Data Tunnel in DDRx Memory Controller

Jih-Ching Chiu
Department of Electrical Engineering,National Sun Yat-Sen University
chiujihc@ee.nsysu.edu.tw

Kai-Ming Yang
Department of Electrical Engineering,National Sun Yat-Sen University
d953010024@gmail.com

***Abstract*— With the growing imbalance between multi-core and memory, the data access from memory system becomes one of the primary bottlenecks, particularly much data movement. To reduce the growing gap, traditional approaches to reduce latency are mainly aimed at several factors such as the number of memory channels, the bandwidth of each channel, burst sizes, and scheduling requests dynamically, etc. For conventional DRAM, these approaches are reaching their practical limits. For instance, DMA popularly resolution still cannot eliminate this enormous gap due to the limited bus width and clock. This paper characterizes a performance examination of the DRAM-system architecture and the proposed memory controller can transfer data to destination and receive from source via data tunnel. In this design of data tunnel, while the lots of data are reading from source DRAM, the destination DRAM will capture data via data tunnel simultaneously. The results of simulation experiments show that the data tunnel mechanism enhances the fifty four percentages periods of time.**

***Index Terms*—DRAM, Memory access, Memory data transfer**

I. INTRODUCTION

DDR2 and DDR3 called DDRx in this paper [1] have the characteristics with high speed data rate, low power consumption and high density. Caused on this it is very popular on the applications of personal computer, working station and server especially on the digital electronic products and embedded multiprocessor systems for example video transformation box, digital camera and embedded SOC platforms. In multiprocessor systems the local memory may be constructed with SDRAM for getting a large memory spaces and the data duplication will be a big issue on large data transfer. According to above characteristics and widespread applications, we could design a memory controller with the standard AHB [2] (Advanced High-performance Bus) interface and implement the fast data switching function and configurable functions on it. To make it applicable to different timing of memory and achieve the rapid data switching approaches. Therefore we can not only reduce the developer's working load study on how to control the memory but also increase the applications of the digital product on the markets.

In this study, we propose the memory controller with internal fast data switching mechanisms via data tunnel and configurable functions. According to these ideas it can both control the different timing memory and switch the different bank address of data without increasing the burden on the work of AHB. Thus it will reduce the working load of AHB and increase the changing rate of data to achieve the demand of time-to-market. The proposed memory controller can support the DDR2 and DDR3 standards called DDRx in this paper.

Table 1. DDR2 standard pins

Port Name	Active state	I/O	Description
Clk, Clk#	N/A	Input	DDR2 Memory clock
CKE	HIGH	Input	DDR2 Memory clock enable signal
A0~A13	N/A	Input	Address lines
BA0~BA2	N/A	Input	Bank address lines
DQ0~DQ15	N/A	IN/OUT	Data bus
DM	N/A	Input	Mask signal
DQS,DQS#,RDQS,RDQS#	N/A	IN/OUT	Data strobe
CS#	LOW	Input	Chip select
CAS#	LOW	Input	Column address strobe
RAS#	LOW	Input	Row address strobe
WE#	LOW	Input	Write enable
ODT	HIGH	Input	On-die termination

II. Previous work and background

A. Overview the SDRAM

SDRAM is one of the DRAM (Dynamic Random Access Memory), its different place lies in synchronous. By synchronized with the memory clock and CPU clock it could transfer the data more

Table 2. DDR2 Command

Function	CKE		CS#	RAS#	CAS#	WE#	BA2~BA0	An~A11	A10	A9~A0
	Previous	Current								
Lode mode	H	H	L	L	L	L	BA	OP code		
Bank Activate	H	H	L	L	H	H	BA	Row address		
Refresh	H	H	L	L	L	H	X	X	X	X
Self refresh entry	H	L	L	L	L	H	X	X	X	X
Self refresh exit	L	H	H	X	X	X	X	X	X	X
			L	H	H	H				
Single bank precharge	H	H	L	L	H	L	BA	X	L	X
All bank precharge	H	H	L	L	H	L	X	X	H	X
Write	H	H	L	H	L	L	BA	Column	L	Column
Write with auto precharge	H	H	L	H	L	L	BA	Column	H	Column
Read	H	H	L	H	L	H	BA	Column	L	Column
Read with auto precharge	H	H	L	H	L	H	BA	Column	H	Column
No operation	H	X	L	H	H	H	X	X	X	X

quickly. The characteristic of dynamic is the data stored in memory will lost because there has not provided enough power supply due to the shutdown of the system or computer crash.

