

## **Address Alignment and Dynamic Mode Change Based SDRAM Memory Controller**

Abhishek Kumar  
*School of Electronics and Electrical Engineering*  
*Lovely Professional University*  
*Punjab, India*  
*abkvjti@gmail.com*

### **Abstract-**

*Computer system is increasingly limited by memory performance. The gap between processor and external memory increases 50% per year, a dynamic memory controller improves the communication speed between processor and memory. Here scheduling algorithm is presented for synchronous dynamic random access memory (SDRAM). The presented scheme reduces the SDRAM latency by 40%.*

**Keywords** -SDRAM, Memory controller, Address scheduler

### **1. Introduction**

The gap between core and memory is continuously increases, which slowdowns the computer performance. SDRAM is most important memory category of its DRAM family but it suffers with variable access time. Latency measures the speed of accessing the instruction/data from memory [1-5]. Pipelining is one of preferred mechanism to improve through put by parallel fetching. SDRAM is organized as multiple memory banks. Modern SDRAM is randomly accessed with three dimension bank address, row address and column address needed. Pipelined access to bank increases the system throughput as data is accessible from one bank other bank can be prepared at the same time. Accessing the row in memory bank need different latency[6-9]. Memory bank with high latency in different row cannot access sequentially, Different tow in one memory bank with low latency can pipelined. Memory is classified in array of memory cell can access independently. Row buffer is treated as cached memory which reduces access latency. When a row activated read and write command performed per cycle. Structure of the paper is as follows, section describes operating mode of SDRAM, architecture of SDRAM memory controller and its internal block diagram is presented in section3 finally concluded in section4.

### **2. Operating Mode of SDRAM memory**

Since distance between processor and memory continuously increases, accessing from DRAM requires more time. High speed memory is one of thr solution but requires a memory controller. SDRAM controller consists of sens amplifier built in each bank serve a s cache memory. The address of each bank is portioned into row and column. Bank possess array of memory cell access complete row. An SDRAM operate either in standby mode or active standby mode.

**2.1 Standby Mode**

SDRAM stays in ideal state in standby mode. Other memory state bank active, read and precharge operation required. As a result the latency is 8 cycle [10]. (Assumed activate, precharge and column access takes 3, 3 and 2 clock cycle respectively).

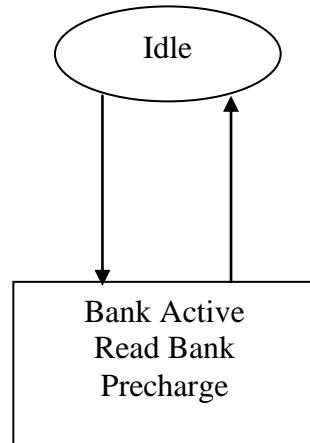


Figure 1 Standby mode read

**2.2 SDRAM active-standby mode**

Active standby mode of Synchronous DRAM stays row active state. Here data will access directly from sense amplifier[11-17]. If current accessed row matches with previous one result in hit shows shorter latency. Active standby mode is focused to maintain high ‘hit-rate’. If current accessed row is other than from last ‘row-miss’ occurs result in large latency. Since additional read operation requires to read data from sense amplifier. Figure2 presents the read of SDRAM active standby mode. Sense amplifier of row address of bank in Synchronous DRAM acts as cache. When the access address is hit shows low latency due to active state precharge state excluded. When a miss occurs the latency is 8 cycle because additional precharge and activate to be performed.

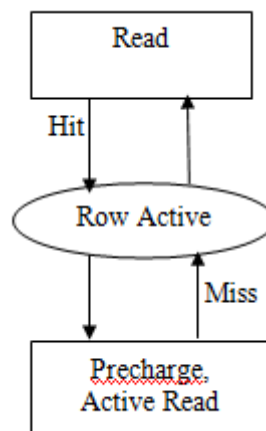


Figure2. Active stand by read operation in SDRAM

**3. SDRAM memory controller**

Objective of SDRAM memory controlled is to reduce latency to access the data from memory. It and maximize the row hit in standby mode and provide standby mode when result a miss. Figure3 presents the architecture of SDRAM memory controller. It contains (a) address alignment unit (b) hit-miss judgment unit, (c) dynamic mode change unit (d) command generation unit (5) memory mapping [4]. All arrived request at the interface is buffered in different queues. The request in each queue is mapped according to the priority request. Priority changed dynamically by reconfiguration ability of hit/miss judgment unit and dynamically mode change unit to change the mapping the request to queues. Address alignment unit contains the address of the arrive request; it contains the required information of the bank address, row address and column address [18-22]. This unit determines the information associated with number of column access in a row and number of row activation in a bank. According to selected algorithm the maximum number of column access in row and minimum number of row activation is considered. Hit /miss judgment unit determine whether arrived request belong to ‘standby’ or ‘active-stand-by mode’ the bank status. Mode of SDRAM changes dynamically from active standby or standby as per result of ‘hit-miss’ judgment. Command generation unit generates the SDRAM command and corresponding address to manage data transfer, corresponding to the scheduled memory access group.

