ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 2, February 2015

Reduction of Congestion in 3d NOC Using PACR

M.Sabari Raja
PG Scholar,VLSI DESIGN
Karpagam College of Engineering,
Coimbatore, India.

Mr.S.Shivkumar
Associate Professor, ECE Department
Karpagam College of Engineering,
Coimbatore, India.

Abstract—Network on chip (NOC) systems can perform better performance than SOC for chip microprocessor systems.When more than one processor transmits its data to a same node there is a possible of occurrence of congestion in the path of transmission. In order to avoid this condition, Path Congestion Aware Adaptive Routing Algorithm is being used here.FSM (Finite State Machine) memory controller will switch data to the other nodes or processor where the transmission path is free of traffic or congestion. The processors which are frequently used will be placed in TRB(Thread Row Buffer) and hence we can directly fetch the processor from TRB, instead of going to memory.

Index terms—Network on chip, Path Congestion Aware Adaptive Routing, Finite State Machine, Thread Row Buffer.

I. INTRODUCTION

Modern Congestion Aware NOC memories have three dimensional structures comprising of bank, row and column dimensions with dedicated row buffers. In order to access an entire row at a time which operates independently over the other banks each Congestion AwareNOC memory bank has been provided with an array of memory cells. If a row is accessed from Congestion Aware NOC memory bank then entire row is transferred to corresponding bank row buffer. After that if a row is active in row buffer, then read or writes operation can be performed. The row buffer operates as a cache in order to reduce the latency of consequent accesses to the corresponding row.

Then row precharge or bank refresh operation has to be performed to written back the row to memory array for subsequent row activation. During the row precharge or activation operation in a bank, read or write operation cannot be performed over the same bank. While accessing Congestion Aware NOC memory array following situations may arise,

1) Row Closed: This situation arise when the requested row is not available in the row buffer, so it needs to issue row activation command to open a row from the row buffer, then read or write command is to be followed .This results in memory bank access latency of T_{RCD+} T_{CL} (row to column command delay + column access latency) .Thus here both row access and column access operations are required.

2) Row Conflict: This situation arise when the requested row is different from the one available in the row buffer, so it needs to issue precharge command to written back the contents of row buffer to the memory array. After that requested row can be accessed through activate and read or write commands. It leads to the highest memory bank access latency of T_{RP+} $T_{RCD+}T_{CL}$(row precharge + row to column command delay + column access latency).

3) Row Hit: This situation arise when the requested row is available in the row buffer, so it needs to issue only read or write command .This results in the lowest memory bank access latency T_{CL} (column

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 2, February 2015

access latency) which is the time between accessing the column of row buffer and data come back by Congestion Aware NOC.

II. Memory Controller

In Congestion Aware NOC, memory controller acts as a mediator between the processor and Congestion Aware NOC. The controller has the capability of translating memory requests into sequences of memory access commands. This memory controller comprises of read buffer, write buffer,request buffer and Congestion Aware NOC memory access scheduler.

Congestion Aware NOC memory access scheduler comprises of logical priority queue with a bank scheduler for every bank in memory array. This memory access bank scheduler has a role of selecting the pending request which has the highest priority comparing to all other current requests. Then sequence of command is issued to read or write requests to or from memory array.

Request Buffer is common for all memory banks or it can be dedicated to each memory bank independently to reorder the requests from processor. This buffer holds the state of each request to memory array. In this requests to memory, arrays are reordered on the basis of FR-FCFS (First Ready – First Come First Serve) algorithm which gives priority to ready column access (read and write commands) over ready row access (precharge and activation commands).As FR-FCFS algorithm always prioritizes accesses to active rows it maximizes row buffer hit rate and in turn improves the overall memory throughput.

Memory access channel scheduler checks the bank's ready commands initially and then provides the command with the highest priority. If a command becomes ready then bank scheduler sends the command to the channel scheduler. This channel scheduler always acknowledges the corresponding bank scheduler as soon as the command is issued.Thus the memory bank access scheduler has a role of updating the current bank state machine and memory channel access scheduler and also inspects the state of address and data bus to ensure that the channel scheduling conflicts are not violated.

III. Thread Row Buffers

Congestion Aware NOC memory arrays have dedicated row buffers for all banks to access memory in a parallel manner and to service the request from multi processor chips simultaneously. But row buffers are required to update every time if a different row is accessed for read or write operation. It leads to the consumption of more time and energy, which results in reduced row locality and lower row hit rate. Hence significant power and energy become lost while replacing the data from row buffer to memory array.

