STT-MRAM BASED NoC BUFFER DESIGN

A Thesis

by

NIKHIL VIKRAM KULKARNI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2012

Major Subject: Computer Engineering

STT-MRAM BASED NoC BUFFER DESIGN

A Thesis

by

NIKHIL VIKRAM KULKARNI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Eun Jung Kim
Committee Members,	Rabi N. Mahapatra
	Paul V. Gratz
Head of Department,	Duncan M.H. Walker

August 2012

Major Subject: Computer Engineering

ABSTRACT

STT-MRAM Based NoC Buffer Design. (August 2012)

Nikhil Vikram Kulkarni, B.E., People's Education Society School of Engineering, Bangalore

Chair of Advisory Committee: Dr. Eun Jung Kim

As Chip Multiprocessor (CMP) design moves toward many-core architectures, communication delay in Network-on-Chip (NoC) is a major bottleneck in CMP design. An emerging non-volatile memory – STT MRAM (Spin-Torque Transfer Magnetic RAM) which provides substantial power and area savings, near zero leakage power, and displays higher memory density compared to conventional SRAM. But STT-MRAM suffers from inherit drawbacks like multi cycle write latency and high write power consumption. So, these problem have to addressed in order to have an efficient design to incorporate STT-MRAM for NoC input buffer instead of traditional SRAM based input buffer design. Motivated by short intra-router latency, previously proposed write latency reduction technique is explored by sacrificing retention time and a hybrid design of input buffers using both SRAM and STT-MRAM to "hide" the long write latency efficiently is proposed. Considering that simple data migration in the hybrid buffer consumes more dynamic power compared to SRAM, a lazy migration scheme that reduces the dynamic power consumption of the hybrid buffer is also proposed.

DEDICATION

To my family

ACKNOWLEDGEMENTS

Firstly, I would take this opportunity to thank my advisor and mentor Dr. Eun Jung Kim who during the complete course of my research and thesis writing has been a constant source of encouragement and guidance. This thesis would have not been possible without her support and experience. I would like to thank my committee members, Dr. Rabi Mahapatra and Dr. Paul Gratz for their valuable feedback and advice.

I would also like to thank my team members at High Performance Computing group especially Hyunjun Jang, Baik Song An, Rahul Boyapati for their help and suggestions.

Finally, I would thank my parents for their constant support and always believing in me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The Moore's law, which describes the trend of doubling the transistor count on a chip while keeping cost same in approximately two years' time frame is still stubbornly being followed and realized by chip makers like Intel with advanced fabrication and transistor technology like 3D transistors in Ivy bridge processors. This continued advancement in VLSI technology has enabled chip manufacturers to incorporate multiple processor cores onto single integrated circuit die known as Chip Multi Processor (CMP). Contrary to modern belief of Moore's law dying out, it can be rightly anticipated that a hundred core CMP based system is not too far away.

Prior to CMP systems, one of the primary aim of chip designers was to increase the performance of processors, but now with ever reducing feature size and increasing performance of CMP systems, other aspects like memory, interconnect latency and power consumption have become dominant factors for performance bottleneck. For instance, many of mobile based processor's performance are intentionally scaled down to lower their power consumption and heat dissipation. In context of CMP systems, there is limited real estate and power budget to be shared between processors, caches and interconnect. Also, network performance has emerged as one of the most significant overhead in ever increasing number of processors in CMP systems. Network overhead can be alleviated to a certain extent by having a bus but this is accompanied by

---

This thesis follows style of *IEEE Transaction on Very Large Scale Integration Systems.*

scalability issues and whole idea of only one party talking at any given instant of time defeats the whole purpose of a high performance computing environment.

Power and heat dissipation has always been a major concern for chip designers, even more so now with large scale CMP systems and battery power constrained mobile devices. Power consumption of a processor is primarily due to switching activity and leakage power. Switching power consumption, also known as dynamic power can be reduced by reducing amount of switching activity or in other words the operating frequency of the processor. But, leakage power is harder to reduce and leakage power woes are getting worse with decreasing feature size. As for memory, there is a huge disparity between the performance increase of processors and memory technology over time. It has also been noted that by only increasing processing speed of a processor will have limited impact on performance of the overall system because of memory access time limitations.

Advancement in recent memory technology has ushered in new type of Non Volatile Memory (NVM). These include PCM, STT-RAM and Flash memory to name a few. Each of these memories have different performance and operational characteristics and each suited for different applications. For Network on Chip (NoC) design, STT-RAM is being regarded as a promising technology. It is next generation memory that uses magnetic field as the main information carrier. Its main advantage is its high density, low leakage power and high endurance compared to other nonvolatile memories which makes STT-RAM an attractive alternative for on-chip memory. However, one of the biggest drawbacks of STT-MRAM is long write latency as compared to SRAM.

Since the fast access time of memories on a chip cannot be compromised at any cost, slow write operations of STT-MRAM limits its practicality, even though it has SRAM comparable read performance. Another considerable drawback of STT-MRAM is high power consumption in write operations.

Despite these shortcomings, using STT-RAM in the NoC design can have significant positive affect on the performance since an on-chip router with a higher density STT-RAM based input buffers can accommodate larger amount of memory compared to pure SRAM for the same area budget. Larger input buffers contribute to improving the throughput of NoC, which results in the enhancement of overall system performance. However, previously mentioned challenges of STT-RAM have to be addressed prior to exploiting the benefit of STT-RAM in NoC.

Contribution of this thesis is based on the underlying assumption that using STT-MRAM based NoC buffer can increase throughput while keeping power consumption in lines with SRAM based NoC buffer. First, relation between write latency, write energy and retention time is exploited based on techniques described in [1],[2]. Next, based on intra-router latency, power consumption of write operations and write latency, an optimum hybrid model comprising of SRAM and STT-RAM is proposed, wherein the incoming flit coming into the router is first stored into SRAM and then immediately migrated to STT-RAM. Since in the hybrid model proposed, each flit is written twice, once onto SRAM and then onto STT-RAM there is added power consumption. Simply migrating each flit from SRAM to STT-MRAM buffer causes significant power consumption due to the high write power of STT-MRAM, compared to existing SRAM

based input buffers. So a lazy migration scheme is proposed that allows flit migration only when the network load exceeds a certain threshold, which helps to significantly reduce power consumption. Simulation results show that the hybrid input buffers improve the network throughput by 21% in synthetic workloads and 14% in SPLASH-2 parallel benchmarks on average compared to pure SRAM based buffers with the same area overheads. Also, the lazy migration scheme contributes to power reduction by 61% on an average compared to the simple migration scheme that always migrates flits from SRAM to STT-MRAM.

Rest of this thesis has following structure: Related work is discussed in Section 2, followed by performance and power model of STT-MRAM in Section 3. In Section 4, hybrid buffer design using STT-MRAM is proposed and explained in detail. In Section 5 simulation and experimental results of the proposed model is presented, and finally Section 6 gives a brief summary of the work presented in this thesis and conclusion is made.

2. BACKGROUND AND RELATED WORK

2.1 Non-Volatile Memories

For most part of modern computer system era, SRAM have been used for on-chip caches, DRAM is used as main memory and magnetic disk/flash memory as secondary memory. However with chip designers expected to follow Moore's law into foreseeable future, limitations poised by memory technologies is threatening to derail this trend. Thus it has become imperative to find new modern memory technology to keep up with advancement of chip design scalability. As a consequence, several non-volatile memory technologies are currently in various stages of development.

2.1.1 Flash Memory

Flash is the oldest and amongst the first NVM to be commercialized. It is an intended replacement for EEPROM. The main attractive feature is flash memory can be electrically erased and programmed.

NOR and NAND are two types of flash memory currently available. In NOR, the cell arrangement is similar to a NOR gate. One of each cell's terminal is grounded and the other is connected to bit line. Similarly in NAND flash, the cell arrangement resembles a NAND gate. Cells are connected in series and all NAND cells constituting the word line are pulled up together. Thus, NAND flash offers higher density but NOR flash offers faster read access.

Flash memory when compared to disk, offers significant write and read performance advantage, and consumes significantly less power due to absence of any movable parts. But it suffers from poor density and very low endurance.

2.1.2 Phase Change Memory

Phase Change Memory(PCM) uses behavior of phase change material as a binary storage medium. The actual physical state of this material is changed by the heat produced by passing of electric current through it.

Phase change material displays two types of states:

- Amorphous(Equivalent to high resistivity)
- Crystalline(Equivalent to low resistivity)

Amorphous state can be attained by cutting off current supply. In the similar manner, crystalline state is obtained by passing high pulse width current to heat the material. The duration and intensity of current required to attain crystalline state depends on the properties of the phase change material used, so type of material used play a considerable factor in determining the speed of write operation.

Compared to flash memory, PCM scores in almost all aspects. It has better read and write latency. Even though endurance is much better than Flash memory, it is still isn't high enough to be used in high write endurance requirement applications like on-chip memory.

2.1.3 Magnetoresistive RAM (MRAM)

MRAM is next generation of NVM that uses Magnetic Junction Tunnel (MJT) as a binary storage medium. A MJT is made up of two ferromagnetic layers and an oxide layer (usually MgO) is used as a tunnel between the two layers. Of the two ferromagnetic layers, one of them keeps the direction of its magnetic field always constant and this constant layer is known as reference layer. The direction of magnetic field in the two ferromagnetic layers determines the resistivity of the MRAM cell: if the direction of magnetic field is same in two layers, MJT displays low resistivity; if the direction is opposite it displays high resistivity. The direction of magnetic field in the two layers is changed by using a current induced magnetic field.

Current MRAM technology suffers from scalability, density and energy constraints. A more advanced technology based on MRAM is currently into advanced stages of development, known as Spin Transfer Torque- Magnetoresistive RAM (STT-MRAM). It is a very promising technology and displays most characteristics of a universal memory. STT-MRAM will be discussed in next section.

2.2 Spin Transfer Torque-Magnetoresistive RAM (STT-MRAM)

As with MRAM, STT-MRAM also uses MJT as the main storage component for binary data. The main difference: In STT-MRAM direction of magnetic field in the free layer is changed by passing spin polarized current through the free layer instead of current induced magnetic field in MRAM. An MTJ comprises of a three-layered stack: two ferromagnetic layers and an MgO tunnel barrier in the middle. One of the

ferromagnetic layer's magnetic spin direction is fixed and the other ferromagnetic layer's magnetic spin is free to be manipulated. High amplitude current is used to change the direction of magnetic spin in free layer by first making it pass through the fixed layer which polarizes the current, and then the spin polarized current is passed through the free layer, and depending on the direction of the spin polarized current the MJT is made to exhibit high and low resistivity. Figure 1 below shows the two states of MJT, resistance is high when direction of spin are anti-parallel in free and reference layer and resistance is low when direction of spin is same in both layers.