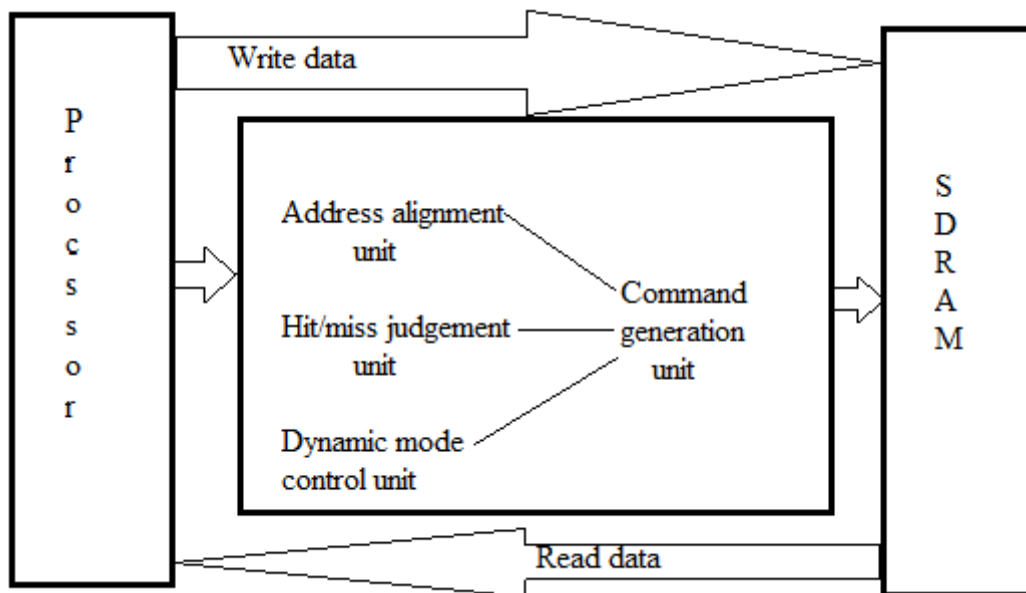


Figure 3. Architecture of SDRAM Memory controller

### 3.1 Address alignment unit

Address alignment unit helps to keep high hit rate of SDRAM. This unit aligns the bank, row and column address. It changes the address alignment statically to have high hit rate. The address alignment unit comprises of Tag and Index. Tag corresponds to bank address and row address while index corresponds to column address. This unit compares the address of arrived request with address of bank and provides the necessary information to hit/miss judgement unit. Three address alignments are shown in fig4. Row with highest ratio of column to row address will be activated. While pending reques

accumulate to other row. Most demanded row is selected first by freed up as soon as possible to progress the other request. It reduce the accessing time, row with little demand remains active, allow request of other row in that bank to make progress sooner. For alignment A the tag addresses A [8] corresponds to bank address B [1] and B [0]. For alignment B the tag address A [22] A [8] corresponds to bank address B [1] and B [0]. For C alignment the tag address A [22] A [21] corresponds to bank address B [1] and B [0].

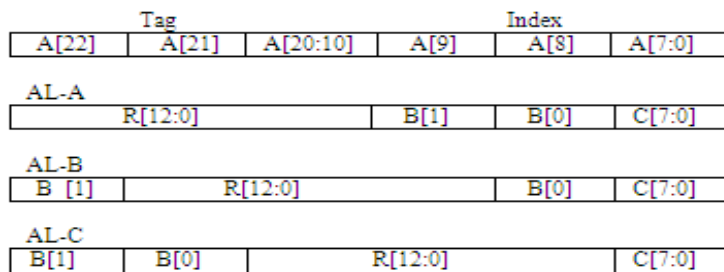


Figure 4 Address alignment of SDRAM [1, 3]

[12:0] R → Row address  
[7:0] C → Column address  
[1:0] B → Bank address

### 3.2 Hit/ Miss Judgement unit

This unit works on the type of arrived request; a request can be refreshing, hit or miss. The key objective is to enhance ‘hit-rate’ in ‘active stand-by’ mode and ‘miss-rate’ in standby mode. When ‘hit-rate’ reaches 100%, latency is minimum. 80% ‘hit-rate’ latency is almost equivalent to ‘active stand-by mode’. 20-80% ‘hit-rate’ mode changes dynamically to standby mode. ‘Hit rate’ less than 20% latency is equivalent to ‘stand-by’ mode [3].