In order to improve the overall performance of memory system, thread row buffers (TRB) are included for Congestion Aware NOC memory array in addition to row buffers. These TRBs have the capability of maintaining an active row per thread or processor to avoid an alternate access to limited number of rows in the bank row buffer which thereby significantly improves the efficiency of Congestion Aware NOC .

A) ExistingTRB

In existing technique, the number of TRBs are assumed to be same as the number of threads or processors sharing Congestion Aware NOC memory array. This leads to higher power consumption and more area overhead, but TRBs are focused on increasing row hit rate to improve the overall memory throughput by means of reusing the same row which consists of similar data. In addition, it reduces the interference of memory demanding request over request with more limited use of memory with dedicated row buffers.

Thus implementation of TRB requires additional storage in each bank of memory array, and it needs respective multiplexers to access from corresponding thread row buffer. To identify whether precharge and activation required for which thread

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 2, February 2015

row buffer, memory controllers have tag of active row and also it includes an address of thread row buffer active rows to avoid if a row was loaded in one TRB the same row could not be allowed to load in other thread row buffer.

B) Proposed [1:2] TRB

To reduce power and area, thread row buffers are optimized on [1:2] ratio for on-chip cores. In this proposed method,thread row buffers are implemented on the basis of total number of thread row buffers have to be equal to half of the number of processors or threads sharing the memory array.

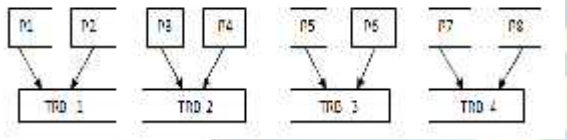

Fig 1: [1:2] TRB

This way of implementing TRB might reduce the power consumption and area overheadby thread row buffers for maintaining multiple active rows.

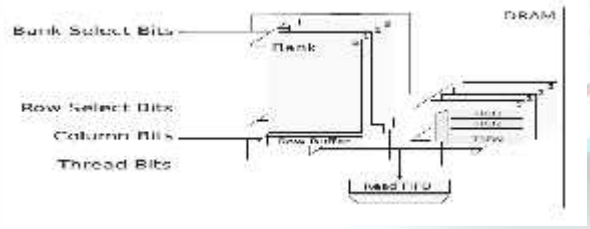

Fig 2: Structure of a single DRAM

In this method, we assumed that eight processors share common Congestion Aware NOC system. In order to access frequently accessed data of each thread or processor, TRB's are incorporated in Congestion Aware NOC memory array which uses the technique of reusing the row with same information. It is also included that last row accessed by each thread or processor is always loaded in TRB.

IV. PROPOSED FSM MEMORY CONTROLLER

The existing register-based traditional memory controller comprises of data bus arbiters, bank arbiters, request buffers and serial buses which needs higher implementation cost and increased implementation complexity, because of complex logical calculation and synchronous clocking. On further, it leads to higher critical path delay and larger area overhead.

To overcome the drawback of area overhead of traditional memory controller, an FSM based transition approach is implemented which uses only combinational approach. Based on this approach pre-defined instruction set is implemented. This enhances high speed computation with low power consumption and low area overhead.

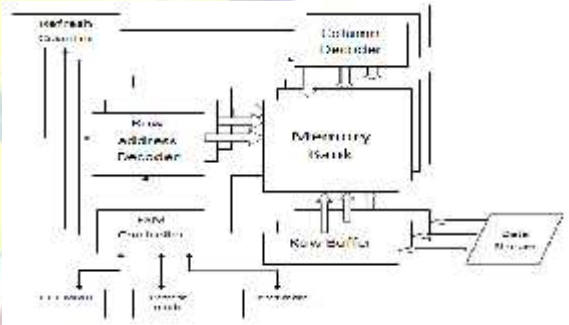

Fig 3: FSM Memory Controller

A) FSM State Diagram: To design state machine based memory controller, state transition diagram is constructed based on relationship between present state and next state of the system. In this diagram, nodes represent the states and arcs represent the transitions from one state to next state of the system. Nodes without outputs are known as accepting states.

