

An Adaptive Deflection Router with Dual Injection and Ejection Units for Mesh NoCs

John Jose, Abhijit Das

MARS Research Lab, Dept. of CSE, Indian Institute of Technology Guwahati, India
{johnjose, abhijit.das}@iitg.ernet.in

Abstract—With the ever increasing core counts in Chip Multi-Processors (CMPs), Network-on-Chip (NoC) has emerged as a preferred framework for communication among various chip components. Among other factors, energy efficiency and congestion management play a vital role in identifying an efficient NoC design. Thus, NoCs with side-buffered deflection routers have gained popularity; mainly because of their simplicity in router design, low energy consumption and better load balancing capacity. Standing on the shoulders of existing state of the art, this paper proposes ADIEU; An Adaptive Deflection Router With Dual Injection and Ejection Units. This router has dual injection units and a minimal set of side-buffers to make adaptive routing decisions. Experimental results on the proposed microarchitecture using both real and synthetic workloads shows reduced average latency, buffer occupancy and deflection rate of flits when compared with the existing side-buffered deflection routers without any change in the critical path delay.

Keywords—Congestion, buffer-occupancy, side-buffer, starvation

I. INTRODUCTION

Advancing VLSI technology by decreasing feature sizes and shortening wire widths unmasked the constraints of the traditional bus based on-chip interconnection systems. Furthermore, this technology scaling has also increased the performance gap between computation and communication efficiency in modern SoCs. Apart from high speed computing cores, efficient and reliable communication is also essential for achieving high performance in multi-core systems. Network-on-Chip (NoC) is now an established framework that can efficiently support the integration of a massive number of cores on a chip by decoupling the on-chip computation and communication infrastructure, thus overcoming the scalability issues in conventional buses [1].

Input buffered routers dominated initial NoC designs due to their simple wormhole switching [1] and high load handling capacity. However, they consume a significant portion of chip power due to the presence of buffers. Studies show that approximately 30% to 40% of chip power is consumed by the NoC [2][3]. Thus, recent router designs are focusing on a short critical path and minimal buffer footprint [4][5].

Buffer-less routers are proposed as an energy efficient alternative to the traditional input buffered routers. Our simulations (Section V describes the experimental setup)

with real workloads on mesh NoCs with input buffered routers show that for low injection rate applications, in 90% cases, less than 25% of the buffers are being occupied, thereby exposing the over provisioning of buffers in routers. For low to medium injection rate applications, buffer-less NoC router is an optimal design choice [6][7].

Deflection routing is the most commonly used approach in buffer-less routers. When two flits (packets are broken down into multiple flits) that want to have the same output port reach a buffer-less deflection router, only one gets the requested port, and the other flit is deflected through an undesired port. The deflected flits eventually reach the destination by proper livelock prevention mechanisms. Buffer-less routers may not be a good design choice for high injection rate applications as the flits experience high deflection rate. Buffer-less routers that are equipped with side-buffers can accommodate some deflected flits thereby reducing the deflection rate [5][6].

In this paper, we address few critical limitations of the existing state-of-the-art side-buffered deflection routers through a proposed energy efficient Adaptive Deflection Router with Dual Injection and Ejection Units (ADIEU) that effectively handles network congestion leading to the reduction in average latency, buffer occupancy and deflection rate of flits.

II. BUFFER-LESS ROUTERS: RELATED WORK

As buffers in the NoC routers are power hungry and buffer management circuits are complex, buffer-less routers are gaining popularity on large mesh networks. Few works have also exploited a hybrid approach that uses a conventional buffered router with a provision to switch to buffer-less mode under low network load by using power gating techniques [8][9].

In buffer-less deflection routers, storage of flits happens only in pipeline registers. Buffering of flits that fail in getting the desired port is replaced by the concept of deflecting the flits [10] to non-productive ports. To avoid fragmentation, deflection routers employ flit level routing. In every cycle, maximum of four flits each can enter or leave a router. The incoming flits enter a routing unit, which computes the desired output port for the flit. After routing, if there is any flit destined to the local core, it is ejected. The port

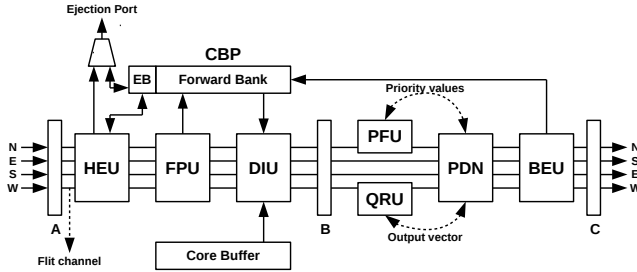


Fig. 1. Router pipeline for DeBAR. HEU-Hybrid Ejection Unit, FPU-Flit Preemption Unit, DIU-Dual Injection Unit, PFU-Priority Fixer Unit, QRU-Quadrant Routing Unit, PDN-Permutation Deflection Network, BEU-Buffer Ejection Unit, CBP-Central Buffer Pool.

allocator assigns output ports to all the flits present in the router. Flits which get the same output port as requested are called productively assigned flits, and the others are called deflected flits. BLESS [11] uses a crossbar with sequential output port allocation unit which increases the router critical path. This allocation unit is replaced by a Permutation Deflection Network (PDN) in CHIPPER [4], which considerably reduces the critical path delay at the expense of increased deflection rate.