In the aspect of transfer data, SDRAM is not like the SRAM, transfer data in one cycle. Cause the architecture of the hardware; we must control the SDRAM by several steps which called command. We could use the commands to control the SDRAM like ACTIVATE, READ, WRITE, PRECHARGE, etc.

Although there has some delay times in accessing the data, but in another way, if we focus on the performance and the lower cost by using the SDRAM as the main memory, the more steps in access will not cause a lot of disadvantages.

B. Overview the DDR2

The standard of DDR2 (Double Data Rate 2) is made by JEDEC [1] (Joint Electronic Device Engineering Council). Compared with the preceding generation of DDR memory, not only enhance the bandwidth but also promote the system's performance and the effect, especially facilitate the system design.

The standard pins of DDR2 as Table 1 shows [3-6]. The clock of DDR2 memory is provided by

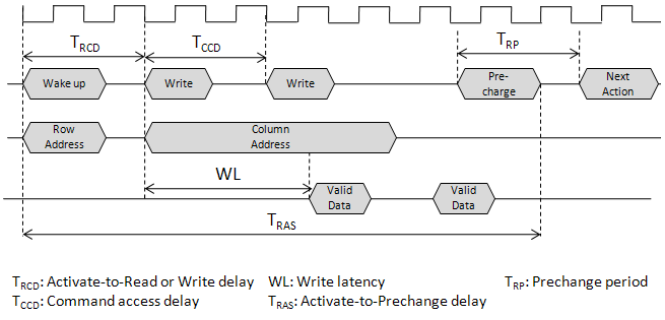

Figure 1. Write timing operation

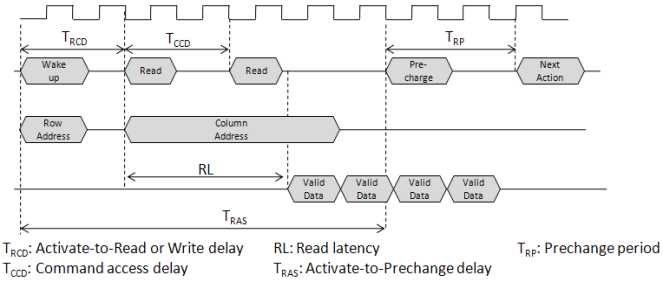

Figure. 2. Read timing operation

controller and CKE decided whether the clock effective or not. A0~A13 are the address lines.BA0~BA2 are the bank address of the memory. DQ0~DQ15 are the data bus and DQS are the strobe signals to access the data. DQ and DQS are bidirectional. Controller will sent out the DQ and DQS into memory when store data. And in another ways controller will receive these two signals if we read the data from memory. DM signal is decided whether the data is available or not.

C. DDR2 Command

All DDR2 memory commands are controlled by the five signals, CS#, RAS#, CAS#, WE#, CKE, as the Table 2 shows.

Lode mode (LM): Using the LM instruction to setup the mode registers of the DDR2 SDRAM by changing the contents of A0 ~ An. Especially notice that the instruction must be work under the situation of bank idle and resend the instruction after tMRD time.

Bank Activate: The instruction is used to open one of the row in the bank in order to access data. The bank address is used to select bank and the row of the bank is selected by A0~An. As the row is activated we could access the data until the PRECHARGE command set to the bank. If we want to activate different row in the same bank, we must sent out the PRECHARGE command first.

REFRESH: We must send the REFRESH

Table 3. Compare with different controller

	Auto initialization	Timing parameters	User data interface	Fast data switching
TI	Yes	Dynamic programmable	EDMA bus	No
Lattice	Yes	Dynamic programmable	Standard data bus of lattice	No
Prime Cell ARM	No	Dynamic programmable	AXI bus for DDR2 SDRAM APB for program the device	No
Beyound	Yes	Dynamic programmable	AHB for DDR2 SDRAM APB for program the device	No
DDC	Yes	Compile time configurable	Standard data bus of DDC	No
Microtronix	Yes	Compile time configurable	Standard data bus of microtronix	No
Our design	Yes	Compile time configurable	AHB for DDR2 SDRAM AHB for program the device	Yes

