Customized Controller Design of Partitioned DDR SDRAM for RF Data Acquisition System

Bhargav B K[1], Veena N[2], Girish G K[3]

[1, 3] Dept of ECE, Nitte Meenakshi Institute of Technolgy, Bengaluru, India.
[2] Director of Design, Centum Electronics Ltd., Bengaluru, India

***Abstract*—DDR SDRAM (Double Data Rate Synchronous Dynamic RAM) controllers are used to control and command the operations such as recording and reading of data from DDR SDRAM. In real time data acquisition system, the data needs to be stored at very high speed and in environments of outer space. High speed ADCs (Analog to Digital Convertors) are used to convert incoming analog data into digital data for further processing to estimate various pulse parameters. Due to effects like Single Event Upsets (SEU), re-programmable FPGAs cannot be used but high speed OTPs (One Time Programmable) such as Actel FPGA (Field Programmable Gate Arrays) is used for designing the controller.**

***Index Terms*—DDR SDRAM, Data Acquisition, SEU, OTP Actel FPGA**

I. Introduction

Data storage is an integral part of data acquisition systems for space application and DDR SDRAMs are the strongest contenders for this purpose. Due to their high speed operation at both edges of clock, and low cost and on board space compared to other high speed storage devices, DDR SDRAM is preferred. In this paper, the design of DDR controller for a partitioned DDR is proposed where customized addresses are generated for Writing and Reading from DDR. This idea is applied for a space application targeted on Actel AX1000 FPGA using Microsemi Libero IDE tool. The paper highlights switching of addresses between two partitions based on an input threshold. The finite state machine, cycle timings, refresh timings are as per data sheet of DDR being used [1].

II. Interfacing DDR with AX1000 FPGA

DDR interface FPGA has 144 data lines and hence five DDRs each of 32 bit wide and of 2 Gb capacity are used. Address lines are of 14 bits wide and are extracted from 30 bit custom address generated exclusively in different modules. A finite state machine is implemented for switching of various cycles. As indicated in fig (1), Clock is passed through LVDS (Low Voltage Differential Signaling) buffers to operate at low voltage and eliminate differential noise.

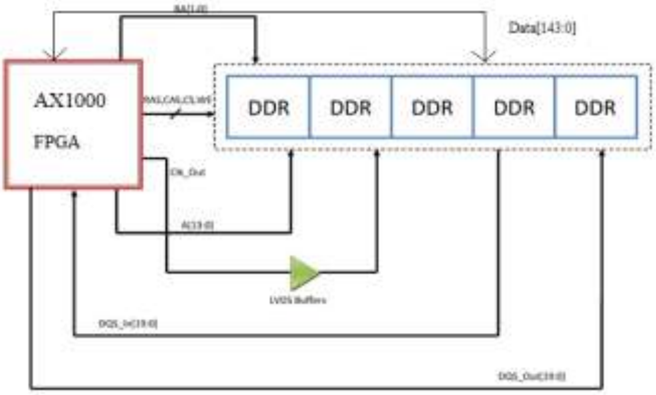

Fig 1: DDR interface design

Write cycle is designed for a burst access of eight and has eight write commands which makes a total of 64 accesses per cycle. Row activation [6] is done before issuing write command [6] and RAS (Row Address Strobe) address is sent at the address lines. t_{RCD} (RAS to CAS delay) delay is met and a write command is issued by sending corresponding CAS (Column Address Strobe) address at the address bus. Entire write cycle including activation and eight write commands using four control signals are shown in fig (2).