### 3.3 Dynamic Mode Change unit

Hit Miss Judgment unit decide whether access result in ‘hit’ or ‘miss’. Mode of SDRAM shift mode of SDRAM from ‘active stand-by’ to ‘stand-by’ vice versa depending upon the result either hit or miss by dynamic mode control unit [1]. Dynamic-mode unit compare with threshold for changing the mode. A threshold of two successive miss is used. Sum of latency with 2 successive miss in active standby mode consist of 18 clock cycle (8+8+2). If two successive miss occurs dynamic mode change transfer mode ‘active stand-by’ to ‘stand-by’. To get back active standby mode only hit will required. The latency increases when mode of operation changes.

### 3.4 Command Generation unit

Command generator unit send necessary instruction and address to the SDRAM to manage the data transfer

### 3.5 Read Operation Mechanism

Figure. 4 shows the read operating mechanism of developed SDRAM controller. If bank is in active standby mode and read hit occurs only column access required latency reduces to 2 cycle, if miss

occurs precharge, row active and read operation. These operation increases latency to 8 cycles [2]. If several successive miss occurs in ‘active stand-by’ mode. Dynamic-mode unit tracks ‘stand-by’ mode and minimize latency if results a ‘miss’. If row access is ‘hit’ the bank gets in active and starts read operation. Dynamic mode unit changes threshold for mode changing. If successively 3 miss occurs dynamic mode control unit will change the bank status from ‘active stand-by’ to ‘stand-by’ state. The sum of latency with three successive miss in active standby mode consist of 18 cycles (8+8+2). This means the latency add up when mode changes; latency is more than 18 cycle the bank changes from active stand by to standby mode. To get back again read hit requires only 2 cycles.

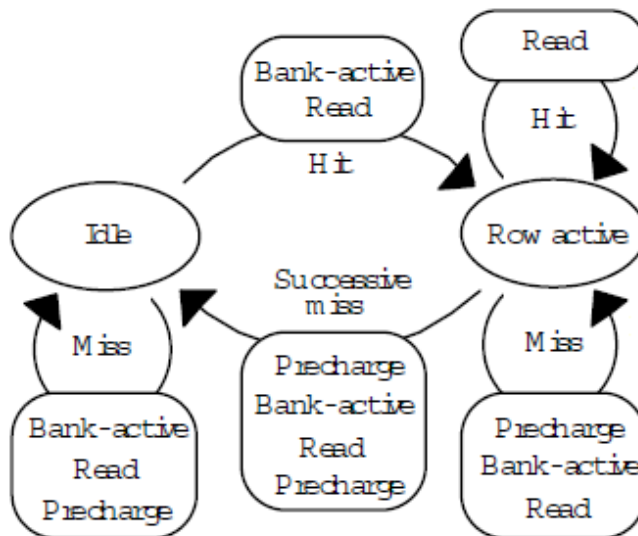


Figure 5 Read operation of proposed controller

**4. Conclusion**

Dynamic architecture of SDRAM memory controller is presented in this paper. To reduces the latency this architecture mainly focused on to enhance the hit rate on active standby mode and miss rate on standby mode. For hit rate of 50% in ‘active stand-by’ mode latency is larger than ‘stand-by’ mode. Latency increases when SDRAM changes its state. Address alignment unit aligns address such that minimum number of mode change takes place. The proposed architecture shows better result than conventional controller architecture.