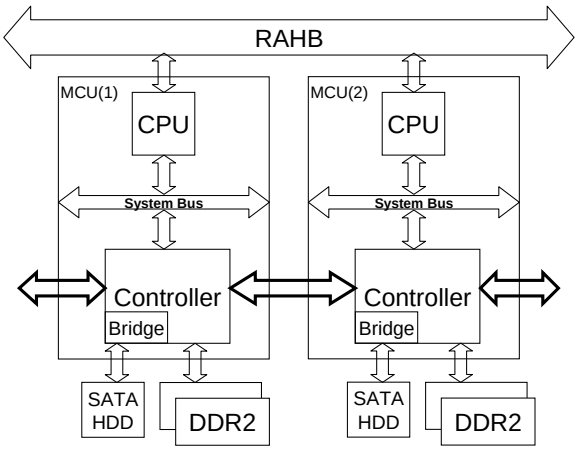

Figure 3. Data access flow

command to all banks in idle mode and the REFRESH command is not continues. After entering REFRESH mode we would return to idle mode in some periods of time, so that we must send out the REFRESH command once in a while.

PRECHARGE: The instruction is used to deactivate the activate row of bank or all activate row of the bank by select the A10 bit. After PRECHARGE one of the bank it will go into the idle mode and if we want to access the bank we must activate it first.

Write: The write command is used to store data in one active row and the bank address is chose what bank we want to access. The start address is selected by A0~An. An is the MSB of the column. A10 is set to see if auto precharge will work or not. If A10 is high it will precharge the active row after writing and if A10 is low the active row would still maintain active to be access.

Read: The read command is used to read data from one active row and the bank address is selected what bank we want to access. The same with write command A0~An is chose what is the start address. An is the MSB of the column. A10 is set to see if auto precharge will work or not. If A10 is high it will precharge the active row after reading and if A10 is low the active row would still maintain active to be access.

No operation: The NOP command is used to prevent the unnecessary command to be carried out in the idle mode or wait states.

D. DDR2 Registers

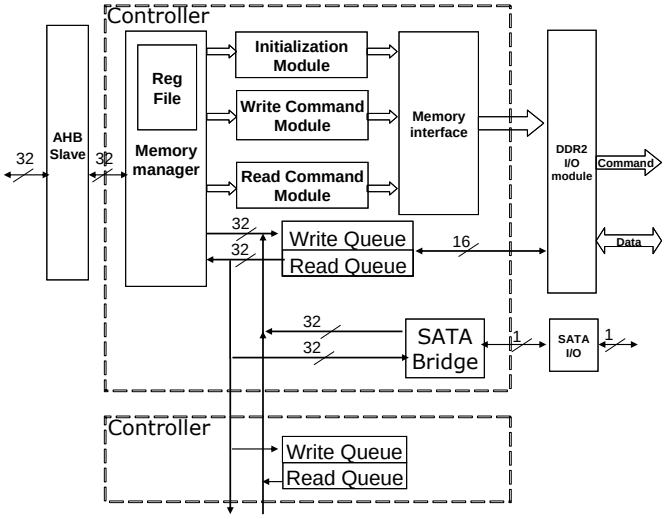

Figure 4. Architecture of controller

DDR2 memory has four registers (MRS, EMRS1, EMRS2, EMRS3) and regulate the memory by set the A0~A13 and BA0~BA2. However by present DDR2 specification EMRS2 and EMRS3 are reservations not any function in the present stage. BA0~BA2 are set to choose what register we want and the bits of A0~A13 have the specific use to setup the registers.

E. Access Timing

If we want to write the data into memory as the Fig 1 shows, we should send out the Activate command with row address and bank address. Then we sent out the write command with column address follow by the previous one. After writing data into memory we should sent the PRECHARGE command as the end of the write.