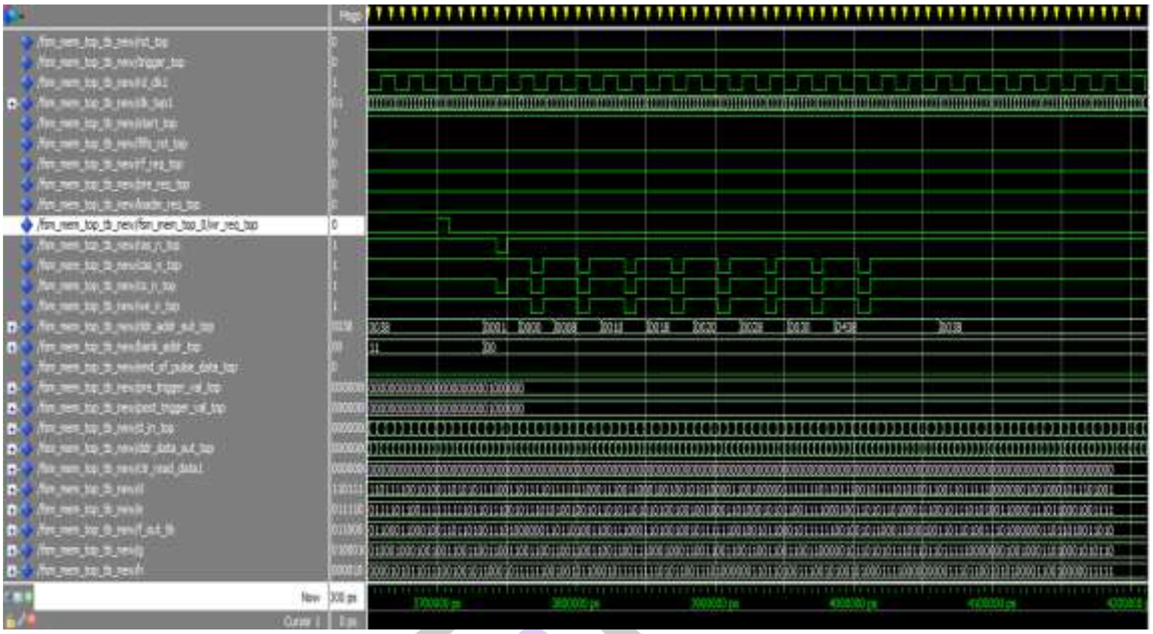

Fig 2. Write cycle of burst of 8X8

Similar to write cycle, Read cycle has a burst of eight accesses and has only one read command per cycle as shown in fig (3), which makes a total of eight read accesses in one read cycle. Similar to write cycle, activation and read commands [6] are issued with RAS and CAS addresses at the address bus maintaining delays [2].

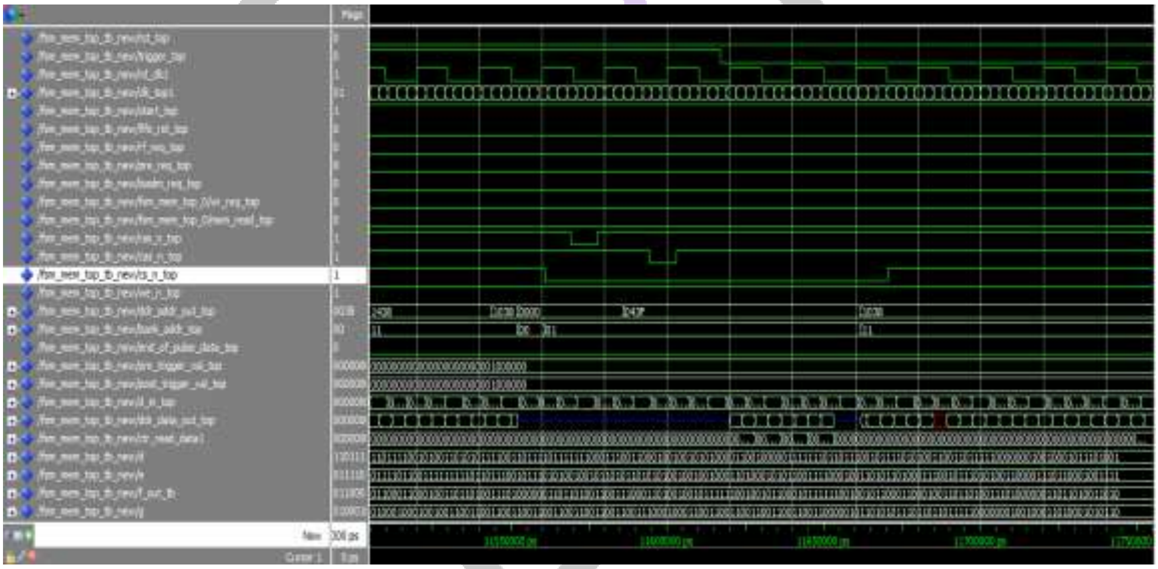

Fig 3. Read cycle of burst 8X1

III. DIVIDING DDR SDRAM

Data that is to be stored is primarily characterized whether it is valid or not. This is done by a threshold detection circuit which comes as an input to controller FPGA and is called as *trigger*. If trigger is high, the incoming data is treated as valid, or else it is invalid. These valid and invalid data are stored in separate partitions as shown in figure (4).

There is a customized 30 bit address which is designed consisting of RAS, CAS, BA (Bank Address) and information regarding mode register. These addresses are to be extracted and sent to DDR address bus at relevant commands.

