REVIEW ON DDR SDRAM CONTROLLER

[1]Divya Tandon, [2]Nagendra Sah

[1]M.Tech Student, [2]Associate Professor

[2]ECE Department

[2]Punjab Engineering College (Deemed to be university), Chandigarh, India

Abstract- With the improvement in processor performance speed limitation of memory interface is becoming bottleneck of overall system performance. SDRAM controller interface between memory and host. Implementation of efficient controller can overcome the speed limitation of memory thus improving the overall system performance. In DDR interface data transfer occurs on both clock edges which help in faster data transfer.
Index terms- SDRAM, DDR

I. INTRODUCTION

Dynamic RAM (DRAM) is comparatively low cost and furthermore have more memory thickness .Be that as it may, the working rate of the equivalent is exceptionally low and can be utilized at all the spots where the planning isn't the matter of concern, application, for example, computer games, Camera and furthermore some little electronic hardware. To improve the speed of DRAM there is requirement for an extra structure that fills in as a communicator between the gadget and the DRAM and furthermore gives all the important sign to the usefulness of the memory component. The go between is called as controller or memory controller. It will play out all the fundamental direction age and give the synchronization among memory and the gadget under task. SDRAM controller can be built as per the application and used by the user as the only task of the controller is to initialize the memory and provide some commands like refresh, Read, Write, etc.
Fig 1 shows physical structure of DRAM device . A DRAM has eight banks and fringe circuits shared by the eight banks. In a bank, different subarrays share a line address decoder, and each sub-array exhibit has its very own physical row-buffer. since a bank has a row-buffer in a logical view, only one physical row-buffer can be activated. The sense-amplifiers are associated with bitlines and cells legitimately, which assume a key job to decide row-access time.

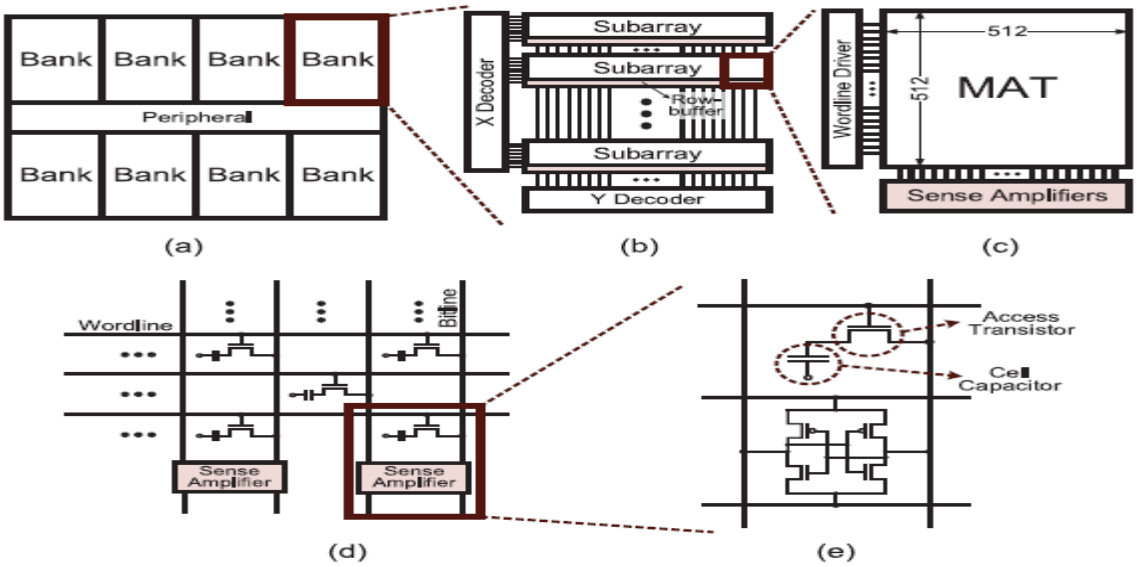

Fig 1 Physical structure of DRAM

DRAM access mechanism is described with Fig. 2 For a memory controller to get to a particular information in DRAM, a row-access command (ACTIVATION) ought to be first issued. The a row-access command summons a row-activation; the objective column in the bank is actuated and stored to the corresponding row-buffer.After the row-activation is done (after tRCD), one column belonging to the row can be accessed by issuing a column-access command (READ=WRITE). To access another column in the same row, another column access command can be issued after column-access-cycle time (tCCD). But, to access another row in the same bank, already open row should be first closed by issuing a row deactivation command (i.e., PRECHARGE). This command can be issue tRAS after issuing the preceding row-access command. Then, after tRP (row-deactivation time), another row-access command can be issued. To total up, row-access cycle time becomes tRC (=tRAS + tRP).

