Design and Implementation of a DDR3-based Memory Controller

Pan Guoteng, Luo Li, Ou Guodong, Dou Qiang, Xie Lunguo

School of Computer Science, National University of Defense Technology, Changsha, Hunan, 410073, China
gtpan.hn@gmail.com

***Abstract*—Memory performance has become the major bottleneck to improve the overall performance of the computer system. DDR3 SDRAM is a new generation of memory technology standard introduced by JEDEC, support multibank in parallel and open-page technology. On the basis of in-depth study of DDR3 timing specification, design a DDR3-based memory controller. Memory access control module is the most key component of the memory controller. Using the stream testbench evaluate the performance, experimental results show that the memory controller of our design can correctly schedule memory access transaction, improve memory bandwidth.**

Keywords-DDR3 SDRAM, memory controller, memory access scheduling, memory bandwidth

I. INTRODUCTION

With the improvement of the semiconductor process level, microprocessor performance has improved at a rate of 60% per year, while the DRAM latency and bandwidth improve slowly, at only 7%-15%. Because of the growing performance gap between processor and memory, memory has become a major bottleneck to further improve the performance of computer systems [1]. In order to improve the memory bandwidth, reduce memory access latency, modern memory devices allows pipelined access, support multi-bank and open-page technology [2, 3], such as DDR SDRAM Series memory devices, RLDRAM memory [4]. In addition, there have been some new storage solutions: LRDIMM [5], eDRAM [6], MRAM [7], PCRAM [8], etc.. DDR series memory device still occupy the mainstream market.

DDR3 is the new generation of DDR memory technology standard that JEDEC (Joint Electronic Device Engineering Council) launched in 2007, compared to DDR2, DDR3 performs faster and consumes less power as it works. Fully exhibit DDR3 performances, efficiently and accurately access memory data is the important goal of memory controller design. Memory controller [9, 10] is the main channel for exchange data between processor and memory, its efficient design and implementation is the key to play memory performance.

Based on the timing specifications of DDR3 memory access operation, we proposed a greedy heuristic memory access scheduling algorithms according to the characteristics of DDR3 memory, and designed a memory controller supporting multi-bank concurrently.

II. SUMMARY OF DDR3

DDR3 uses the same Double Data Rate Transmission (DDR) technology as DDR2, but relative to the DDR2, DDR3 uses a higher frequency, lower operating voltage and more memory bank, thus providing a higher performance, lower power consumption, and greater memory capacity. DDR3 operating frequency starting from 400MHz up to 1066MHz, the corresponding data transfer rate from the 800MT/s to 2133MT/s. In order to support a faster clock frequency, DDR3 uses lower operating voltage (1.5v), power consumption is 30% lower than DDR2. Compared to the 4 logical banks that widely used in DDR2, DDR3 adopts 8 banks design, so memory capacity is greater. Because of 8bit prefetch technology (4bit in DDR2), DDR3 can provides twice the bandwidth of DDR2 at the same core frequency, so the core frequency is only 1/4 operating frequency. When core frequency is 200MHz, the data transfer rate up to 1600MT/s. In addition, in order to get a better signal integrity, DDR3 memory module adopted fly-by topology, reducing the number and length of the stub, but also led to a delay skew between DQS and clock signal, and therefore need the write leveling technology to adjust the controller internal skew.

DDR3 accelerate the concurrent processing of multiple memory transaction using multi-bank and row buffer technology. As DDR2, each memory access of DDR3 starts from active command, followed by the read and write commands. Active command gives the bank address and row address to be accessed, while the read and write commands give the first column address to be accessed in the burst access. DDR3 support Open-page technology, after a certain row in a memory bank is opened by active command, the open status of the row is maintained as long as it does not receive a pre-charge command, subsequent access to the rows can eliminate the active command time. DDR3 allows to choosing automatically pre-charge by programming, each time the end of the burst read and write access, SDRAM will automatically pre-charge the accessed row.

DDR3 specification [11] gives the specific timing requirements. Take the 533MHz memory as a example, the parameters setting of each request is: tCK = 1.875ns, tCCD = 4, tRC = 26, AL=0, BL=8, CL=6, WL=CWL=6, tWTR = max(4nCK, 7.5ns)=4. The relationship between the current memory request and the next is showed in Table 1.