BURST LENGTH	/CAS	/RAS	BANK ADDR	/CAS

29 26 20 7 5 0

It is clear that the data coming at the data bus of DDR is not always valid. But there is a need to store all the data that is coming on the data bus irrespective of whether it is valid or not. If all the data is stored into a single portion, there is a possibility of DDR SDRAM getting filled up. If the DDR gets filled up, overwriting of the data takes place which may lead to loss of data that belongs to valid region of trigger. Therefore, the DDR is divided into two equal and separate partitions, such that all the data that is not in the range of trigger will be stored in one partition and all the data corresponding to trigger (valid data), is written into partition 2. By dividing the DDR SDRAM in this fashion, there is no loss of valid data that is being written into partition 2 because of overwriting of invalid data into it. However, there is a possibility of data being overwritten into partition 1. This will not create a problem because of data being read out as and when the trigger occurs.

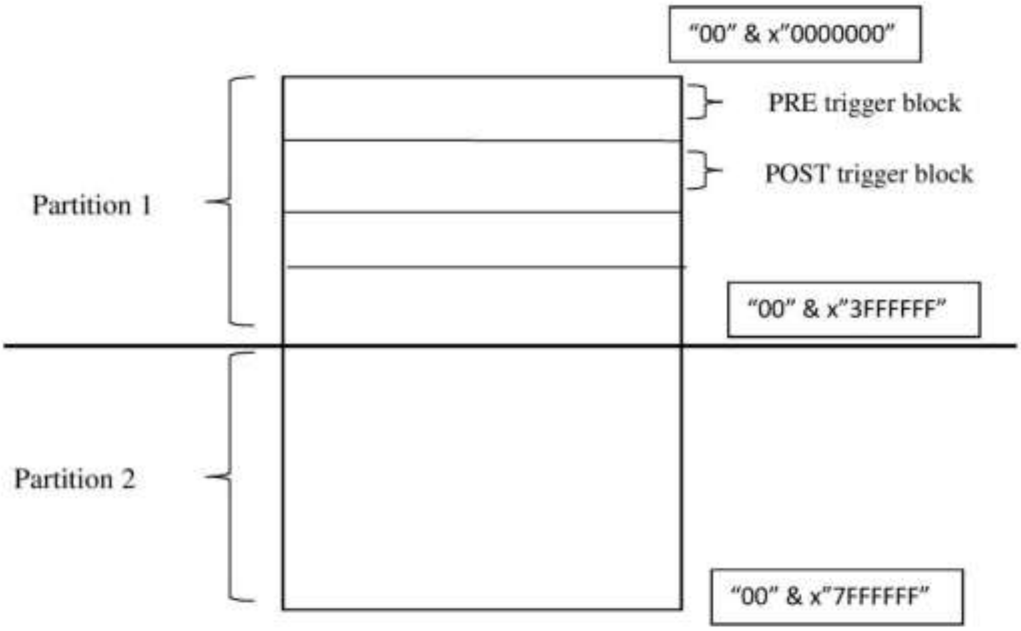

Fig 4. Division of DDR SDRAM

Further, the partition 1 is again subdivided into blocks of 1Mbit locations. These locations are divided in such a way that they alternately correspond to pre trigger and post trigger blocks pertaining to one trigger.

If the DDR is divided based on the addresses, the WRITE/READ happens for a particular row in one or more banks. While the WRITE/READ cycle is happening, the addresses keep incrementing in such a way that it takes less than 7.8µs of access time for that row in a particular bank. This releases the address bus for refresh cycle to take place in the previous bank. Also the Refresh cycle is skipped if in a particular row, a WRITE access is taking place.

The division of DDR starts from the first address "00" & x"0000000". Half of DDR memory is allotted for partition 1 which implies that up to the address, "00" & x"3FFFFFF", all the storage corresponds to partition 1. Here it can be noted that the 26th (MSB) bit of the address is '0', which is the indicator that the address belongs to partition 1. The memory addresses in which the 26th bit is '1', will indicate that the addresses correspond to partition 2 where valid data is stored. The partition 2 addresses will start from "00" & x"4000000" and end at "00" & x"7FFFFFF".

IV. Generation of Addresses for writing into DDR

Whenever the START of acquisition command is given, the data is available for storage. If the trigger is not present, the data is continuously written into partition 1. Since partition 1 is divided into sub blocks of 1Mbits each, the data coming before the occurrence of trigger will be written into pre trigger block of partition 1. The addresses are incremented after every write cycle end. The 30 bit address is chosen in such a way that bank addresses will be incremented after end of cycle and not row address. Thus activation of another row is not required after incrementing addresses after every write command. The addresses of pre trigger block will be incremented till the end of the block is reached and then the addresses are rolled back to starting address of pre trigger block similar to a circular buffer fashion. The data is continuously overwritten in the pre trigger block until the trigger.