**References**

- [1]Seije Miura and Satrou Akiyama, "A Memory Controller that reduces latency of cached SDRAM" IEIC Technical Report, pp 5250-5253, May 2005
- [2]S. Rixner et al., "Memory access scheduling". Proc. of the 27<sup>th</sup> annual international symposium on Computer architecture, vol 28, no. 2, pp 128-136, 2000.
- [3]Seiji Miura, Kazushige Ayukawa and Takao Watanabe, "A dynamic SDRAM-MODE-Control Scheme for Low-Power Systems with a 32 bit RISC CPU" Proc. International symposium on low power electronics and design, pp 358 – 363, 2001.
- [4]J.Tan and X.Fu, "Addressing hardware reliability challenges in general-purpose GPUs", Emerging Trends in Computer Science and Applied Computing, pp 649-705, 2017.
- [5]Park, Seong, and In-Cheol Park. "History-based memory mode prediction for improving memory performance." In Proc. International Symposium on Circuits and Systems, vol. 5, pp. V-V,2003.
- [6]Hussain, Tassadaq, Oscar Palomar, Osman S. Ünsal, Adrian Cristal, and Eduard Ayguadé, "Memory controller for vector processor." Journal of Signal Processing Systems, vol 90, no. 11, pp.1533-1549, 2018.
- [7]Ragab, Esraa, Mohamed A. Abd El Ghany, and Klaus Hofmann,"DDR2 Memory Controller for Multi-core Systems with AMBA AXI Interface." Proc. 30<sup>th</sup> International Conference on Microelectronics, pp. 224-227 2018.
- [8]Sudarshan, Chirag, Jan Lappas, Christian Weis, Deepak M. Mathew, Matthias Jung, and Norbert Wehn. "A Lean, Low Power, Low Latency DRAM Memory Controller for Transprecision Computing." Proc. International Conference on Embedded Computer Systems, pp. 429-441, 2019.
- [9]Sudarshan, Chirag, Jan Lappas, Christian Weis, Deepak M. Mathew, Matthias Jung, and Norbert Wehn. "A Lean, Low Power, Low Latency DRAM Memory Controller for Transprecision Computing." In International Conference on Embedded Computer Systems, pp. 429-441. Springer, Cham, 2019.
- [10] Matthias Jung and Norbert Wehn. "A Lean, Low Power, Low Latency DRAM Memory Controller for Transprecision Computing." Proc. 19<sup>th</sup> International Conference In Embedded Computer Systems: Architectures, Modeling, and Simulation, vol. 11733, p. 429,2019.
- [11] Hyoseung Kim, Dionisio De Niz, Björn Andersson, Mark Klein, Onur Mutlu, and Ragunathan Rajkumar. "Bounding and reducing memory interference in COTS-based multi-core systems." Real-

- Time Systems, vol 52, no. 3, pp 356-395, 2016.
- [12] Yonghui Li, Hrishikesh Salunkhe, Joao Bastos, Orlando Moreira, Benny Akesson, and Kees Goossens. "Mode-controlled data-flow modeling of real-time memory controllers." Proc. of 13th IEEE Symposium on Embedded Systems For Real-time Multimedia, pp. 1-10, 2015.
- [13] Hussain, Tassadaq. "Hmmc: A memory controller for heterogeneous multi-core system." Microprocessors and Microsystems, vol 39, no. 8, pp 752-766, 2015.
- [14] Vivek Seshadri, Thomas Mullins, Amirali Boroumand, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, and Todd C. Mowry. "Gather-scatter DRAM: in-DRAM address translation to improve the spatial locality of non-unit strided accesses." Proc, of the 48<sup>th</sup> International Symposium on Microarchitecture, pp. 267-280, 2015.
- [15] Meaney, Patrick J., Lawrence D. Curley, Glenn D. Gilda, Mark R. Hodges, Daniel J. Buerkle, Robert D. Siegl, and Roger K. Dong. "The IBM z13 memory subsystem for big data." IBM Journal of Research and Development, vol 59, no. 4/5, pp. 4-1, 2015.
- [16] Wooyoung Jang. "Unaligned Burst-Aware Memory Subsystem." IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol 27, no. 10, pp 2387-2400, 2019.
- [17] Graczyk, Rafał. "Vol. 41-Dependable control and measurement systems modeling for unmanned spacecraft." PhD diss., Warsaw U. of Tech., 2016.
- [18] Karthikeyan Palanisamy, "Memory Controllers." In Designing with Xilinx® FPGAs, pp. 49-64, 2017.
- [19] Xusheng Zhan, Yungang Bao, and Ninghui Sun, "DearDRAM: Discard Weak Rows for Reducing DRAM's Refresh Overhead." In Conference on Advanced Computer Architecture, pp. 109-124, 2018.
- [20] Hussain, Tassadaq, Amna Haider, Adrian Cristal, and Eduard Ayguadé. "Emvs: Embedded multi vector-core system." Journal of Systems Architecture, vol. 87, pp. 12-22, 2018.
- [21] Jin, Tian, Wenxin Li, and Xiangyu Hu. "A Tow-Level Buffered SDRAM Controller." Proc. Of 3rd International Conference on Information Science and Control Engineering, pp. 126-128, 2016.
- [22] Meng, Wang, Jiang Feng, and Xie Haolan. "Design and Implementation of DDR2 SDRAM Controller Interface on FPGA." Computer Measurement & Control, vol 12, no.34, 2016.