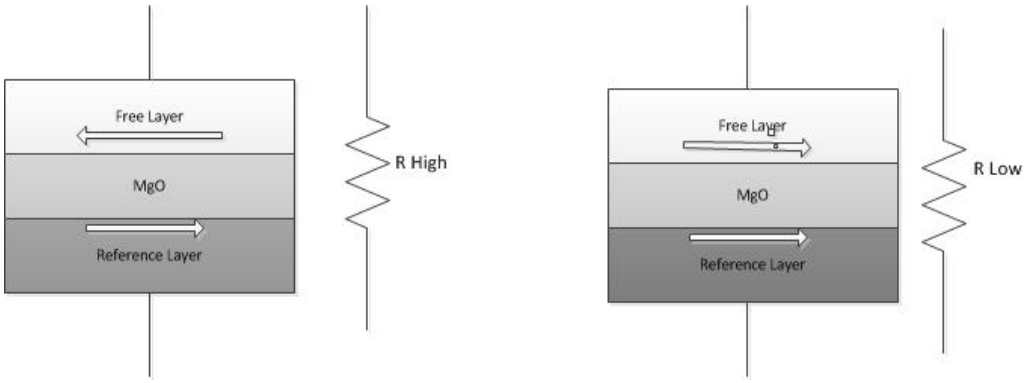


Figure 1. Two states of MJT Module

2.2.1 STT-MRAM Cell Design

A simple STT-RAM based memory cell can be designed with a single MJT connected in series with and a single transistor: commonly known as 1T-MJT as shown in Figure 2. Memory array of a STT-RAM based memory cell is similar to that of any SRAM and DRAM based memory array. Each memory cell consists of source line, bit line and word line. The source line is connected to source of the transistor, bit line is

connected to free layer of the MJT and the word line is connected to the drain of the transistor.

Read operation is carried out by applying an appropriate amount of voltage on the world line to select the desired cell and biasing bit line and source line. To ascertain if current is passing through the MJT, a sense amplifier is used which is connected to the bit line of the memory cell. If the magnetic field in the two layers of MJT is parallel the resistance through the MJT is low, indicating a "1" state and if the magnetic field in the two layers is anti-parallel then there is high resistance through the MJT, indicating a "0" state. Write operation requires an application of much higher V_{dd} voltage through the access transistor to provide enough current to modify the spin in the free layer. Figure 3 describes the schematic of a STT-MRAM based cell array.

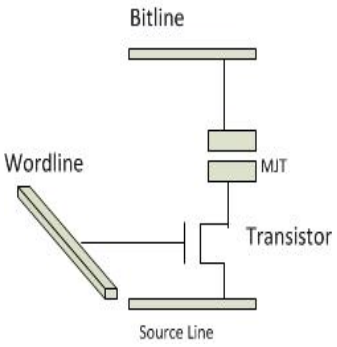


Figure 2. 1T MJT Cell

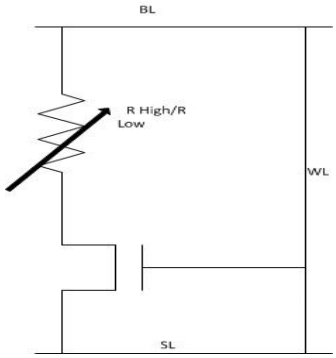


Figure 3. STT-MRAM Schematic

2.2.2 Retention Time

Retention time of an MJT is the time until there is a random flip in the bit information and it is determined by the thermal factor (Δ)

$$\Delta \propto \frac{V.H_K M_S}{T} \quad (1)$$

$$I_C(writetime) = A.(J_{c0} + \frac{C}{writetime^{\gamma}}) \quad (2)$$

$$t \propto C * e^{k\Delta} \quad (3)$$

From Equation 1 it can be seen that Thermal factor is directly proportional to volume (V) of the free layer and inversely proportional to operational temperature (T). Equation 2 gives the relation between switching current density and switching time. Equation 3 describes the relation between retention time (t) and thermal barrier (Δ), in other words, retention time increases exponentially with increase in Thermal factor.

There are multiple ways to reduce the thermal factor. One technique as described in [1], using Equation 1 above, thermal factor is reduced by decreasing the thickness of the thermal barrier and lowering saturation magnetization. Switching current of MJT decreases as thermal barrier decreases thus reducing write energy consumption and achieving faster write speed (Equation 2). The second technique described in [5] is to increase the write current, increasing write current will decreasing switching time and thus decreasing write latency.

2.3 Network On-Chip Basics

There are multiple benefits to using a NoC than a shared bus. In a shared bus, only one party can talk at a time and this leads to scalability issues. In a traditional Integrated Circuit design with point-to-point wire between two communicating nodes is impractical in CMP context since a CMP based system is expected to have more than hundred cores in near future. So, if a dedicated line exists between every two cores then this leads to massive overhead and impractical for commercial implementations.

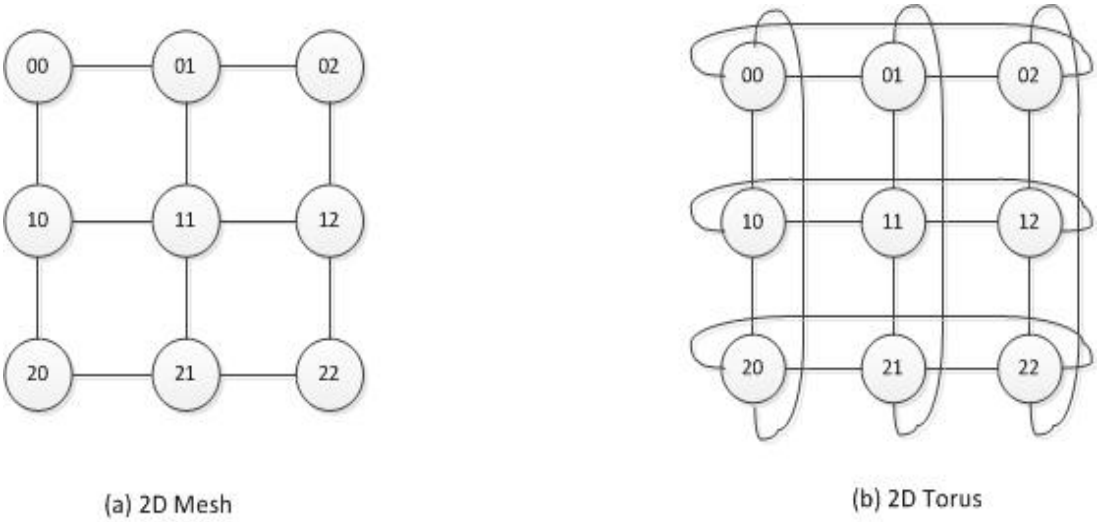
NoC limits the above stated problems by reducing complexity of implementing a communication medium capable of handling more than hundred nodes efficiently. There can also be more than one path between source and destination which give redundancy to commutating nodes. Also, route can be efficiently chosen based on Quality of Service (QoS) requirements.

2.3.1 Topologies

Network on-chip (NoC) consists of channels and nodes. Nodes constitute elements like routers, terminals and channels which is the actual interconnection between the nodes. Topology is a graphical representation of these communicating elements along with interconnects. Figure 4 shows different topologies commonly used in NoC design. Selecting the topology is the initial step NoC designers take because all other factors like routing algorithms and flow control mechanism depends on it. Selecting a good topology depends on many of the requirement constrains put forth

towards the NoC designers; these mainly include but not limited to bandwidth, radix of switch, number of I/O ports. Assessment of a NoC topology is based on two factors: cost and performance. Cost of a topology is the complexity and implementation overhead for realizing the topology. Performance is measured based on bandwidth and latency, these two attributes are measured based on bisection bandwidth, channel load, and path delay.

Figure 4(a) and 4(b) describes the Mesh and Torus network, both these networks are bidirectional. Designers can choose the appropriate network based on the requirements. For instance Torus network has better path diversity choices and better load balancing compared to Mesh network but Torus network displays lager hop count compared to Mesh network. Figure 4 (c) describes a butterfly network, which is a unidirectional network in which each node is connected to a switch. Switching nodes pass the packets along the appropriate output link.



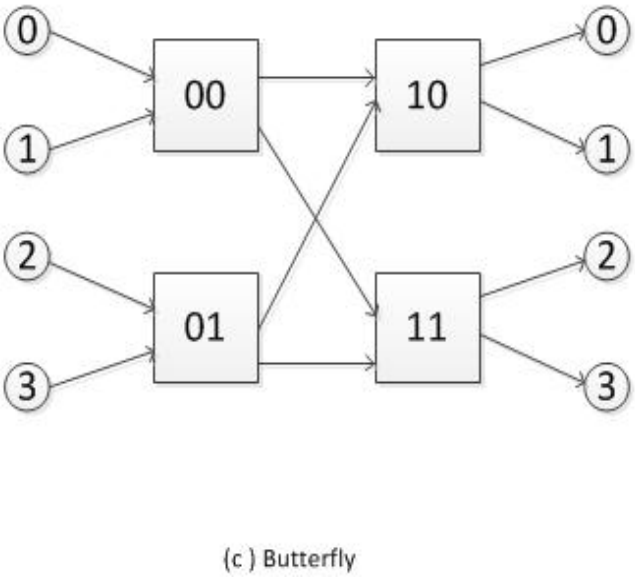


Figure 4. Common NoC Topologies

2.3.2 NoC Router Architecture

A router is a NoC component that is responsible for routing and flow control of flits in a NoC. A typical virtual-channel router Figure 5, has two major groups: datapath and control plane. Datapath is responsible for temporary storage of incoming flits and forwarding it to the appropriate output port. Control plane is responsible for implementing resource allocation and flits movement.

Input buffers are used to store the flits that come into the router. Before the packet is forwarded, the route to the next hop is computed by the route computation block. After output port is determined, output virtual channel is requested from virtual channel allocator. Same virtual channel is used by all flits belonging to a packet to get to next downstream router. Switch allocator is responsible for allocating the time slot and output channel for the switch for each packet.