If the trigger has occurred, the current address of pre trigger block is captured and the pointer is shifted to the starting address of partition 2. The pointer remains in partition 2 as long as the trigger remains high. The addresses of partition 2 get incremented at the end of every write cycle belonging to partition 2.

After the trigger goes low, the pointer is shifted to partition 1 post trigger block. In this block, the information belonging to a particular trigger after it goes low, is being written. Based on the post trigger value, the number of locations that needs to be written in post trigger block is computed. The data is written into the post trigger block and the addresses of post trigger block keep on incrementing after every write cycle end. After writing into those many number of locations, the current address of post trigger block is captured. The pointer shifts to next pre trigger block of 1Mbit for writing the data that is available after the post trigger data is written.

V. Generation of Addresses for reading from DDR

The data that has to be read from DDR has to contain data belonging pre trigger, trigger and post trigger locations. Hence there has to be a switching of addresses from pre trigger block, partition 2 and then to Post trigger block. For this purpose, three flags are generated to differentiate between the addresses and the order of their generation is FLAG_PRE_TRIGGER, FLAG_PART2 and FLAG_POST_TRIGGER.

FLAG_PRE_TRIGGER is generated for reading from pre trigger block of partition 1. When this flag is high, the addresses of pre trigger blocks are incremented by 8 at the end of every read cycle. This flag will not be generated if the pre trigger value is zero.

FLAG_PART2 is generated in order to read the data of the trigger that is present in partition 2 of DDR. The part2 addresses are incremented by 8 at the end of every read cycle end. This flag remains high until the MEM_ADDR reaches the Part2_End_Address of a trigger.

FLAG_POST_TRIGGER will correspond to reading of data from partition 1 post trigger block. The addresses of post trigger block are incremented at the end of read cycle belonging to post trigger locations. This flag is not generated if post trigger value is zero.

VI. Simulation Results and conclusion

The results show switching of write addresses based on occurrence of trigger. If trigger has occurred while a cycle has not completed, the addresses are switched at the end of cycle to avoid ambiguity.

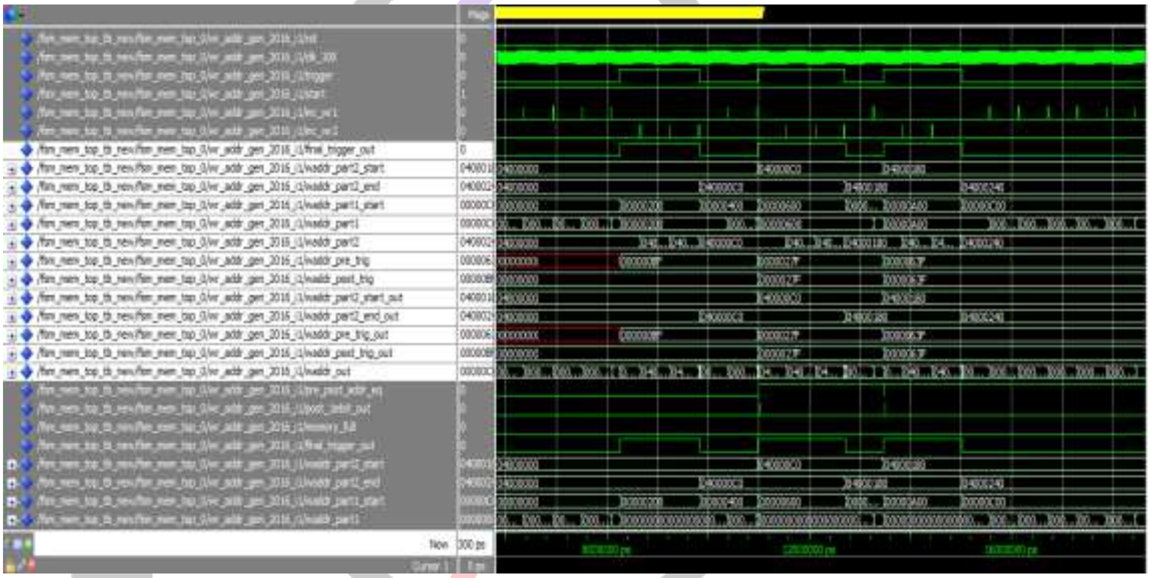

Fig 5. Switching of write addresses based on trigger

Fig (5) shows switching of addresses between partition 1 and partition 2, based on occurrence of trigger.
Reading of data is based on three flags and generation of all three flags are shown for non-zero pre and post trigger values.