978-0-7695-4923-1/12 $26.00 © 2012 IEEE
DOI 10.1109/ISDEA.2012.132

Table 1 Timing constrain relationship of two subsequent request

Relationship of two subsequent requests	Order of write and read	Minimum time interval (tCK)	Parameter (tCK)
same bank and same row/ different bank	RAR	tCCD	4
	WAR	RL + tCCD + 2 - WL	6
	RAW	WL+8+tWTR	18
	WAW	tCCD	4
same bank and different row	ANY	tRC	26

Based on the DDR3 timing specification, we designed a memory controller supporting open-page and multi-bank technology, implemented efficiently and right scheduling for multiple concurrency memory access using memory scheduling algorithm.

III. THE DESIGN OF MEMORY CONTROLLER

The functions that DDR3 memory controller needs to be done include: to receive and process the memory access requests, memory self-test operation, memory refresh and initialization operation, implementation of CPU register access path, read and write of all register in DDR3 controller.

The process of DDR3 controller for memory access requests is: receiving all kinds of requests and scheduling the memory access according to the priority of the request, generating the correct memory address control signals in accordance with the selected request and finishing the sending and accepting of memory data.

A. Overall Structure

The DDR3 memory controller that we designed is comprised of register control module, memory access control module, memory access datapath module and DDR3 SDRAM interface module, the structure is showed in as Figure 1.

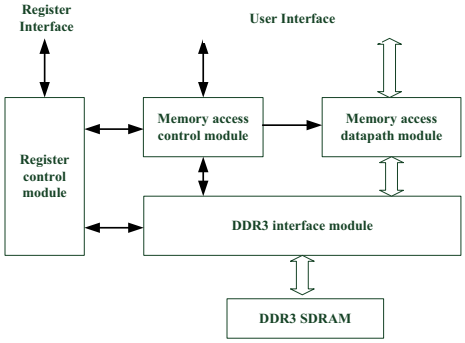

Figure 1. The structure design of DDR3 memory controller

Register control module consists of memory initialization sequence logic, self-test control logic, read and write control logic of all registers in DDR3 controller. Memory control module implements memory scheduling logic. Memory access datapath module is used to control the input and output of the memory data. DDR3 interface module implements the physical interface of memory, includes transmission of address and command, data transmission and generation of memory clock signal. Because memory control module is the key part of DDR3 controller, we will focus on it.

B. Structure and function of memory control module

Memory control module is the most critical part in the memory controller. As the memory chip has 8 banks, in order to solve the problem of address conflicts as well as to facilitate to implement multi-bank concurrent scheduling, memory access requests are divided into eight request groups in accordance with their bank address before scheduling, then schedule is carried out among the eight request groups according to a certain scheduling policy with starvation protection mechanism.

Memory control module receives memory read and write request, initialization sequence request from the register control module, refresh request and self-test request. Initialization sequence request only run once at system startup and can generate pre-charge, load mode register and refresh requests. Upon completion of initialization, automatically refresh request, self-test request initiated by CPU and memory access requests must be sent to the memory access scheduling module, and then memory access control signal generation module generate memory access control signals for selected request, including the address control signal, data strobe control signal, and read and write control signal. Memory access control signals are then sent to the DDR3 SDRAM interface to generate DIMM memory physical interface signals. The Figure 2 shows the structure of memory control module.

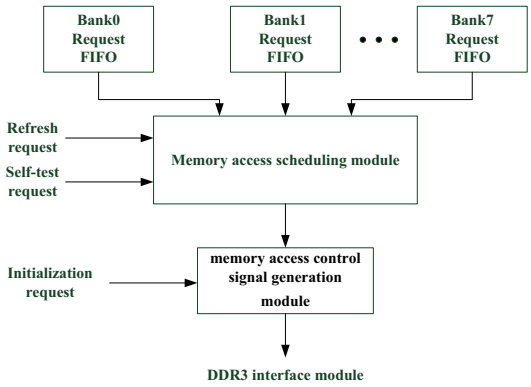

Figure 2. The structure design of memory control module

C. Memory access scheduling policy

The idea of memory access scheduling is that in order to fully use of memory bandwidth, reduce access latency, improve overall system performance, re-order the memory

access requests under the premise of meeting the memory timing requirement.

Assume i-1 requests have been scheduled among n requests, then there are $n-i+1$ candidate requests at the i time. For each candidate, we can find the scheduled time of the request, then we will use the scheduled time for each candidate as heuristic rules to choose the request with shortest candidate scheduled time as the scheduling request at the i time. We always choose the request with shortest scheduled time to schedule.