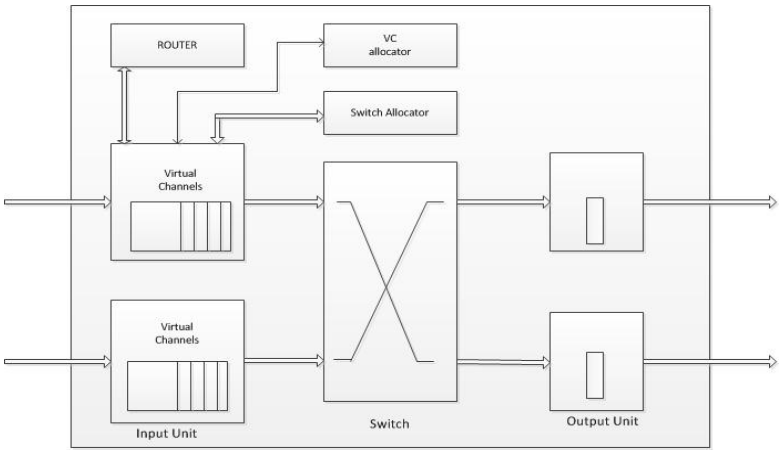


Figure 5.Typical NoC Router

To get higher throughput, routing is implemented as 4 pipeline stages as shown in Figure 6. Four pipeline stages are Routing Computation (RC), virtual-channel allocation (VA), switch allocation (SA), and switch traversal (ST). Routing begins when the header flit comes into the router; in RC stage the output port is determined. Output virtual channel is allocated in VA stage. ST stage handles switch arbitration between input and output ports.

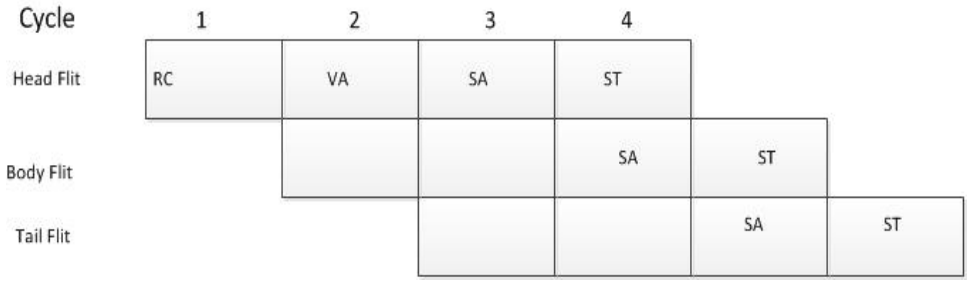


Figure 6. Router Pipeline Stages

Several techniques have been devised to decrease router pipeline depth which results in increased throughput. As shown in Figure 7, Lookahead routing and Speculation routing is implemented in the first cycle.

- Lookahead routing eliminates one routing pipeline depth by computing the route one hop prior.
- Speculation routing works based on the speculation or in other words assumption that virtual channel is granted, so SA stage can be executed along with VA stage thereby decreasing the pipeline depth by one. If

there is failure in virtual channel allocation then both the above SA and VA stages are to be repeated again in next cycle.

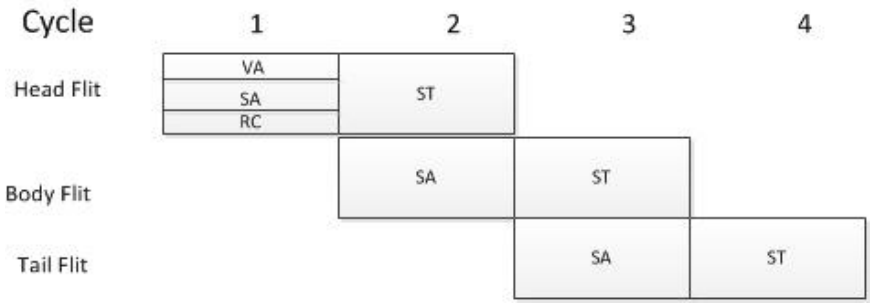


Figure 7.Router Pipeline with Lookahead and Speculative Routing

2.3.3 Routing Algorithms

There are many routes that packets can take between source and destination. Routing algorithm determines the exact route the packets must take. The decision can be based on current state of the network, resource availability among many other things.

There are mainly three types of routing algorithms:

- **Deterministic:** The path between source and destination is pre-computed and it remains constant.
- **Oblivious**: Like deterministic routing, oblivious routing too doesn't consider the present state of the NoC network to determine the path between source and destination and there is only one path between any two pair of source and destination. But it can distribute traffic across path based on a random algorithm.

- **Adaptive Routing**: Adaptive routing constantly tracks the state of the network and determines the best possible route between a source and destination based on current resource requirements and state of the network

2.3.4 Router Buffer

Router buffers are the most important aspect of a NoC router and influences the throughput and hence performance of over-all system to a great extent. Buffers are also the main culprits when it comes to power consumption and latency of router. So, it's highly critical to ensure an appropriate buffering mechanism which provides an acceptable tradeoff between power consumption and performance of the router.

There are multiple ways to partitioning a buffer as shown in Figure 8

- **Central Memory:** In central memory, a single memory is used to hold all flits coming in from all input channels and is also responsible for servicing all the output channels. This technique virtually eliminated the switch (for directing flits from input to appropriate output) but it requires one multiplexer for input side and one de-multiplexer, so this saving of a switch is nullified. Central memory's major drawback is its bandwidth could become a serious bottleneck.
- **Separate memory per input port:** Separate memory port is provided to each input port. This maximizes bandwidth utilization and decreases latency. But can cause non-uniform buffer memory utilization if one of the memory ports has high traffic while others do not.

- **Separate memory for each virtual channel:** This technique has the maximum input speedup and throughput because the switch can access more than one virtual channel belonging to same physical channel.

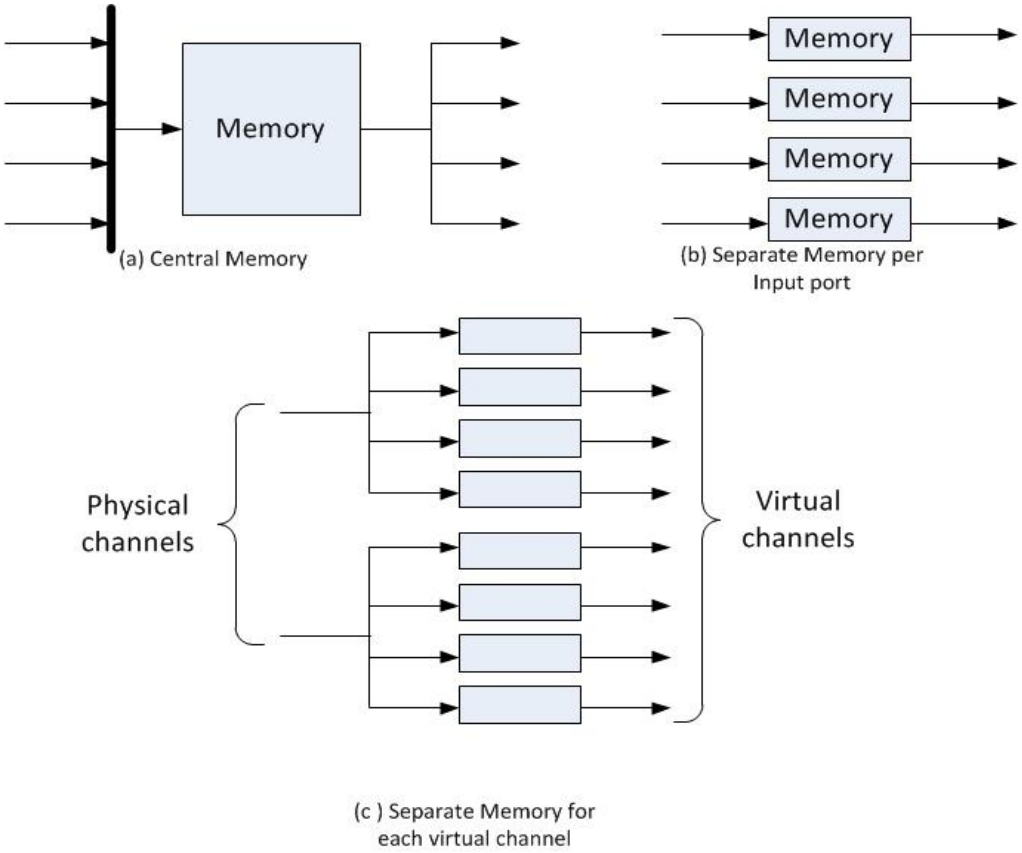


Figure 8.Buffer Partitioning

2.4 Related Work

There has been no prior work with incorporating STT-MRAM in NoC level design, so only relevant work and background of STT-MRAM memory and application of other NVM in various memory hierarchy of a computer system is discussed.

Several schemes have been proposed to provide architectural support for applying NVMs to system components. Jog et al. [1] proposed to achieve better write performance and energy consumption of STT-MRAM-based L2 cache through adjusting data retention time of STT-MRAM. Similarly, Smullen et al. [2] reduced the write latencies as well as dynamic energy of STT-MRAM by lowering the retention time for designing on-chip caches. In [3], they integrated STT-MRAM into on-chip caches in a 3D CMP environment and proposed a mechanism of delaying cache accesses to busy STT-MRAM banks to hide long write latency. Prior to that, Sun et al. [4] stacked MRAM-based L2 caches on top of CMPs and reduced overheads through read-preemptive write buffer and hybrid cache design using both SRAM and MRAM. Guo et al. [5] resolved the design issues of microprocessors using STT-MRAM in detail for more power-efficient CMP systems. PCM also has been constantly explored to replace existing SRAM or DRAM-based memory systems. Due to its lower endurance compared to SRAM or STT-MRAM, PCM is mainly adopted for off-chip memories rather than on-chip caches. Several designs of PCM-based main memory were discussed in [6], [7], [8]. In [9], adaptive write cancellation and write pausing policies were proposed to reduce energy and improve performance.

3. STT-MRAM MODELING

ITRS 2009 projections [11] and cell parameters from Guo et. al [5] are used to obtain the area model of a 1T-1MJT cell size where each cell is considered to be $30F^2$ in the 32nm technology. If SRAM cell size is considered to be about $146F^2$ in 32nm technology, four STT-MRAM cells can be packed instead of one SRAM cell for the same area constraint. For energy model, read and write parameters are adopted from [5] which are about 0.01pJ for read and 0.31pJ for write energy. As discussed in section 2.2.2, there are two ways by which write latency can be reduced. Both these techniques are used in this work. Write latency can be reduced to 3.2ns which corresponds to about 10 cycles in a 3GHz machine with a 30F2 STT-MRAM cell size [5]. But 10 cycles for a write operation is considerably long for on-chip memory application compared to a SRAM based memory system which can perform read and write operation in a one clock cycle. Retention time can be reduced from 10 years to about 10ms while keeping same write latency but reducing retention time decreases write current required for a write operation by about 33% [1]. In a 1T-1MJT STT-MRAM based cell design transistor accounts for major part of memory cell area, so by decreasing write current a smaller transistor can be used thereby decreasing STT-MRAM cell area. But in this work, reducing write latency is one of primary objective so write current is increased slightly so that switching time is reduced and thus decreasing write latency. The write latency is reduced from 3.2ns to 1.8ns by increasing the write current from 50μA to 75μA at 125 C temperature. By increasing write current, write latency is reduced from 10 to 6 cycles in

3GHz system. Since MJT switching time decreases, increased write current has negligible overhead with write energy consumption [5]. Increase in write current could affect read latency performance but it has been verified that decreasing write latency by a magnitude from 3ns to 1.8ns has limited effect on read latency [2]. An increased read latency can also be accommodated in a one cycle since a read delay of 122ps [5] is far smaller than a 333ps duration of a single of 3GHz based system.