Then if we want to read data form the memory we could sent out the command like Fig 2. Just like the same with write period, we should send out the Activate command with row address and bank address first then we could send out the read com-

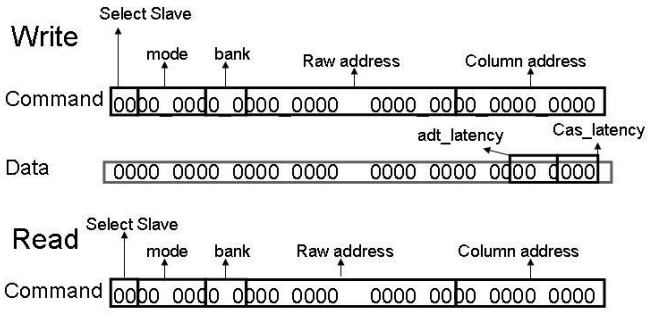

Figure 5. Instruction format

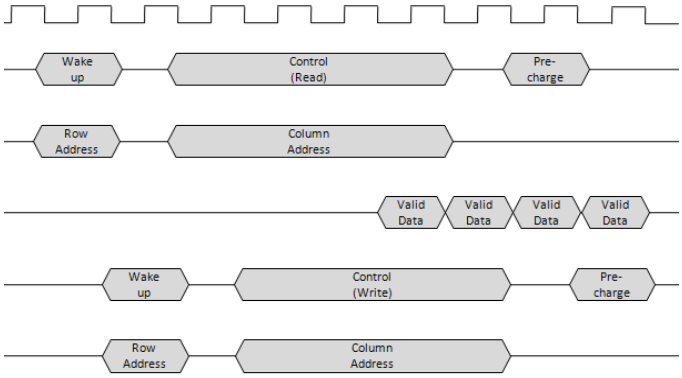

Figure 6. Data switching timing

mand with column address. After the period of time we must send out the PRECHARGE command if we want to stop the reading procedure.

F. Compare the characteristics with different controller

Table 3 shows the characteristics of different controller made by several manufacturers [7-12]. The timing parameters of these controllers could separate into two types, dynamic programmable and compile time configurable. Dynamic programmable is set every parameter in advance and enable it or not by setting the device. Compile time configurable is set the parameter by user, according to using which timing of memory.

In above DDR2 controllers, they only take the DDR2 SDRAM control signals for the bus masters, but not support the memory-to-memory fast transfer functions, which will be used in embedded multiprocessor systems. In this paper, we design and implement the DDR2 controller with data switching, which can support the fast data switch function to save lots of time on switching data without through the AHB. Especially, it didn't cost lots of spaces to buffer the data but offered the fast data switching utility.

III. Controller Architecture

A. The goal of the architecture

This section will descript the whole architecture with memory controller to establish a memory controller with fast data switching algorithm on the platform of the AMBA architecture. In traditional the memory controller accessing data through AHB will waste a lot of memory spaces and the read times on reading data from one address to another new address continually. So that this paper proposes an internal data switching mechanism, show as Fig 3. Establishes a transmission mechanism in two controllers and discard the buffer space when access data. It could reduce the total areas of the controller. If the user issues the DMA command, it will open the data switching bus and exchange the data from different memory. Especially it will not occupy the AHB transmission path and release AHB the right of use to achieve the goal of data transfer.

B. The architecture of the controller

In this section the editor will introduce the architecture of the controller, see as Fig 4. The decode block is used to analyze the address as command and will sent out the enable signal according to the command to active the initial, write and read block. Every block will create the corresponding command signal to DDR2 module. When the user issue the read command it will sent back the data by read queue block and it will see whether the data is requested by CPU or internal DMA transfer. If the user requests the data from the CPU it will transfer it by AHB. And if the data is requested by the other controller it will sent it to the write block of the other controller.

C. Instruction format

The Instruction format of the decode block is show as Fig 5. A0~A9 are the column address, A10~A22 are the row address, A23~A24 are the bank address, A25~A29 are special designed for special applications named mode. And the last two addresses are set to select slave of the controller. If the mode is selected to issue the initial function the first six bits of the data are set to decide what the cas_latency and adt_latency are. The user can fix the number to achieve the different timing of DDR2.

The mode of the command is shows as Table 4. The paper proposed the several kinds of applications. According to the different mode select the user could do the single read write or multiple read write. Even if the application of data switching the user could only set the start address and the destination address before sending the DMA start signal.