Buffer-less deflection router suffers from performance degradation at high injection rate due to high deflection rate of flits. To address this issue, MinBD [5], DeBAR [6] and SLIDER [12] use a minimal set of side-buffers. Entry path to these side-buffers is kept after the PDN unit to accommodate a fraction of the deflected flits, thereby reducing the mis-routed flit traffic in the network. Minimally buffered routers outperform input buffered routers in low injection traffic and buffer-less deflection routers in high injection traffic.

DeBAR is the best available deflection router that proposed an effective solution by combining the merits of buffered and buffer-less routing. It has also successfully addressed the limitations of BLESS, CHIPPER and MinBD.

III. MOTIVATION

DeBAR is a 2-stage deflection router that uses a Central Buffer Pool (CBP) to accommodate a fraction of misrouted flits. The block diagram of DeBAR is shown in Fig. 1 where A, B, and C are the pipeline registers. Four internal flit channels carry input flits through various units of the router pipeline. The core-buffer contains newly created flits from the local processing core. We identify three performance limitations in the DeBAR design that motivated us for the proposed work. We analyse the cause for each of these limitations and suggest suitable cost-effective solutions.

A. Starvation of Side-Buffered Flits Due to Ineffective Priority Scheme

In DeBAR, flit priority is calculated based on the hops-to-destination of the flit from current router. The flit with least hops-to-destination is given the highest priority during port

allocation. Because of port conflicts, flits with low priority may be allocated non-productive output ports, leading to subsequent buffering in CBP by BEU. As the priority of flits buffered in CBP is not changing when such flits are re-injected into the network (because hops-to-destination of those flits are not changed), there is a high chance that they can be buffered again in CBP due to port conflicts. This leads to starvation of such flits and increases the average flit latency. Close to saturation load, for uniform traffic we identify 27% of such starvation cases (flits from CBP get back to CBP again due to low priority) upon using DeBAR for an 8×8 mesh NoC system. We propose that the flits that are buffered in CBP should get a higher priority when they are re-injected to ensure that they make forward progress.

B. Output Channel Wastage

In DeBAR, at saturation load, we observe that in 35% of cases at least one of the output ports of a router is idle while flits are waiting in CBP or core-buffer with those unused ports as their productive ports. This is due to the forwarding of misrouted flits to CBP after PDN by BEU. Flits waiting in CBP / core-buffer may not be able to inject to DIU if all four internal flit channels are busy. However, after PDN, due to side-buffering of a deflected flit in CBP the port already assigned to such a flit by PDN will be idle. Presence of such idle output channel is a wastage of resource. If an idle channel can be assigned to a flit that is waiting in either CBP or core-buffer, this channel wastage can be reduced.

C. Sequential Positioning of Independent Operations

HEU in DeBAR is functional only if there is a flit ejection. FPU aims at creating an idle slot in the internal flit channels of the router pipeline so that flits from core-buffer and CBP can be injected into these slots when they reach their starvation thresholds. A flit removed by HEU for ejection creates an idle slot in the internal flit channel. If HEU ejects a flit, FPU does not perform any flit removal as HEU has already created an idle slot. If all flit channels are busy and none of them are ejection flits, then HEU can be idle, and FPU has to pre-empt one flit. This means that only one of these two units are operational in a given cycle. Hence these two units can be combined to form a single unit to reduce hardware cost and critical path delay.

Experimental analysis on real and synthetic workloads have confirmed that the above-identified limitations of DeBAR create a critical performance bottleneck. In the proposed work, we modify the existing priority scheme to reduce the starvation of side-buffered flits. We also provide one more injection unit late in the pipeline thereby reducing the side-buffer/core-buffer occupancy of flits. This additional injection unit can also reduce channel wastage.

IV. ADIEU ARCHITECTURE

The basic working of ADIEU is similar to that of DeBAR except for few additional units that improve performance.