Relaxed retention implies data content uncertainty beyond the retention time threshold. So, implication of reduced retention time on NoC buffer design has to be determined. Maximum time a flit is stored in the buffer is determined based on intra router latency. If time spent by flit in buffer is more than retention time then this could imply possible data corruption and considered a dropped flit. Intra-router delay is computed by computing the time difference between arrival time at the buffer and departure time in a router. Figure 9 shows maximum intra-router latency for various injection rates and various numbers of buffers available per virtual channel using uniform random synthetic workloads. It can be inferred from the graph that the latency does not exceed 16 cycles, which is still substantially less compared to 10ms corresponding to 30 million cycles of a 3GHz system. Hence, reducing retention time will have effect on reliability of flits being stored in buffer.

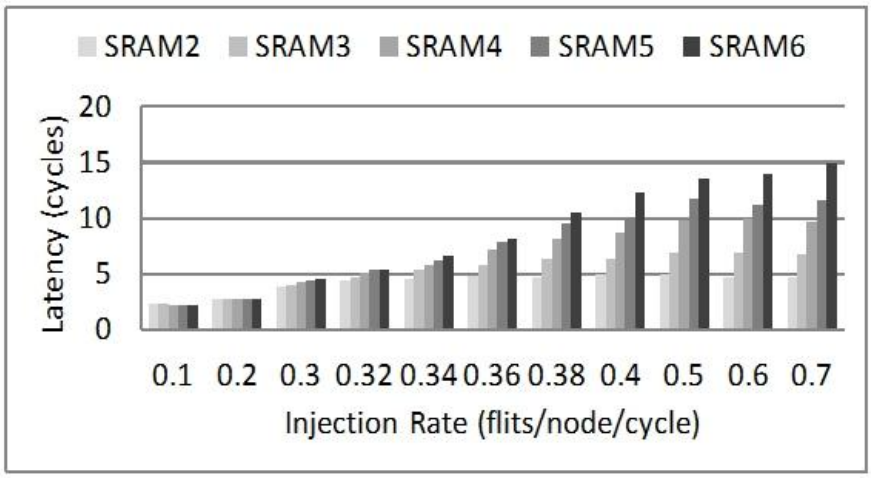


Figure 9. Maximum Intra-Router Latency of an On-Chip Router

4. STT-MRAM HYBRID BUFFER ROUTER

In this section, generic SRAM buffer based router architecture is described. Next, in order to exploit advantages of STT-MRAM while minimizing its drawbacks at the same time, a hybrid SRAM and STT-MRAM based router buffer is described.

4.1 Generic Router Buffer

A generic SRAM based router is shown as in Figure 10. It is similar to the router described in section 2.3.2. It is a speculative router, using lookahead routing scheme and uses credit based flow control.

Buffers are the main culprits when it comes to power consumption and latency in the router. To minimize these effects, buffers are implemented as simple First-in-First-Out (FIFO) structure. In a VC based NoC, each physical channel is associated with multiple virtual channel with its own individual buffer, as shown in Figure 11, FIFO buffers are implemented as serial and parallel. In serial FIFO, each flit has to traverse all buffer entries as shown in Figure 12, but in parallel FIFO on the contrary eliminates this restriction [16]. But implementing parallel FIFO involves keeping track of two pointers: one for read and other for write compared to just one for serial FIFO. Read pointer points to the head of the queue and controls input demultiplexer, and write pointer points to the tail of the queue and controls output multiplexer. During read operation, the flit pointed to by the read pointer is transmitted to the crossbar and eventually to the appropriate output port of the router. During write operation the flit coming in from the

input channel is stored in the location pointed to by the write pointer. Pointer control logic takes care of updating each pointer after every read and write operation.

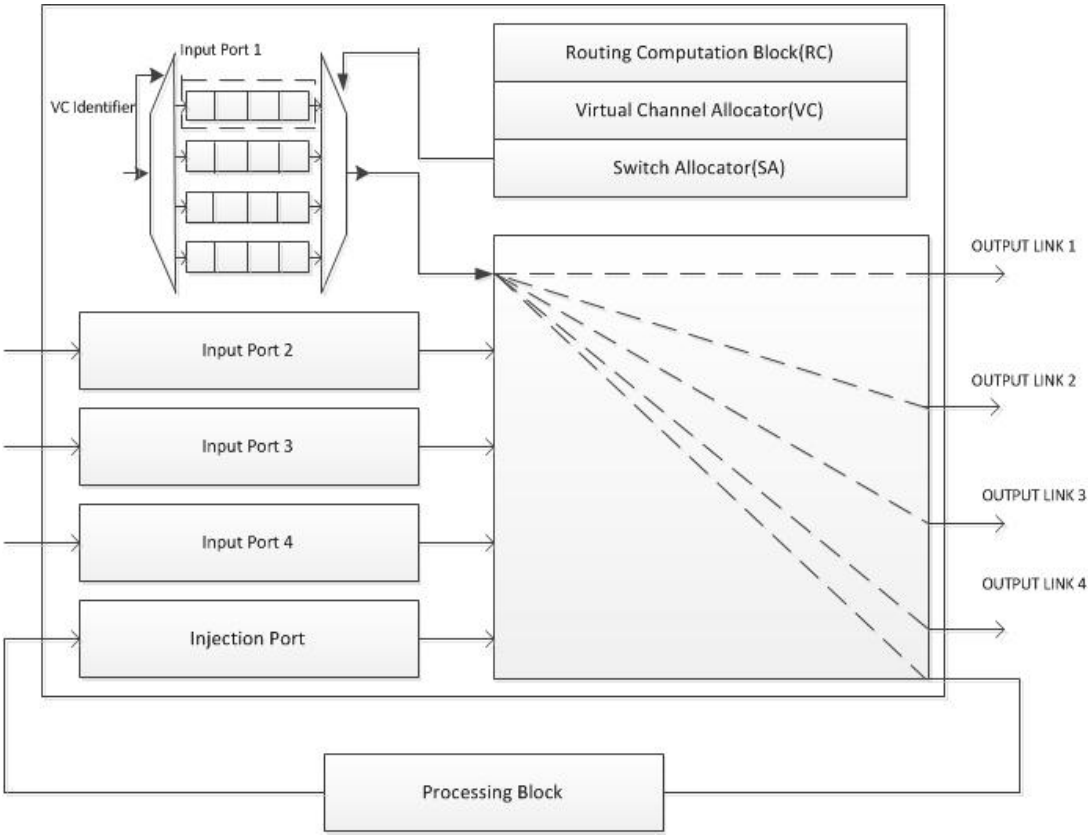


Figure 10. Generic Router Architecture

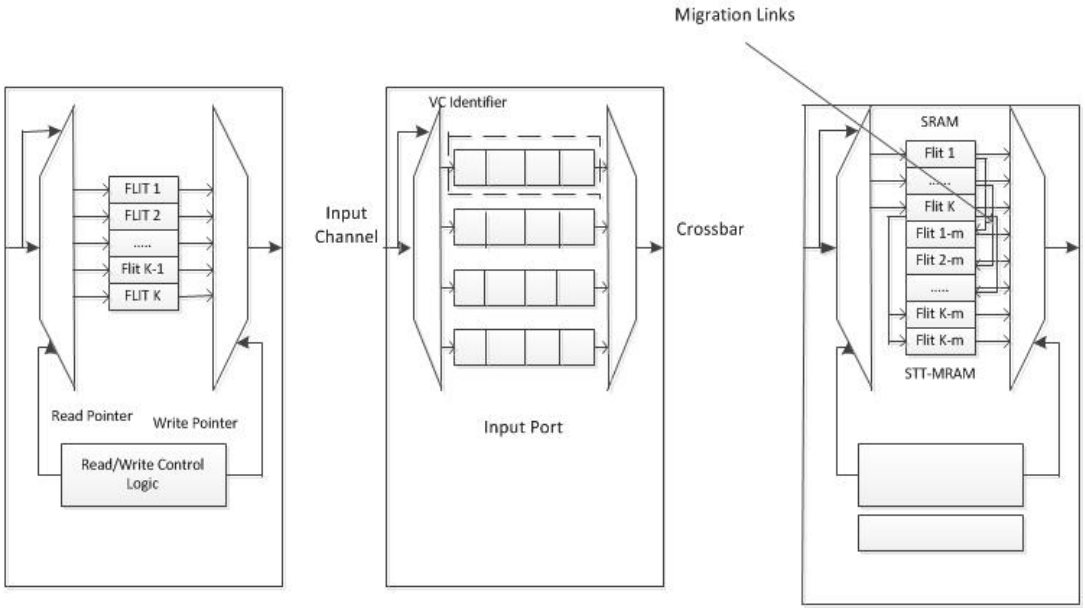


Figure 11.Generic SRAM Input Buffer and Hybrid Input Buffer

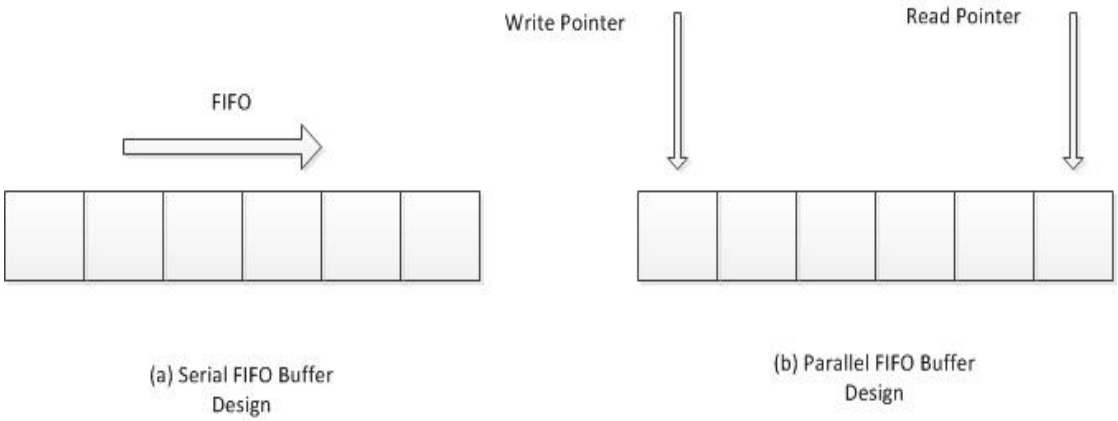


Figure 12. Series and Parallel FIFO

4.2 Hybrid Buffer Design

In this section hybrid router architecture is proposed that will effectively hide the major disadvantage of long write latency of STT-MRAM while maximizing its advantages. As previously stated one of the key advantages of STT-MRAM is that it possible to accommodate four times the memory for the same area budget [5], [17], thereby increasing buffer size by effectively four times for the same area budget. Increased buffer memory would imply increased throughput as a larger buffer size would imply a higher saturation point for the NoC. Since write latency of STT-MRAM is considered to be six cycles, this implies data has to be provided to the write port of STT-MRAM for at least six cycles. Providing input for six cycles is impractical on an on-chip router buffer. One solution to this problem is to design a memory system that will effectively hide this write latency issue while still delivering the advantages of STT-MRAM memory.