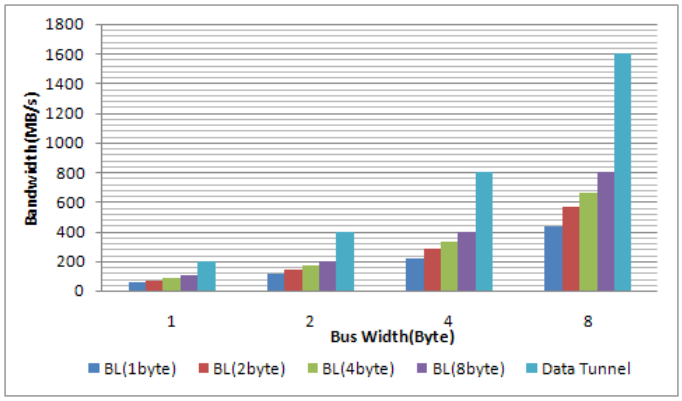

Figure 7. Performance Evaluation

Then the data will transfer by the internal bus not by the AHB. So that it will save a lot of times in transmission data.

D. Data switching timing

For the DMA command the most important part is to align the data with read and write periods. With the read period data will get after the time of read latency. But the write latency is not the same with read latency. The mechanism of switching data shows as Fig 6. Following this data switching timing mechanism the DMA command would access the data without any register buffers. It could save a lot of areas in the memory controller.

IV. Simulation result

This chapter will descript the result of the performance by comparing with DMA transfer and traditional access, the simulation models are Micron MT47H32M16 [13-15].

Transmitting the first data with AHB would take fourteen clocks in reading and ten clocks in writing in traditional transfer shows as Fig 7. The first data of the transmission would just take eleven clocks only such that it will switch data automatically without any other redundant clocks. This idea save thirteen clocks on switching data and reduce the working load on AHB. Especially the controller will not construct an unnecessary register buffer to store the template data for next writing.

The memory controller with data switching is implemented by using Verilog language and synthesized by Synopsys Design Compiler in a 0.18um technology see as Fig 8. The total area and critical path are 214076.10um2 and 3.82ns latency. The whole controller can work on the environment of ModelSim 5.7d and the working rate can achieve 192MHZ.

V. Conclusion

In this paper the novel memory controller with data switching is proposed. It discards a large number of temporary spaces to reduce the total areas. And with the DMA command of the architecture user can access the data more quickly without crossing AHB. So that it not only reduce the traffic load of the AHB but also the access time when switching data. By the simulation result, it improves 54% of time to access data from one address to another.

REFERENCE

[1] JEDEC DDR2 SDRAM Specification.

[2] AMBA Specification (Rev 2.0).

[3] Shuang-yan Chen, “An Innovative Design of the DDR/DDR2 SDRAM Compatible Controller” Volume 1, 24-27 Oct. 2005 Page(s):62 - 66. ICASIC.2005

[4] Fu-Min Huang, “An Aggressive Memory controller with Memory Access Scheduling and Bank Precharge Strategies” Dept. Elect. Eng. NCKU, Tainan, Taiwan, ROC. 2004

[5] Ning-Yaun Ker, “A low-Power SDRAM Controller on an 8-bit RISC CPU” Dept. Elect. Eng. NCKU, Tainan, Taiwan, ROC. 2002

[6] Datta, A.; Singhal, V, “Formal verification of a public-domain DDR2 controller design” 4-8 Jan. 2008 Page(s):475 – 480, VLSI.2008

[7] TI DDR2 SDRAM Controller. Available: http://www.ti.com/

[8] Lattice DDR2 SDRAM Controller. Available: http://www.latticesemi.com

[9] Primecell DDR2 Dynamic Memory Controller. Available: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0418d/index.html

[10] Beyond DDR2/DDR SDRAM Memory Controller. Available : http://www.beyondsemi.com/page/products/interface_cores/beyond_ddr_sdram_memory_controller

[11] DDC DDR/DDR2 Memory Controller. Available: http://www.digidescorp.com/products/IP/DataSheets/DDC_DDR_datasheet.pdf

[12] HyperDrive Multi-port DDR2 Memory Controller IP. Available : http://www.altera.com.cn/products/ip/iup/memory/m-mtx-multiport-hyperdrive-sdram.html

[13] Micron 512Mb SDRAM Data Sheet “MT47H32M16”

[14] Micron 512Mb SDRAM Simulation Model “MT47H32M16”

[15] Micron 256MB DDR2 SDRAM UDIMM Data Sheet “MT4HTF3264A”