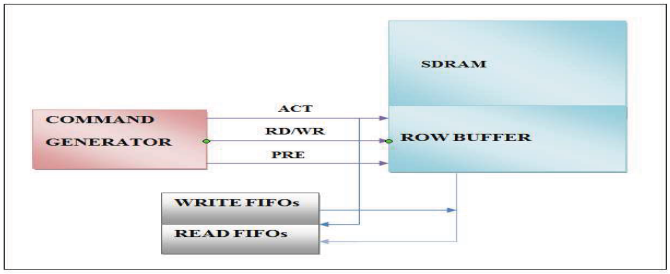

Fig.4.5. Command generator Diagram

Command generator basically generates the command which serve the SDRAM without violating timing constraints. Coming to the command priorities, the read and write commands have high priority, as they put data on the bus. Recharge and activate commands have lower priorities. The command generator works on the state machine shown in the figure below to take the decision for which command has to be generated for what condition.

4.5. Command Scheduler

The command scheduler gives the timing analysis for all the commands which are generated from the command generator. A parameter T is defined to every operation of the command taking place as shown in the figure and the relevant timing parameters which are described for our memory.

V. ADVANCE HIGH PERFORMANCE BUS INTERFACE

AHB is a new generation of AMBA bus which is intended to address the requirements of high-performance synthesizable designs. It is a high-performance system burst that supports multiple bus masters and provides high-bandwidth operation. AMBA AHB implements the features required for high-performance, high clock frequency systems including

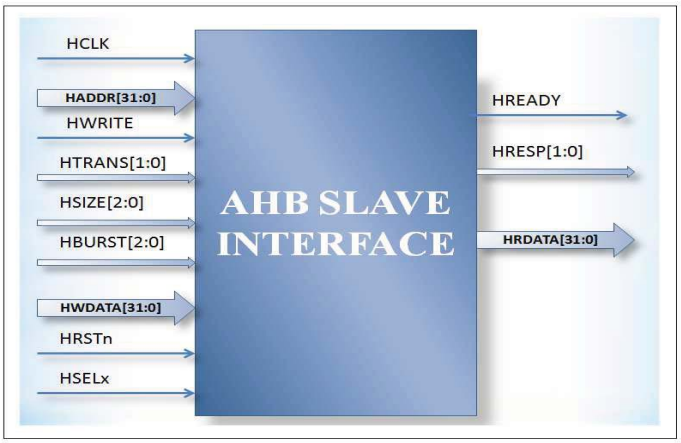

Fig.5.1. AHB slave Interface module

5.1. Basic transfer

An AHB transfer consists of two distinct sections:

- The address phase, which lasts only a single cycle.
- The data phase, which may require several cycles. This is achieved using the READY signal. In a simple transfer with no wait states
- The master drives the address and control signals onto the bus after the rising edge of HCLK.