Figure 11 shows an architectural level implementation of hybrid buffer design. STT-MRAM is connected to each VC and each SRAM cell in parallel. Separate migration links are used to link each SRAM cell to STT-MRAM cells. As with parallel FIFO buffer implementation, one pointer for read and one for write are to be maintained. But in hybrid, since flits can be read from both SRAM and STT-MRAM, two read pointers are required: one pointer to read from SRAM if flit is to read before migration to STT-MRAM is completed and other read pointer to read from STT-MRAM. Only one write pointer is sufficient as flits are written only in SRAM.

Flow control based on feedback of buffer availability in downstream router is based on the availability of only SRAM as flits are to be written in SRAM first.

4.2.1 Write Mechanism

This section describes simple write mechanism in hybrid buffer design.

- VC flow control is allocated based only on the availability of SRAM in downstream router; availability of STT-MRAM is not of concern.
- There is a single write pointer in hybrid buffer design and it points to an empty slot in SRAM.
- After a flit is written into SRAM, it is migrated to STT-RAM

4.2.2 Read Mechanism

This section describes read mechanism in hybrid buffer design.

- There are two read pointers, one for SRAM and other for STT-MRAM
- SRAM and STT-MRAM can be thought as one large FIFO buffer. If STT-MRAM part of the buffer is empty it implies that migration has not been completed from SRAM. Thus read operation must be completed from read pointer SRAM and migration of flit to STT-MRAM must be terminated immediately.
- If STT-MRAM part of the buffer is not empty, slot pointed by STT-MRAM read pointer is read.

4.2.3 Migration Scheme

The main design goal of the previously described hybrid model is to provide a seamless read and write operation while incorporating latest NVM: STT-MRAM. As stated earlier, STT-MRAM offers far superior memory density compared to SRAM, and thus by incorporating STT-MRAM onto NoC router buffer design increased throughput can be achieved. But to hide long write latency an effective migration scheme as describe in Figure 13 is proposed in which each VC consists 12 STT-MRAM augmented with 6 SRAM buffer entries.

Two types of migration scheme are described: simple and lazy.

- **Simple Migration**: In this mechanism, write latency of STT-MRAM is considered to be six cycles. STT-MRAM is augmented with SRAM to hide this long write latency. If a flit comes into the router at 1^{st} cycle, it is written first into SRAM. Immediately, migration of the flit to STT-MRAM is starts and it completes at 6^{th} cycle. Considering maximum NoC utilization and flit arriving at the router every cycle, no space will be available in SRAM at the end of 6^{th} cycle. But at 6^{th} cycle, migration of flit to STT-MRAM is completed and an empty buffer entry is available in SRAM to accommodate the next incoming flit.
- **Power-Efficient Lazy Migration:** In simple migration scheme previously discussed, there is added power consumption due to two writes per flit: once in SRAM and once in STT-MRAM. This high power consumption can be justified in high load network where high throughput could be a good

tradeoff for higher power consumption, but in low load network wherein there is no need for a large buffer as there is a high probability that flits will leave the buffer before complete migration to STT-MRAM. Based on this observation, a trigger based migration scheme is proposed in which migration of flits from SRAM to STT-MRAM occurs only when network load per VC exceeds a predetermined threshold. This threshold is based on the ratio of number of flits in the SRAM buffer to the total SRAM buffer size. If the threshold exceeds a predetermined value, migration mechanism is triggered. By using the above sated lazy migration method in low network load scenario, considerable saving of write power is observed compared to simple migration mechanism. Detailed analysis is done in results section.

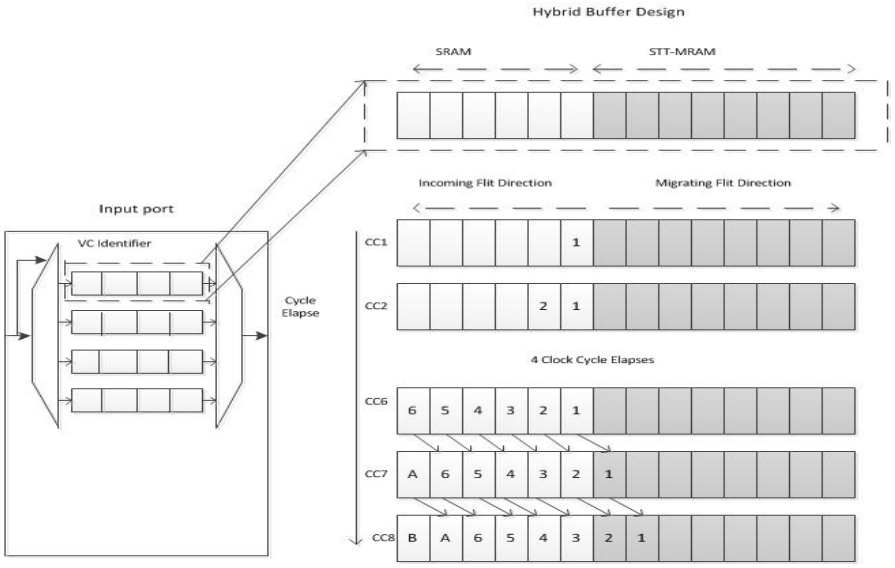


Figure 13. Hybrid Buffer Design Migration Scheme

5. EXPERIMENTAL RESULTS

In this chapter performance of baseline NoC router with SRAM based buffer and performance of proposed hybrid buffer design NoC is compared and analyzed. Various benchmarks and synthetic workloads are used along with different buffer configuration for evaluation purposes.

5.1 System Configuration

Simulations are carried out under 32nm technology, in 8x8 network having 32 out-of-order processors with each processor associated with its own L2 cache, as shown in Figure 14. Network is implemented with two stage speculative routers with lookahead routing, as describe in section 2.2.2. Each flit size is set at 16 Bytes and number of VCs is set as 4 while buffer depth is varied to simulate different configurations of buffer implementations. XY, O1TURN [18] routing algorithms are used along with wormhole switching flow control.

Performance evaluation is done using a variety of synthetic workloads like uniform random (UR), bit complement (BC) and nearest neighbor (NN). Traces are obtained using SIMICS [20] simulation tool. To evaluate performance under realistic environment and workloads SPLASH-2 [19] parallel benchmarks traces are used. Table 1 summarizes the configuration of the simulation environment.

To estimate power consumption Orion 2.0 [21] is used. Power consumption is estimated based on different configuration of the proposed hybrid design. Table 2

summarizes different power parameters associated with SRAM and STT-MRAM, and Table 3 summarizes simulation configuration. Unless explicitly stated, write latency of STT-MRAM is six cycles. As previously stated, STT-MRAM provides four time the memory density compared to SRAM. Size of SRAM and STT-MRAM are denoted by sram# and stt#

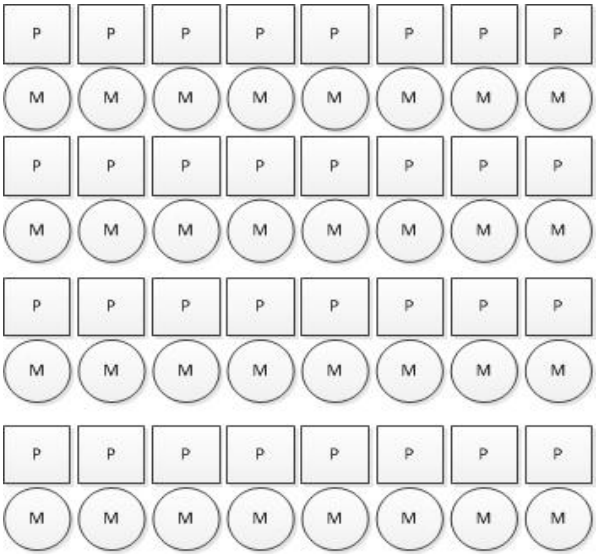


Figure 14. CMP Layout

Table I: CMP System Configuration

System Parameters	**Details**
Clock frequency	3GHz
# of processors	32
L1 I and D caches	direct-mapped 32KB (L1I) 4-way 32KB (L1D), 1 cycle
L2 cache	16-way 16MB, 20 cycles 32 banks, 512 KB/bank
Cache block size	64B
Coherence protocol	Directory-based MSI
Memory latency	300 cycles
Flit size	16B
Packet size	1 flit (Benchmark-control), 4 flits(Synthetic)

Table II: SRAM and STT-MRAM Parameters

Parameter	SRAM	STT-MRAM
Read Energy (pJ/flit)	5.25	3.826
Write Energy (pJ/flit)	5.25	40.0
Leakage Power (mW)	0.028	0.005
Area per cell	$146F^2$	$30F^2$

Table III: Simulation Configuration

Feature	Configuration
Topology	8 x 8 Mesh, 2D-Torus, Butterfly
Routing Algorithms	XY, O1TURN
Router Delay	2 cycles
Virtual Channel Per Port	4
Virtual Channel Depth	Variable
Workload	Unifor Random, Bit Complement, Nearest Neighbour

5.2 Performance Analysis

Performance of a NoC can be determined by latency flits experience as they traverse through the network. Less latency implies flits can travel same number of hops in lesser clock cycles and thus increasing throughput, and finally increasing performance of the entire system. Figure 15, 16 and 17 shows the performance of pure SRAM based buffer and various configuration of proposed hybrid model under UR, BC and NN traffic patterns respectively. Total area budget is fixed at SRAM 6 per VC of the input buffer; an equivalent hybrid buffer for the same area would include configuration SRAM2_STT16, SRAM3_STT12, SRAM4_STT8, SRAM5_STT4. In all cases, the hybrid design shows throughput improvement by 18% for UR, 28% for BC, and 17% for NN on an average. In all the three synthetic benchmarks simulation results, SRAM6 has

the worst performance and SRAM2_STT16 has the best performance. These results clearly validate the previously stated assumption that SRAM can effectively hide long write latency of STT-MRAM and resulting in increased performance due to larger buffer space availability provided by the hybrid model for the same area budget. Larger buffer can further delay network saturation and hence handle higher load compared to smaller buffer; for instance in SRAM6 configuration, the network saturates at approximately 0.32 injection rate. But with SRAM2_STT16 configuration the network saturates at 0.4 injection rate, which is an improvement of about 16%.