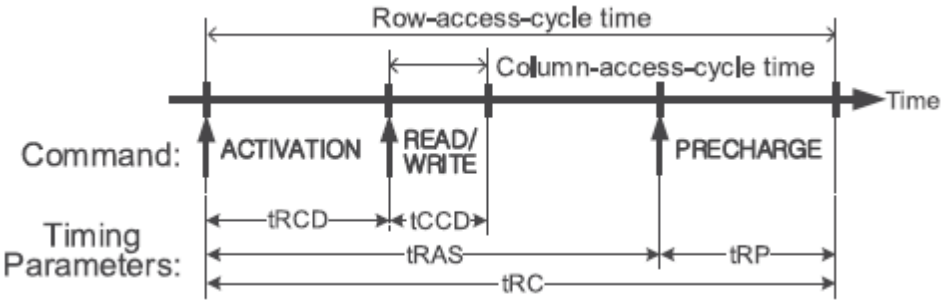

Fig 2 DRAM access commands and relevant timing parameters

Due to cost-sensitive nature of DRAM market, there is very little motivation to modify the DRAM internal architecture. Thus tweaking with the memory controller architecture offers a great scope in hiding the DRAM internal latency.

II. LITERATURE REVIEW

Hiroaki Ikeda and Hidemori Inukai in 1999 presented an overview of the high-speed DRAM architecture developments where he discussed developments on density growth, interface technology, memory-core architecture, and DRAM + ASIC technology. In density development 2X DRAM density growth every two years is in line with system requirement. For interface technology developments, the author has discussed the double-data-rate feature and terminated bus structures. In core architecture highly segmented memory-array architecture and the hierarchical memory-core architecture will be the candidates for future DRAM core technology. [1]

Sven Heithecker and Rolf Ernst in 2005 proposed an SDRAM scheduler IP that supports several concurrent access sequence types with different requirements including predictable periodic real-time sequences and cache accesses with a minimum latency objective. They have introduced a combination of prioritized scheduling and flow control by which the real-time behaviour and system performance was significantly improved. They have showed that allowing multiple accesses closely following each other (access bursts) also improved the system performance while the real-time behaviour was not affected. [2]

Nagendra Gulur et al. in 2011, proposed re-organization of the existing single large row buffer in a DRAM bank into multiple smaller row-buffers which simultaneously improves performance and reduces energy in DRAMs. The single large row-buffer is replaced with 4 small row-buffers, each one fourth the size of the original row-buffer. Activate and Precharge energy consumption is significantly reduced in the proposed organization owing to two improvements: hit-rate directly reduces the number of activate and precharge operations. As activate and precharge operations now work on smaller rows, fewer capacitors need to be charge/discharge. This methodology improves performance by 35.8%, 14.5% and 21.6% in quad, eight and sixteen core workloads along with a 42%, 28% and 31% reduction in DRAM energy. [3]

Yoongu Kim et al., 2012, proposed mechanisms SALP-1, SALP-2, and MASA that overlap the latencies of different access requests that go to the same bank. They overlap different components of the bank access latencies belonging to multiple requests that go to different sub arrays within the same bank thus removing the negative impact of bank serialization. These three mechanisms show that they significantly improve performance for single-core systems compared to conventional DRAM: 7%/13%/17% for SALP-1/SALP- 2/MASA, respectively and also interact positively with application-aware memory scheduling algorithms and further improve performance for multi-core systems.[4]

Li Wang et al. in 2012, proposed DDR SDRAM Controller Based on FPGA in Satellite Navigation System in which a new strategy of reading and writing, similar to the ping-pong storage strategy is implemented. The write address generation unit and the read address generation unit generate the write data address and send them to the read/write control unit. The read/write control unit is controlled by a state machine.[5]