This Finite State Machine based memory controller performs address decoding, refresh and reset memory array using preloaded instructions. In addition to this FSM controllers are error prone as it is based on the state transition approach and it also checks the availability of bank in a parallel manner in order to service the request issued by multiple thread or processor simultaneously.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 2, February 2015

B) Priority Calculation: In case of Chip Multi Processor (CMP) system, more number of processors might share Congestion Aware NOC memory array in a concurrent manner. To service the request issued by a processor or thread to Congestion Aware NOC memory array, priority has to be given for the respective processor. This priority assignment is based on the processor which does not issue any request is provided with the highest priority. But priority of processor which issues more number of request is assumed to be zero. The priority of processor gets decremented after completion of servicing the request. In terms of multi threaded application, requests from a single thread has to be scheduled on the basis of service partitioning algorithm. This technique performs scheduling on the basis of servicing the request from two or more threads in an alternate manner simultaneously.

V. CONCLUSION

In this work, state transition based Finite State Machine memory controller is designed with (1:2) TRB ratio for on-chip cores which optimize area and power without interfacing buses and registers comparing to existing traditional memory controller. Hence FSM controller has lower logical elements with higher speed computation and reduced implementation cost. As ratio of TRB is reduced with half of number of processors or thread comparing to existing work, this may reduce area overhead to a larger extent. Thus with TRB's power consumption for row access operation like precharge and activation is reduced to a greater extent with reduced area overhead. As a result the combination of FSM memory controller with TRB's achieved remarkable reduction in both area and power with improved memory throughput.

VI. RESULTS

The proposed approach which comprises of 4 TRB technique performs far better then 8 TRB. The results for 4 TRB technique has more then 50% of power optimization, similarly in case of area optimization it implies the better results. The respective power analysis flow summary for 4 TRB and 8 TRB techniques is given as a chart in the figure 5.

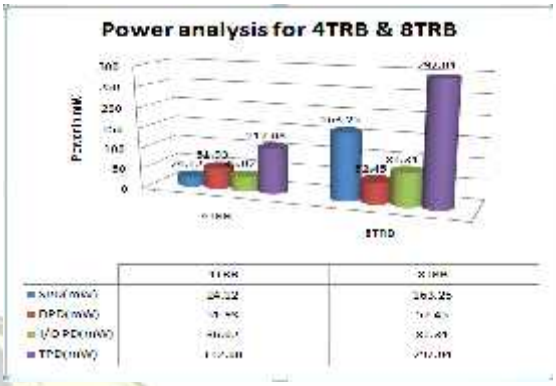

Fig 5: Power analysis for 4 TRB and 8 TRB

REFERENCES

[1]. En-Jui Chang, *Hsien-Kai Hsin, Shu-Yen Lin, and An-Yeu (Andy) Wu,* "Path Congestion-Aware Adaptive Routing with a Contention Prediction Scheme for Network-on-Chip Systems",vol.33.no.1,JANUARY 2014.

[2]. Akram Ben Ahmed," On the Design of a 3D Network-on-Chip for Many-core SoC", University of Aizu,FEB 2014.

[3]. R. Marculescu, U. Y. Ogras, L.-S.Peh, N. E. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: System, microarchitecture, and circuit perspectives, *"IEEE Trans. Computer-Aided Design Integr. Circuits Syst., vol. 28, no. 1*,pp. 3–21, Jan. 2009.

[4]. P. Gratz, B. Grot, and S. W. Keckler, "Regional Congestion Aware ness for load balance in networks-on-chip," in *Proc. Int. Symp.High-Perform.Comput.Arch.*, pp. 203–214, 2008.

[5]. G. Ascia, V. Catania, and M. Palesi, "Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip," *IEEE Trans. Comput.*, vol. 57, no. 6, pp. 809–820, Jun. 2008.

[6]. G.Ascia, V. Catania, and M. Palesi, "Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip," *IEEE Trans. Comput.*, vol. 57, no. 6, pp. 809–820, Jun. 2008.

[7]. J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, and C. R. Das, "A low latency router supporting adaptivity for on-chip interconnects," in *Proc. Design Autom.Conf.*, 2005, pp. 559–564.

[8]. U. Y. Ogras, P. Bogdan, and R. Marculescu, "An analytical approach for networkon-chip performance analysis," *IEEE Trans. Computer-Aided Design Integr. CircuitsSyst.*, vol. 29, no. 12, pp. 2001–2013, Dec. 2010.