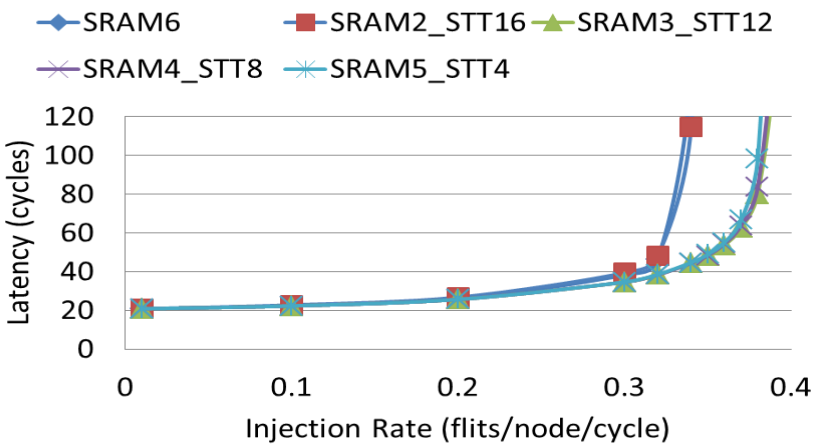


Figure 15. Performance Analysis with Synthetic Workload: Uniform random

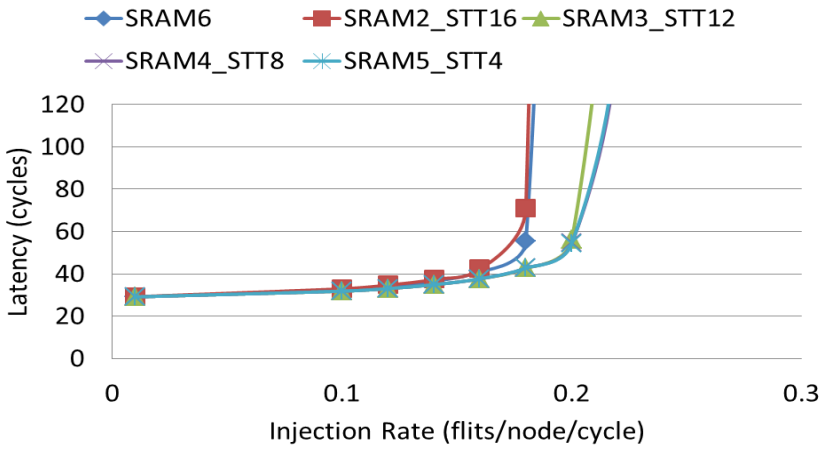


Figure 16. Performance Analysis with Synthetic Workload: Bit Complement

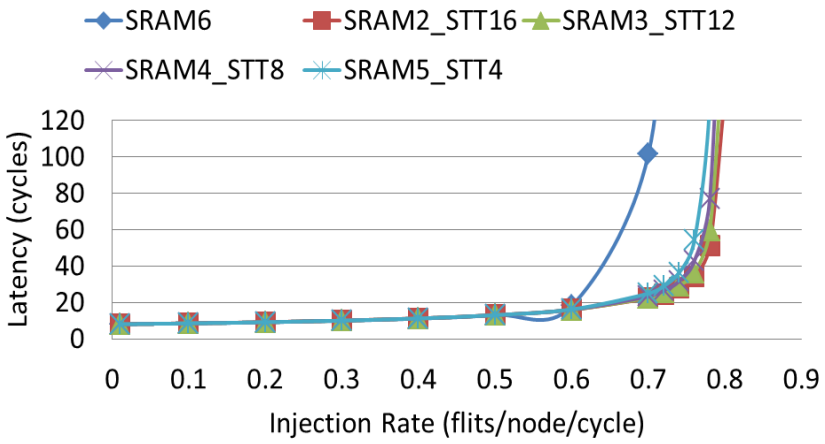


Figure 17. Performance Analysis with Synthetic Workload: Nearest Neighbor

To evaluate performance improvements in different topologies and routing algorithms, evaluation of hybrid design is also done using O1TURN [18] routing algorithm, and 2D-torus and flattened butterfly [22] topology. Figure 18 shows the

performance with O1TURN in the 8x8 2D-mesh topology, where the overall throughput increases by 15% on average, while Figure 19 shows the throughput is increased in 2D-torus and Figure 20 flattened butterfly by 13% and 15%, respectively.

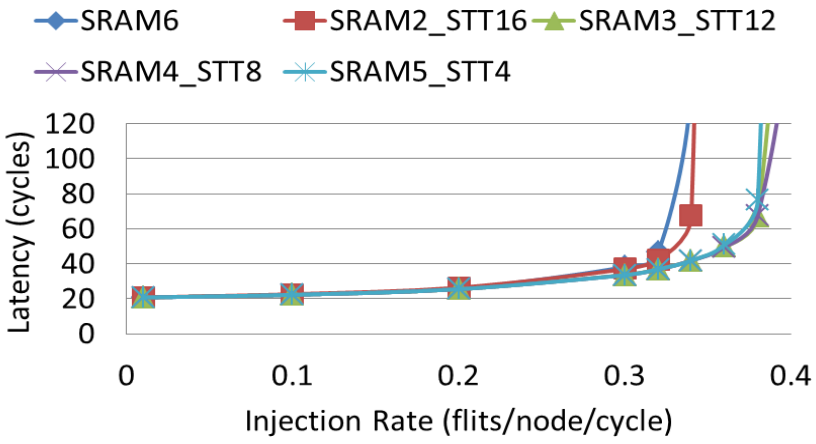


Figure 18. Performance Analysis O1TURN Routing Algorithm

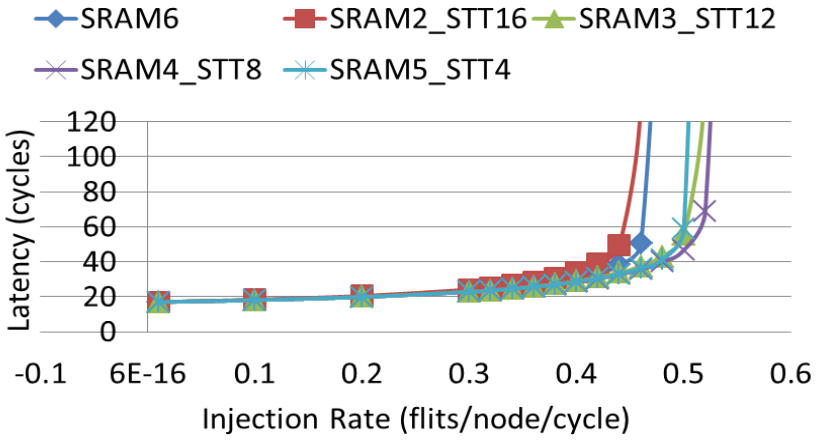


Figure 19. Performance Analysis with 2D Torus Topology

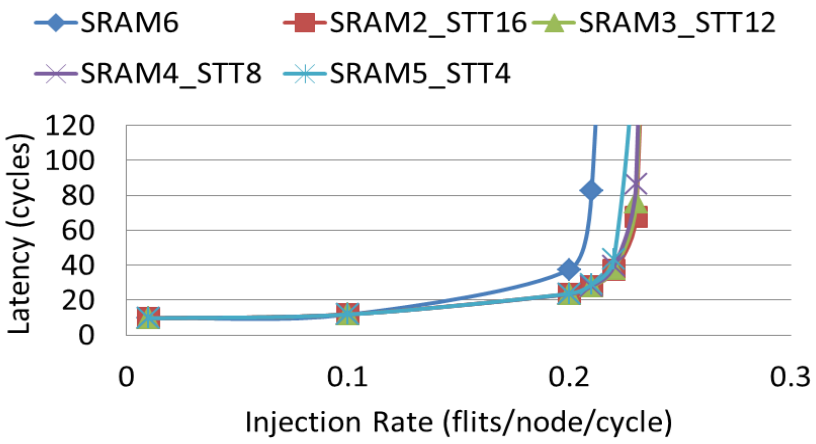


Figure 20. Performance Analysis with Flattened Butterfly

Next, to justify the need to decrease the write latency of STT-MRAM by reducing retention time as discussed in section 2.2.2, impact of different write latencies on overall performance of the network is evaluated. Figure 21, 22 and 23 shows network performance with 30, 10 and 6 cycles respectively as write latency of STT-MRAM. Experiments are carried out on 2D Mesh with XY routing. It can be easily inferred from the results that write latency plays a big role in determining the overall network performance. Among different hybrid buffer configuration, SRAM2_STT12 has the worst performance, even worse that pure SRAM6 buffer when write latency is 30 and 10 cycles, it matches performance of SRAM6 only when write latency is 6. This is because since in the proposed hybrid model, SRAM has to hold the flits equivalent to write latency of STT-MRAM, this implies in SRAM2_STT6 only two flits will be held in SRAM and free space will be available in SRAM only migration occurs after 30 or 10 cycles. This holds up all the subsequent flits in the upstream router greatly decreasing

the throughput of the network and also leads to early network saturation. As inferred from Figure 9, a flit remains in the buffer for at least three cycles, so any configuration less than SRAM3 in the proposed hybrid design leads to reduced performance.