Donghyuk Lee et al. in 2013 proposed Tiered- Latency DRAM (TL-DRAM), which achieves both low latency and low cost-per-bit. Commodity DRAMs connect many DRAM cells to each sense-amplifier through a wire called a bit line. Due to long lengths of bitlines which introduces larger parasitic capacitance they are dominant source of DRAM latency. In TL-DRAM, each long bit line is split into two shorter segments by an isolation transistor, allowing one segment to be accessed with the latency of a short-bitline DRAM without incurring high cost-per-bit. [6]

Satish reddy N et al. in 2014 pipelined DDR SDRAM controller which can perform multi-operations like read and write in different memory locations simultaneously. In normal SDRAM, the data transfer (either read or write) occurs from single location of memory, but in pipelined version of DDR SDRAM controller, it can do multiple operations from different memory locations simultaneously such as read from one location and write to another location The proposed controller supports data width of 64 bits, Burst Length of 4 and CAS (Column Address Strobe) latency of 2. In pipelined DDR SDRAM controller, for a single row and multiple columns, the data is transferred at a rate faster than normal SDRAM controller. This has shown improvement of 28.57% in performance of memory accessing. [7]

Wongyu Shin et al. in 2014 proposed new memory controller: NUAT which focuses on reducing memory access latency without any changes in existing DRAM architecture. NUAT exploits DRAM's intrinsic phenomenon: electric charge variation in DRAM cell capacitors. NUAT scales every memory access request and the request with the highest score is prioritized over the others. Author introduces two new concepts: Partitioned Bank Rotation (PBR) and PBR Page Mode (PPM). PBR is a method that draws information of access speed from refresh timing and position; the request having faster access speed gains higher score. PPM selects a better page mode between open- and close-page modes based on the information from previous

method ie PBR. NUAT can reduce the latency by 15% and improve total execution time by 7.7% in average without any modification of the existing DRAM structure. [8]

Seongil O et al. in 2014 proposed a novel DRAM micro architecture that can eliminate the need for any prediction when to close a row due to a lack of information on future requests. By decoupling the row buffers from the BLs through adding isolation transistors between them, start BL precharging right after activating a row while making the row buffer hold the active row. This early-precharge policy, enabled by row-buffer decoupling, allows for BL precharging off the critical path in most row-buffer misses, achieving the latency of the open-page policy for row-buffer hits and a latency nearly identical to the close-page policy for misses which is the better of both static policies. The simulation results show that row-buffer decoupling improves the instruction per cycle and MIPS2/W by 24% and 43%, respectively, compared to the open-page policy on average for nine main memory bandwidth demanding SPEC CPU2006 applications.[9]

Veena H K and A H Masthan Ali in 2015 proposed design that offers effective power utilization, reduced gate count, reduced area of chip and improved speed of system by reducing the gates. This controller consists of four architectural blocks namely main control, signal generation, data path and PLL. The maximum latency achieved for an access is for highest priority requestor in the worst-case 375ns the drawback of this controller is complex schematic with large number of buffers in the circuit which increases the amount of delay. [10]

Wongyu Shin et al. in 2016 proposed a technique to reduce DRAM latency without any modification in the existing DRAM structure with focus on an intrinsic phenomenon in DRAM: electric charge variation in DRAM cell capacitors and based on this phenomenon draw two key insights: DRAM row-access latency of a row is a function of the elapsed time from when the row was last refreshed and DRAM row-access latency of a row is also a function of the remaining time until the row is next refreshed. Based on these two insights, two mechanisms are presented to reduce DRAM latency: NUAT-1 and NUAT-2. This technique improves average 4.95 percent (up to 14.1 per cent) performance in multi-program environments. [11]

Amaresh.H et al. in 2017 designed and ASIC implementation of high speed DDR controller using cadence 45nm CMOS technology. The DDR SDRAM controller is capable of generating the signals for memory refresh, read and write functions and also memory initialization of DDR SDRAM to transfer the data using the designed controller. The three main modules in the architecture of controller are namely main control module to initialize the DDR SDRAM and to generate commands for memory based on input control signals, signal generation module to generate addresses and DDR SDRAM commands based on output of main control module and data path module. Data path module is designed using two delay flip-flops and a multiplexer. Same clock has been provided to two delay flip-flops and select line of 2:1 mux. Hence, by using single clock for all the blocks in the method the two data's will appear at the output at both rising and falling edges of the clock signal.[12]