For ease of discussion, we assume that the first scheduled memory access request has been identified, and subsequent n memory access requests have arrived. Based on the above assumptions and principles, as well as DDR3 timing requirements, we propose a greedy heuristic memory access scheduling algorithm, as shown below.

greedy heuristic memory access scheduling algorithm

INPUT: the first scheduled memory access request R0; the memory access requests collection composed of n subsequent requests $M=\{R_1,\cdots,R_n\}$; starvation protection threshold F;
OUTPUT: memory access scheduled sequence $R_0^{'},R_1^{'},\cdots,R_n^{'}$, scheduled time $T_0,T_1,\cdots,T_n$ for each requests;

Algorithm Steps

1) Divide the memory requests in collection M into 8 groups $G_0,\cdots,G_7$ according to bank address, making the bank address of memory requests in group G_i is i;
2) Choose the request R^j with earliest arrival time in group G_j as the current candidate request;
3) Sequentially solving the minimum interval between the current candidate request and six times prior;
4) Find the scheduled time of current candidate request;
5) Prior consideration of the memory bank have not been accessed after F cycles, determine the request should be scheduled;
6) Remove the selected request from corresponding request group;
7) Update the beat counter for scheduling of each memory bank;
8) Repeat above steps 2-7, until all memory access scheduling is completed;
9) Return $R_0^{'},R_1^{'},\cdots,R_n^{'},T_0,T_1,\cdots,T_n$.

Memory access scheduling policy is implemented in memory access scheduling module, which determines the performance of the memory controller.

D. Design of memory access scheduling module

Memory access scheduling module consists of requests application logic of 8 memory bank, inter-bank requests scheduling logic, requests processing logic and memory access scheduling state machine. The requests application and processing logic of each bank are responsible for receiving read and write memory requests from the corresponding FIFO of memory bank, and then send the application to inter-bank requests scheduling logic and wait for the grant to access memory, once the application is successful, the memory access transaction can be scheduled for execution. The logic structure of memory access scheduling module is showed in Figure 3.

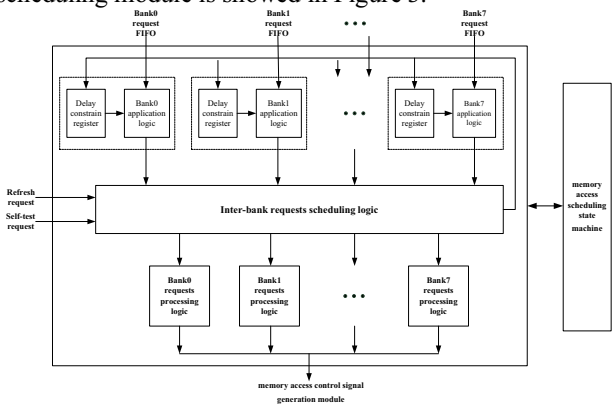

Figure 3. The logic structure of memory access scheduling module

The scheduled principles of memory access scheduling module are listed below:

1) After completion of initialization, there is only refresh request, self-test request initiated by CPU and memory access requests can be scheduled, according to the priority order of refresh request, self-test request and memory access request;

2) If exists a refresh request, the request can be scheduled after completion of current operation, and the process other self-test and memory access request;

3) If exists a self-test request, the request can be scheduled after completion of current operation, and the process other memory access request;

4) If no refresh and self-test request, the read and write requests of 8 memory bank FIFO will be scheduled using greedy heuristic principles; if several requests are ready, in order of priority of from bank0 to bank7 for scheduling.

In order to prevent the starvation of some memory bank request FIFO, we set a anti-starvation counter for each bank, recording the wait cycles from the last scheduled. If the counter is larger than a preset anti-starvation threshold F, the memory requests of the bank can be scheduled priority.

IV. PERFORMANCE EVALUATION

We use Verilog to implement the memory controller, and use STREAM to evaluate the performance. STREAM is a small synthetic benchmark, composed of Copy, Scale, Sum and Triad 4 sections, mainly used for the evaluation of computer memory bandwidth and computing speed. Copy section is mainly used to test memory bandwidth and have no relationship with computing speed. We implement a Verilog testbench which program behavior is similar as the STREAM Copy, called QSCV (Quasi Stream Copy in Verilog). The core part of the QSCV is that reading data from a continuous memory space and writing them to another continuous memory space, load / store instructions in program is implemented by our defined HostRd / HostWt task, sending only one read or write request to each memory block. Pre-fetch of the address can be realized by first sending a HostRd for the address to be accessed when HostRd Invoked. Since the instruction concept has not involved in QSCV, we defined the offset of pre-fetch address relative to current address as pre-fetched distance.