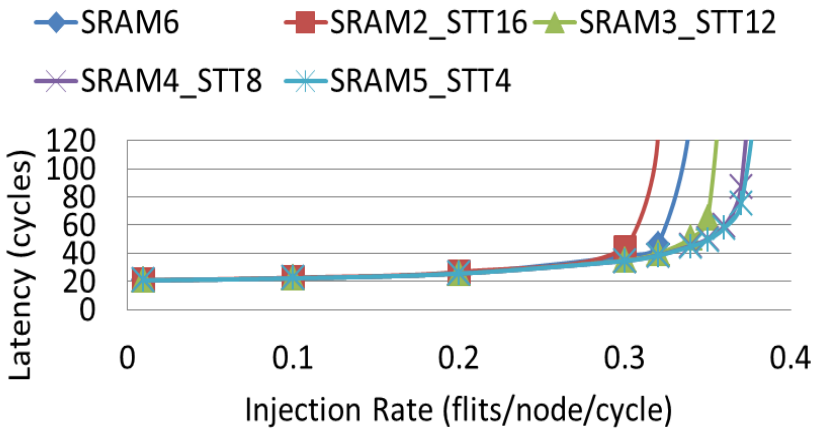


Figure 21. Performance Analysis with Write Latency: 30 cycles

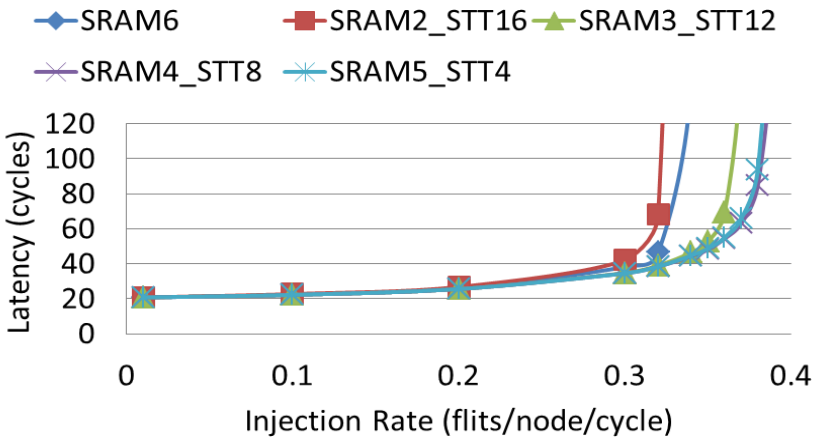


Figure 22. Performance Analysis with Write Latency: 10 cycles

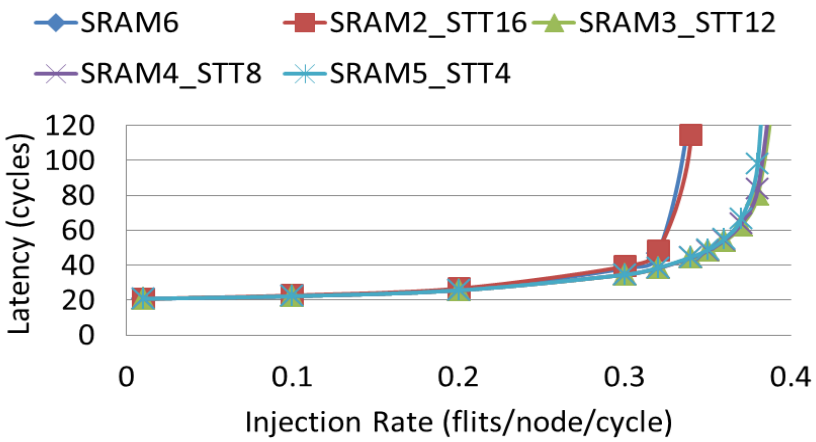


Figure 23. Performance Analysis with Write Latency: 6 cycles

It is also interesting to note that when write latency is very high: 30 cycles, better performance is seen in configuration having larger SRAM, as seen from the results SRAM5_STT4 provides highest throughput. When write latency is moderate: 10 cycles, SRAM4_STT8 provides the best throughput but when write latency is less: 6 cycles, best performance is provided by the configuration with maxim buffer size, except SRAM2_STT12 as previously stated minimum of SRAM3 is required for hybrid configuration, so STT3_STT12 shows the highest throughput compared to other configurations. To clearly see the relation between write latency and size of the buffer, Figure 24 shows normalized throughput for different configuration of hybrid buffer for corresponding write latencies. In case of a relatively long write latency, 30 cycles, the hybrid input buffer having the largest SRAM buffer outperforms the others by up to 11% compared to the pure SRAM6 buffer. Likewise, in case of low write latency, 6 cycles, except the SRAM2 STT16 case, the one having the largest total buffer size, SRAM3 STT12 beats the other configurations by up to 18% in terms of network throughput.

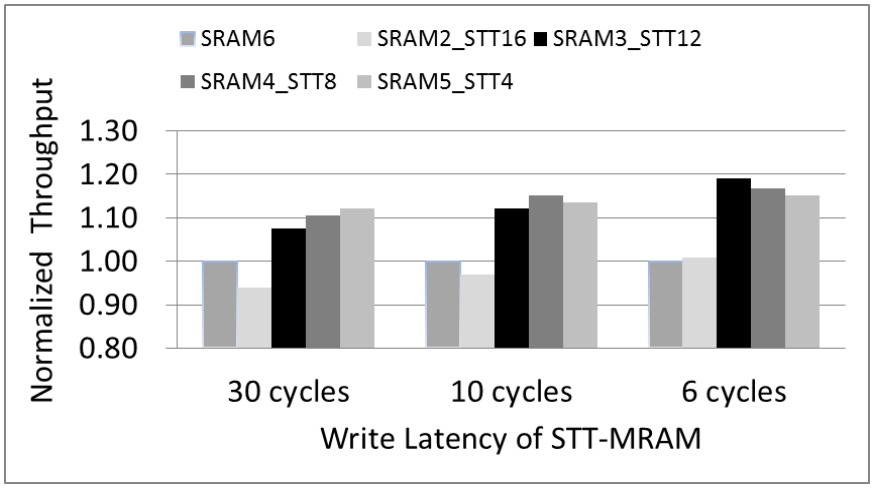


Figure 24. Normalized Throughput for different write latencies.

Experiments are also done using realistic SPLASH-2 benchmarks. Area budget of SRAM4 per VC is set, Figure 25. In general, the hybrid input buffer outperforms the pure SRAM based, by approximately 14% on average. Specifically, water-nsquared shows the best improvement by 34.5% while ocean shows the least improvement by 3.2%. The amount of improvement varies depending on the traffic patterns. It is observed that benchmarks showing higher improvement, hot spots exist in their communication, whereas in the benchmarks with slight performance improvement, communication is evenly spread across the whole network. Simulation was also done with setting area budget at SRAM6 as in Figure 26 and compared with the throughput of the pure SRAM-based buffer and the hybrid buffer. It is observed that there is negligible performance increase. This can attributed to the fact that since SPLASH-2 is a realistic workload benchmarks, injection rate seems to be too low to take advantage of the proposed hybrid buffer mechanism As the budget decreases from SRAM6 to SRAM4, the amount of improvement from using the hybrid buffer increases by approximately

5.5%. This trend indicates that the hybrid buffer is more beneficial as the area budget in CMP environments becomes tighter.

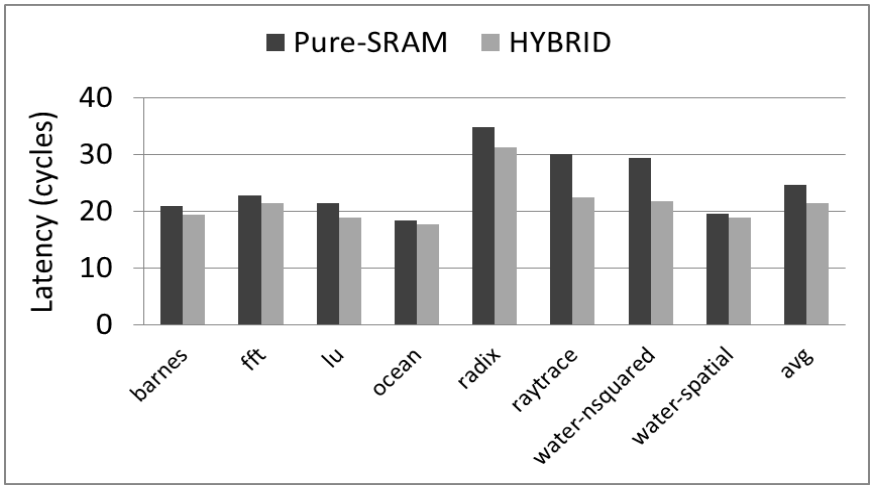


Figure 25. SPLASH-2 Benchmarks results with area budget SRAM 4

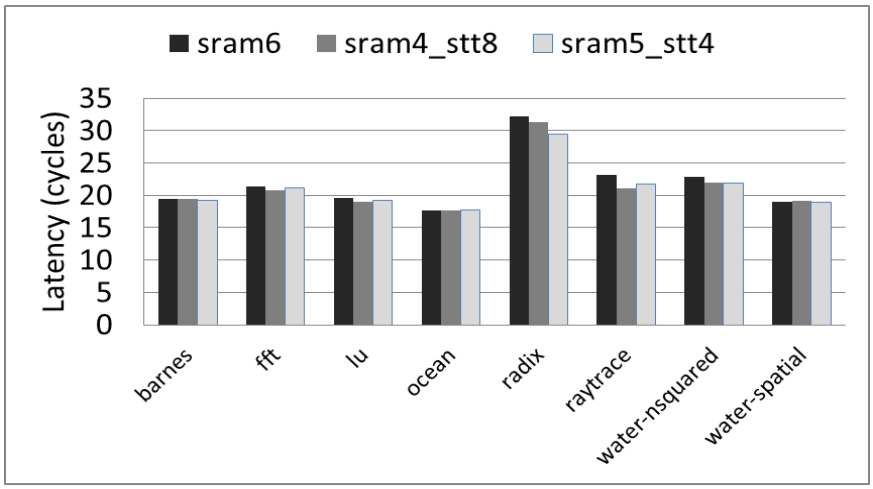


Figure 26. SPLASH-2 Benchmarks results with area budget SRAM 6

5.3 Power Analysis

Power consumption is one of the main issues to deal with for NoC designers; evaluation of power consumption is done on the proposed hybrid buffer mechanism based with both simple and lazy migration schemes described in section 5.3. Figure 27 compares the dynamic buffer power consumption of SRAM based buffer with four different migration schemes in SRAM3_STT12 hybrid configurations, all results are normalized to SRAM6 area budget. For lazy migration scheme, three thresholds, 0.25, 0.5 and 0.75 are used. It can be inferred from the graph that a lazy migration scheme with a threshold of 0.75 consumes 53% less power compared to naive migration scheme. As expected from lazy migration scheme, in a low load traffic of about 0.1 injection rate, power consumption of the propose hybrid buffer with 0.75 migration threshold is almost equivalent of a pure SRAM based buffer. But in a high network load of 0.4, flit migration scheme is constantly used and as a consequence there is substantial increase of about 170% in buffer power consumption due to two writes, once onto SRAM and other onto STT-MRAM.