III. STANDARD DDR SDRAM CONTROLLER

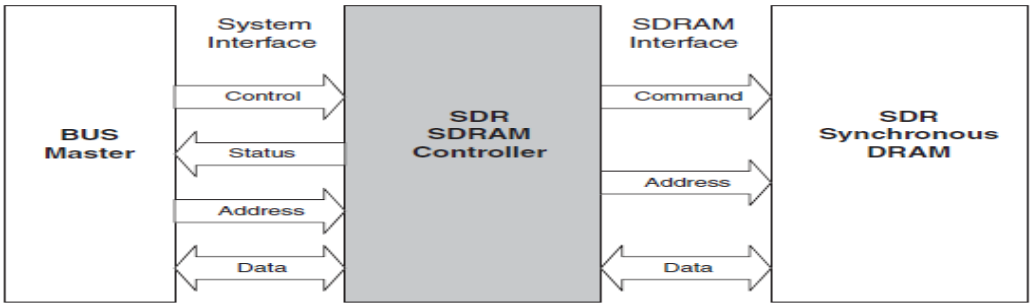

Fig 3 Architecture of DDR SDRAM Controller

Fig 3 the generalized block diagram for the memory element connected to the device via memory controller. Below shown is the simplified state diagram as per standard JESD79E

.

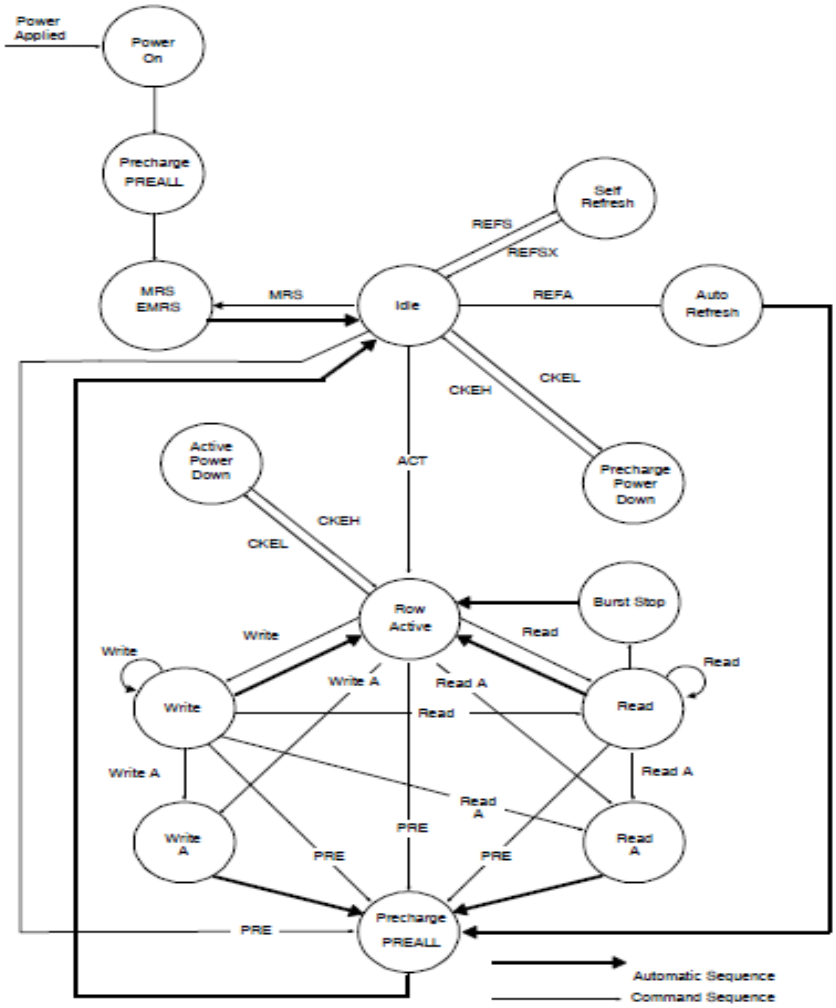

Fig 4 Simplified state diagram

PREALL = Precharge All Banks
CKEL = Enter Power Down
MRS = Mode Register Set
CKEH = Exit Power Down
EMRS = Extended Mode Register
Set ACT = Active
REFS = Enter Self Refresh
Write A = Write with Autoprecharge
REFSX = Exit Self Refresh
Read A = Read with Autoprecharge
REFA = Auto Refresh PRE = Precharge