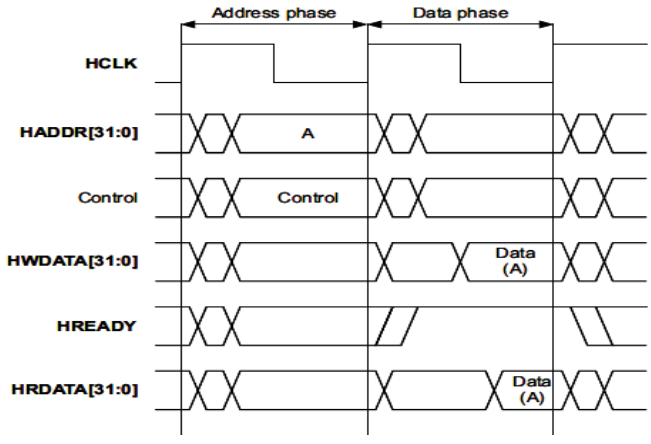

Fig.5.2. Simple transfer

VI. RESULTS

A. SIMULATION RESULTS

1. SDRAM

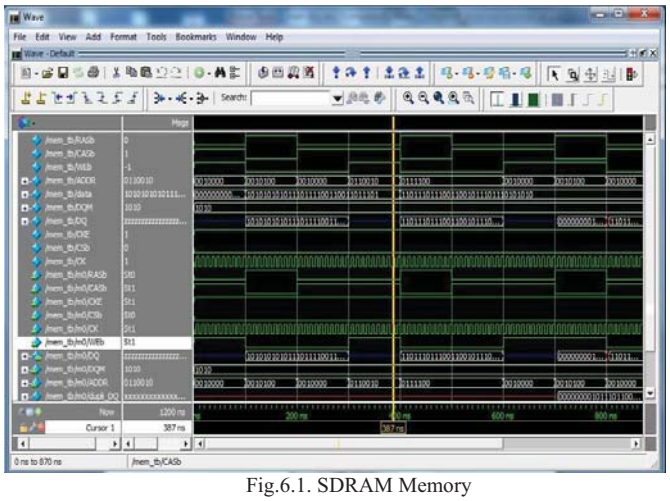

Fig.6.1. SDRAM Memory

2. Modified SDRAM Controller with FIFOing

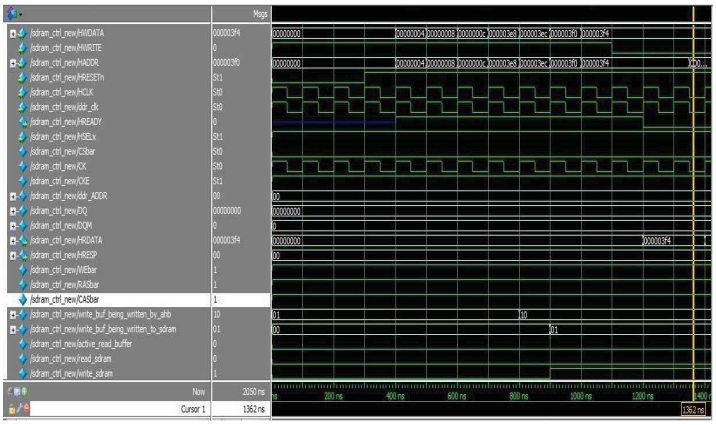

Fig.6.2. Modified SDRAM Controller

VII. FUTURE SCOPE

The XDR DRAM (extreme Data Rate DRAM) memory controller can be implemented. This technology can achieve every high clock speeds. The much high speed memories can further be designed using the DDR SDRAM (double data rate SDRAM) and XDR DRAM.

VIII. CONCLUSION

An AHB interfaced high-performance SDRAM Controller has been proposed, verified and evaluated we find this SDRAM controller has high performance by taking good use of the features of SDRAM architecture and utilizing the well-known techniques such as data FIFO ing and ping-ponging and burst-mode-data transfer. The testing results shows 91.66% of reduced delay with FIFOing when compared to the latency produced with FIFO ing t, which is nothing but the in-built memory used within the SDRAM controller to improve its performance.

REFERENCES

[1] "AMBA Specification (Rev2.0)", ARM Inc.

[2] Ching - SDRAM Controller Applications". IEEE J. Solid- StateCircuits,Vol.39, Nov. 2004.Che Chung, Pao-Lung Chen, and Chen-YiLee "Delay- Locked Loop for DDR

[3] Micron Technology Inc.Synchronous DRAM Data Sheet,2001.

[4] Hynix Semiconductor Inc., SDRAM Device operationRev.1.1, Sep.2003.

[5] "Memory Controllers for Real-Time Embedded systems" Benny Akesson Kees Goossens vol. 3, no. 3, pp. 75–77, Mar1999.

[6] Clifford E. Cummings, "Synthesis and scripting Techniques for

[7] Designing Multi-Asynchronous Clock Designs", Sunburst Design, Inc

[8] Samir Palnitkar, Pearson 2nd edition "Verilog HDL, A Guide to Digital Design and Synthesis.