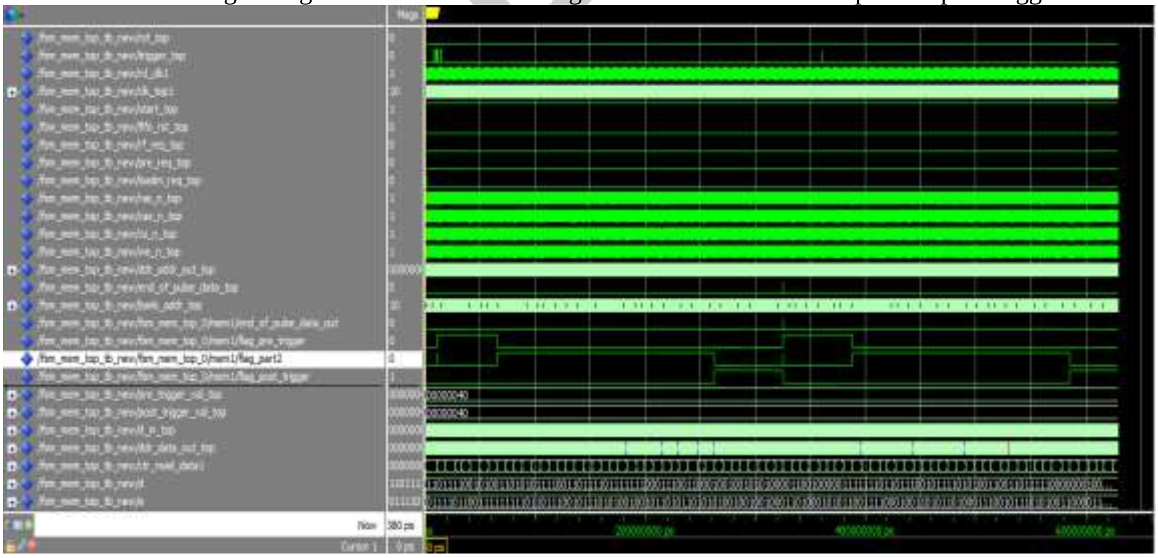

Fig 6. Flag generation for reading out data

Fig (6) shows generation of flags in their order of pre trigger, trigger and post trigger regions of data during reading of data from DDR.

Conclusion:

A controller for DDR SDRAM is designed with custom address generation for writing and reading RF data from DDR. A 30 bit address is chosen in such a way that for a particular cycle of writing and reading, upon the incrementation of addresses, only bank addresses will be updated and not row addresses. This will eliminate row activation before issuing any command (WRITE OR READ) and t_{RCD} delay can be reduced. Further, if DDR is divided based on addresses, the refresh timing of 7.8us can be achieved without any disturbance of priorities. The simulation results show

REFERENCES

[1] "Micron Technology", Inc. DDR SDRAM 1Gb DDR x4 x8 x16 features (Data sheet)

[2] "Microsemi Axcelerator Family FPGAs" , Microsemi Corporation, Version 18, March 2012 (Data sheet)

[3] "VHDL Modelling Guidelines", European Space Agency, European Space Research and Technology Centre ASIC/001 Issue 1 September 1994

[4] "NASA Technical Handbook", Programmable Logic Devices, December 2013

[5] Zude Zhou, Songlin Cheng, and Quan Liu, "Application of DDR Controller for High-speed Data Acquisition Board," Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06), 2006

[6] "How to Use DDR SDRAM", Elpida Memory Inc., E0234E30 (Ver.3.0), April 2002 (K) Japan (Data sheet)

[7] http://www.radio-electronics.com/info/data/semicond/memory/sdram-ddr-memory.php

[8] https://www.pctechguide.com/computer-memory/ddr-sdram-explained

[9] Chen Shuang-yan, Wang Dong-hui, Shan Rui Hou Chao, "An Innovative design of DDR/DDR2 SDRAM Compatible Controller", ASICON International Conference, pp 62-66 24th Oct 2005.

[10] Guo Li, Zhang Ying, Li Ning, and Guo Yang, "The Feature of DDR SDRAM and the Implementation of DDR SDRAM Controllers via VHDL", The Journal of China Universities of Posts and Telecommunications, 2002, vol.9, no. 1, pp. 61-65.