IV. CONCLUSION

DRAM latency has caused a performance bottleneck in computing systems. With the advancement in technology scaling more and more cells are getting accumulated in DRAM memory, which is increasing the size of memory. Long bitlines are used to connect these cells to row buffer and sense amplifier. Long bitlines is one of the major sources of parasitic capacitance which contributes to DRAM latency. Thus the motivation to decrease DRAM latency lies in processor-memory interfaces which expose DRAM micro architecture to a memory controller so that the memory controller can track internal bank states and make the best scheduling decision for the pending requests to minimize their average service latency

REFERENCES

[1] Ikeda, H. and Inukai, H., 1999. High-speed DRAM architecture development. IEEE Journal of Solid-State Circuits, 34(5), pp.685-692.
[2] Heithecker, S. and Ernst, R., 2005, June. Traffic shaping for an FPGA based SDRAM controller with complex QoS requirements. In Design Automation Conference, 2005. Proceedings. 42nd (pp. 575-578). IEEE.
[3] Gulur, N., Manikantan, R., Govindarajan, R. and Mehendale, M., 2011, October. Row-buffer reorganization: Simultaneously improving performance and reducing energy in drams. In Parallel Architectures and Compilation Techniques (PACT), 2011 International Conference on (pp. 189-190). IEEE
[4] Kim, Y., Seshadri, V., Lee, D., Liu, J. and Mutlu, O., 2012. A case for exploiting subarray-level parallelism (SALP) in DRAM. ACM SIGARCH Computer Architecture News, 40(3), pp.368-379.
[5] Wang, L., Wang, J. and Zhang, Q., 2012, October. Design and implementation of DDR SDRAM controller based on FPGA in satellite navigation system. In Signal Processing (ICSP), 2012 IEEE 11th International Conference on (Vol. 1, pp. 456-460). IEEE.
[6] Lee, D., Kim, Y., Seshadri, V., Liu, J., Subramanian, L. and Mutlu, O., 2013, February. Tiered-latency DRAM: A low latency and low cost DRAM architecture. In High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on (pp. 615-626). IEEE.
[7] Reddy, N.S., Chokkakula, G., Devendra, B. and Sivasankaran, K., 2014, February. ASIC implementation of high speed pipelined DDR SDRAM controller. In Information Communication and Embedded Systems (ICICES), 2014 International Conference on (pp. 1-5). IEEE.
[8] Shin, W., Yang, J., Choi, J. and Kim, L.S., 2014, February. NUAT: A non-uniform access time memory controller. In High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on (pp. 464-475). IEEE.
[9] Seongil, O., Son, Y.H., Kim, N.S. and Ahn, J.H., 2014, June. Row-buffer decoupling: A case for low-latency DRAM microarchitecture. In Computer Architecture (ISCA), 2014 ACM/IEEE 41st International Symposium on (pp. 337-348). IEEE.
[10] Veena, H.K. and Ali, A.M., Design and Implementation of High Speed DDR SDRAM Controller on FPGA. July 2015, International Journal of Engineering Research & Technology (IJERT) ISSN, pp.2278-0181.
[11] Shin, W., Choi, J., Jang, J., Suh, J., Moon, Y., Kwon, Y. and Kim, L.S., 2016. DRAM-Latency Optimization Inspired by Relationship between Row-Access Time and Refresh Timing. IEEE Transactions on Computers, 65(10), pp.3027-3040
[12] Amaresh, H., Amaresha, S.K. and Yellampalli, S.S., 2017, August. ASIC implementation of high speed DDRSDRAM controller. In Smart Technologies For Smart Nation (SmartTechCon), 2017 International Conference On (pp. 754-757). IEEE.