For simplicity, we assume that the data of each memory access is returned in the order of requesting at fixed delay. The testbench first issues four read requests, when the read data is returned, write the four data to another memory space, and then another four new read requests are issued, and so the cycle continues. While this simplification is different in the real environment, for example, each memory access latency may be different, the returned order of the data also may be different, the real memory access sequence is not simple periodically repetition, but does not affect the overall performance evaluation.

We use QSCV to test the memory bandwidth of the controller. Figure 4 shows the bandwidth under different pre-fetched distance, the X-axis represents the pre-fetched distance while the Y-axis represents the memory access bandwidth (MB/s).

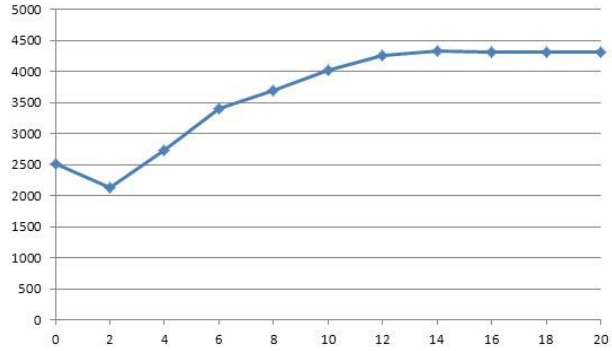

Figure 4. Memory bandwidth under different pre-fetched distance

Through experiments and analysis, we can draw the following conclusions:

1) Pre-fetched distance of QSCV has a great influence on the memory bandwidth;

2) Smaller the pre-fetched distance, more sensitive the memory bandwidth variation;

3) Memory bandwidth can reach to 4.5GB/s in the appropriate pre-fetched distance.

V. CONCLUSIONS

DDR3 memory supports the multi-bank concurrent and open-page technology, has higher performance compared to DDR2. In order to fully exhibit the advantages of DDR3 performance, efficient design and implementation of memory controller is essential. We have studied the DDR3 specification in-depth, under the premise of meeting memory timing constrain, a memory controller based on DDR3 is implemented in which the greedy heuristic memory access scheduling algorithm is adopted to discover the potential of the DDR3 memory structure. The experimental evaluation results show that the memory controller can guarantee the correct and efficient work of DDR3 SDRAM.

REFERENCES

[1] Bruce Jacob, Spencer Ng, David Wang. Memory Systems: Cache, DRAM, Disk. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2007.

[2] Dimitris Kaseridis, Jeffrey Stuecheli, Lizy Kurian John. Minimalist Open-Page: a DRAM page-mode scheduling policy for the many-core era. Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture. pp24-35, 2011

[3] Grannaes, M., Jahre, M., Natvig, L. Low-Cost Open-Page Prefetch Scheduling in Chip Multiprocessors. IEEE International Conference on Computer Design(ICCD 2008), pp390-396, 2008

[4] Micron. RLDRAM memory: Unparalleled Bandwidth and Low Latency[EB/OL]. http://www.micron.com/products/dram-modules/lrdimm [2012-08-20]

[5] Micron. LRDIMM: Rack Up Server Performance with Load-Reduced DIMMs [EB/OL]. http://www.micron.com/products/dram/rldram-memory[2012-08-20]

[6] Alejandro Valero, Julio Sahuquillo, Salvador Petit ,et al. An hybrid eDRAM/SRAM macrocell to implement first-level data caches. Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, pp213-221, 2009

[7] Xiaochen Guo, Engin Ipek, Tolga Soyata. Resistive computation: avoiding the power wall with low-leakage, STT-MRAM based computing. Proceedings of the 37th annual international symposium on Computer architecture, pp371-382, 2010

[8] Yuan Xie. Modeling, Architecture, and Applications for Emerging Memory Technologies. IEEE Design & Test. 28(1):44-51, 2011

[9] Y.Kim, D.Han, O.Mutlu and M.Harchol-Balter. ATLAS: A Scalable and High-Performance Scheduling Algorithm for Multiple Memory Controllers. IEEE 16th International Symposium on High Performance Computer Architecture,pp1-12, 2010

[10] Chang Joo Lee, O.Mutlu, Veynu Narasiman, Yale N.Patt. Prefetch-Aware DRAM Controllers. Proceedings of the 41st annual IEEE/ACM International Symposium on Microarchitecture, pp200-209, 2008

[11] JEDEC Solid State Technology Association. DDR3 SDRAM Specification(JESD79-3E)[S]. Arlington, VA,USA. 2010