However, when total router power consumption is compared as in Figure 28, lazy migration scheme with 0.75 as threshold consumes almost equivalent power compared to a pure SRAM based buffer router. In fact, for a low network load of 0.1, power consumption of proposed hybrid router is about 16% less compared to a pure SRAM based NoC router. But, in a high network load of 0.4 due to additional writes there is increased power consumption of only 4%. This tradeoff is justifiable for significant

increase in throughput. Lower consumption of router is ascertained due to almost zero leakage power of a STT-MRAM based NoC buffer.

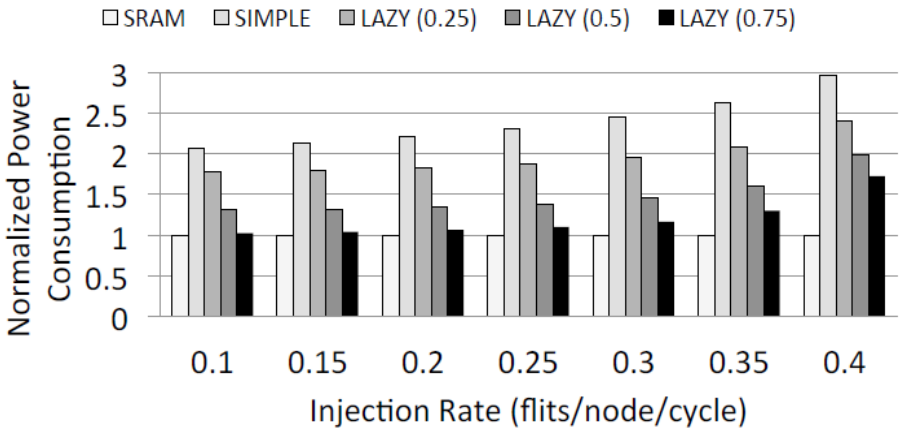


Figure 27.Dynamic Power Consumption of Input Buffers

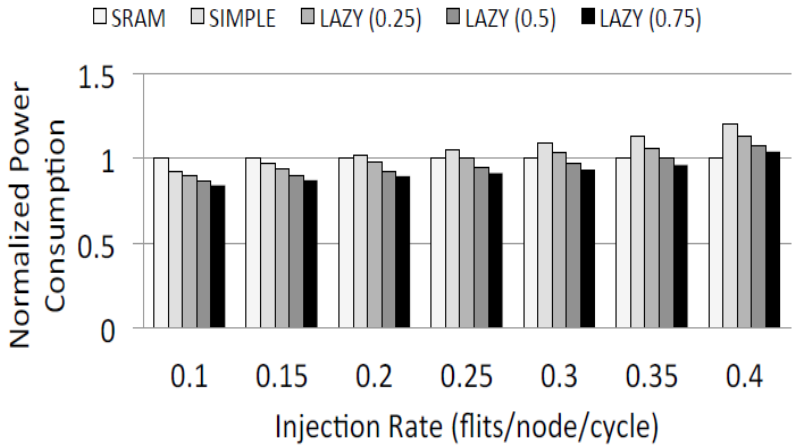


Figure 28. Total Router Power Consumption

It is observed that as threshold value is increased from 0.25 to 0.75 in lazy migration scheme, overall network throughput is slightly reduced: around 0.5% on an

average, which is negligible. Figure 29 shows the performance of SRAM3_STT16 configuration.

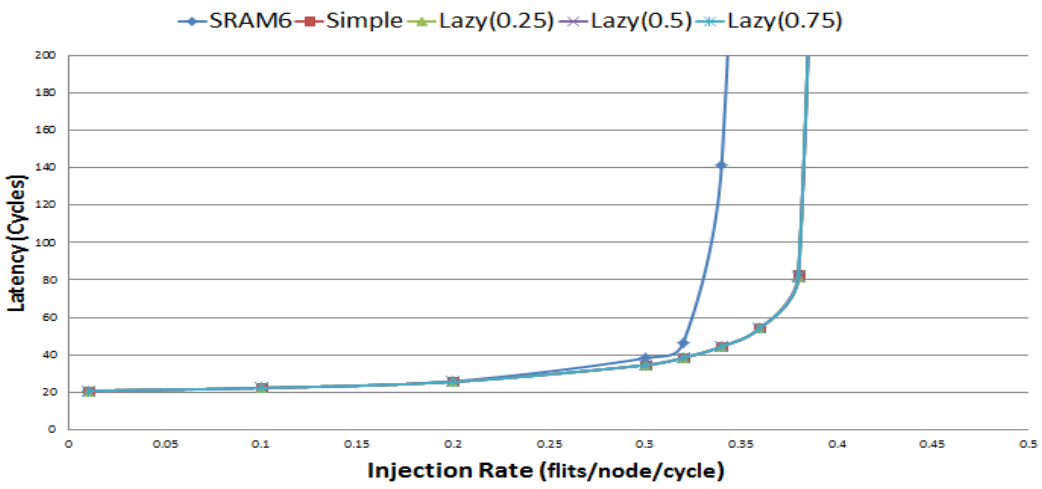


Figure 29. Performance comparison with different lazy thresholds

6. CONCLUSION

With Chip Multi Processors becoming reality and Network-on-chip being the only viable replacement to shared buses, throughput and power efficiency has become the major area of concern for NoC designers.

In this work a hybrid input buffer design using STT-MRAM with SRAM to achieve better network throughput with marginal power overheads in on-chip interconnection networks. The high density of STTMRAM facilitates to accommodate larger buffer compared to the conventional SRAM under the same area budgets. Through the flit migration schemes, the long write latency of STT-MRAM is effectively hidden while minimizing the power overheads. Simulation results indicate performance improvement of around 21% and 14% on average under the synthetic workloads and benchmarks, respectively, compared to the conventional on-chip router with the SRAM input buffer.

Future work includes reducing retention time further, developing STT-MRAM aware routing algorithms and to provide architectural level support to reduce overall power consumption.

REFERENCES

[1] A. Jog, A. K. Mishra, C. Xu, Y. Xie, N. Vijaykrishnan, R. Iyer, and C. R. Das, " Cache Revive: Architecting Volatile STT-RAM Caches for Enhanced Performance in CMPs," The Pennsylvania State University CSE Dept., Tech. Rep. CSE-11-010, June 2011.

[2] C. W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. R. Stan, "Relaxing Non-Volatility for Fast and Energy-Efficient STTRAM Caches," in *Proceedings of HPCA*, 2011.

[3] A. K. Mishra, X. Dong, G. Sun, Y. Xie, N. Vijaykrishnan, and C. R. Das, "Architecting On-Chip Interconnects for Stacked 3D STT-RAM Caches in CMPs," in *Proceedings of ISCA*, 2011.

[4] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen, "A Novel Architecture of the 3D Stacked MRAM L2 Cache for CMPs," in *Proceedings of HPCA*, 2009.

[5] X. Guo, E. Ipek, and T. Soyata, "Resistive Computation: Avoiding the Power Wall with Low-Leakage, STT-MRAM Based Computing," in *Proceedings of ISCA*, 2010.

[6] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "A Durable and Energy Efficient Main Memory Using Phase Change Memory Technology," in *Proceedings of ISCA*, 2009.

[7] M. K. Qureshi, V. Srinivasan, and J. A. Rivers, "Scalable High Performance Main Memory System Using Phase-Change Memory Technology," in *Proceedings of ISCA*, 2009.

[8] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger, "Architecting Phase Change Memory as a Scalable DRAM Alternative," in *Proceedings of ISCA*, 2009.

[9] M. K. Qureshi, M. M. Franceschini, and L. A. Lastras-montao, "Improving Read Performance of Phase Change Memories via Write Cancellation and Write Pausing," in *Proceedings of HPCA*, 2010.

[10] P. Zhou, Y. Du, Y. Zhang, and J. Yang, "Fine-Grained QoS Scheduling for PCM-based Main Memory Systems," in *Proceedings of IPDPS*, 2010.

[11] ITRS, "International Technology Roadmap for Semiconductors: 2009 Executive Summary," http://www.itrs.net/Links/2009ITRS/Home2009.htm.

[12] L.-S. Peh and W. J. Dally, "A Delay Model and Speculative Architecture for Pipelined Routers," in *Proceedings of HPCA*, 2001.

[13] M. Galles, "Scalable Pipelined Interconnect for Distributed EndpointRouting: The SGI SPIDER Chip," in *Proceedings of Hot Interconnect 4*, 2009.

[14] W. J. Dally and C. L. Seitz, "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks," *IEEE Trans. Comput.*, vol. 36, pp. 547–553, May 1987.

[15] W. J. Dally, "Virtual-Channel Flow Control," *IEEE Trans. Parallel Distrib. Syst.*, vol. 3, pp. 194–205, March 1992.

[16] A. V. Yakovlev, A. M. Koelmans, and L. Lavagno, "High-Level Modeling and Design of Asynchronous Interface Logic," *IEEE Design and Test of Computers*, vol. 12, pp. 32–40, 1995.

[17] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy Reduction for STTRAM Using Early Write Termination," in *Proceedings of ICCAD*, 2009.

[18] D. Seo, A. Ali, W.-T. Lim, N. Rafique, and M. Thottethodi, "Near- Optimal Worst-Case Throughput Routing for Two-Dimensional Mesh Networks," in *Proceedings of ISCA*, 2005.

[19] S.C.Woo, M.Ohara, E.Torrie, J.P.Singh, and A.Gupta, "The SPLASH- 2 Programs: Characterization and Methodological Considerations," in *Proceedings of ISCA*, 1995.

[20] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, B. Werner, and B. Werner, "Simics: A Full System Simulation Platform," *Computer*, vol. 35, no. 2, pp. 50–58, 2002.

[21] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi, "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration," in *Proceedings of DATE*, 2009.

[22] J. Kim, J. Balfour, and W. Dally, "Flattened Butterfly Topology for On-Chip Networks," in *Proceedings of MICRO, 2007.*

VITA

Name: Nikhil Vikram Kulkarni

Address: H.R. Bright Building, Room 335
College Station, Texas 77843-3112

Email Address: nikhilvk@tamu.edu

Education: B.E., Computer Science and Engineering, PESSE, Bangalore 2010
M.S.,Computer Science and Engineering, Texas A&M,University, 2012