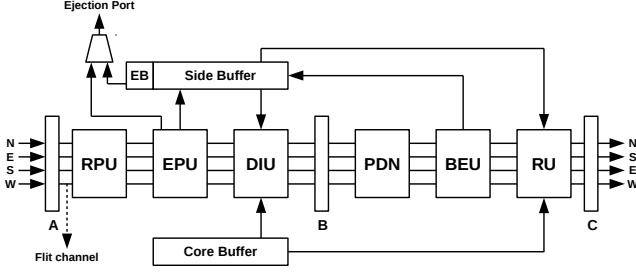


Fig. 2. Router pipeline of ADIEU. RPU-Routing and Priority unit, EPU-Ejection and Pre-emption unit, DIU-Dual Injection Unit, PDN-Permutation Deflection Network, BEU-Buffer Ejection Unit, RU-Re-injection Unit, EB-Ejection Bank.

Fig. 2 shows the block diagram of ADIEU. Like DeBAR, here also input flits are stored in a pipeline register, and a fraction of deflected flits are stored in side-buffers. ADIEU differs from DeBAR in the following aspects:

- The priority scheme is modified such that the re-injected flits to the DIU will get the highest priority and will not be side-buffered again on the same router. This reduces the side-buffer occupancy of flits thereby addressing the issue mentioned in Section III-A.
- A Re-injection Unit (RU) is included as the last unit in ADIEU pipeline to give chance for injecting flits that are waiting in the core/side-buffer. These flits are assigned productive output ports in idle output channels to address the issue mentioned in Section III-B.
- HEU and FPU of DeBAR are combined into a single unit called Ejection and Pre-emption Unit (EPU) to address the issue mentioned in Section III-C.
- Route and priority computations are done in the first stage of the pipeline (at RPU) to accommodate the new RU in the second stage.

The internal architecture and working of various units in ADIEU are discussed below.

A. Routing and Priority Unit (RPU)

RPU reads the destination information of all the incoming flits from pipeline register A. Based on the destination address of a flit, the desired output port is identified. We use dimension order routing algorithm [1] for identification of a productive output port. By this routing operation, the locally destined flits (ejection flits) are also identified. For the ejection flits, RPU sets an ejection flag in the flit header. Similarly, from the destination address of a flit, the hops-to-destination value is computed, which is considered as the priority of that flit. The 2-bit priority value (similar to the one as in DeBAR) is stored in the flit header itself.

B. Ejection and Pre-emption Unit (EPU)

EPU can act as an ejection unit or as a pre-emption unit based on the value of the ejection flag (already set/reset

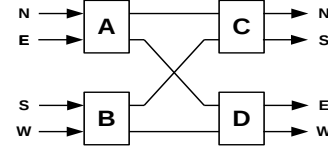


Fig. 3. Permutation Deflection Network (PDN).

by RPU) in the incoming flits. EPU consists of an ejection flag checking circuit and two parallel combinational blocks; one for ejection unit and other for pre-emption unit. If the ejection flag is set, EPU forwards the flit from router pipeline to the ejection port. Similar to DeBAR, EPU performs at most two flit ejections/cycle with the help of single ejection port and the Ejection Bank (EB) in the side-buffer.

If the ejection flag is not set, EPU acts like a flit pre-emption unit. It checks whether all internal flit channels are occupied and whether the starvation threshold is crossed or not. If so, EPU will pre-empt a flit from the router pipeline to the side-buffer. Similar to DeBAR, the starvation of flits waiting in the buffers are addressed by fixing a threshold to Re-Inject Interval (RII) for side-buffer and Core Inject Interval (CII) for core-buffer. By this flit pre-emption, EPU makes a free channel for buffer injection. We consider the threshold value of CII and RII as 2 cycles each.

C. Dual Injection Unit (DIU)

The basic working of DIU in ADIEU is same as that in DeBAR except for the priority variation of re-injected side-buffered flits. In DeBAR the priority of flits does not change even if they are re-injected from the CBP. We see that this could lead to unnecessary penalisation of flits entering the side-buffer. In ADIEU, the re-injected flits from the side-buffer are assigned the highest priority to ensure that the side-buffered flits are not penalised again on the same router. At the end of the first cycle, all the flits reach pipeline register B.

D. Permutation Deflection Network (PDN) and Buffer Ejection Unit (BEU)

PDN and BEU in ADIEU are same as that in DeBAR. PDN is a two-stage arbitration circuit that performs parallel allocation of output ports. Fig. 3 shows how the arbiter blocks (A, B, C, and D) are arranged to form a PDN. For each arbitration stage, the priority level and the desired output port of incoming flits (given by RPU) are used for determining the actual output port. The highest priority flit always gets its productive port. Other flits may or may not get a productive port depending on current port conflicts. From among the flits coming out from PDN, BEU selects at most one flit that is assigned a non-productive port for storing into the side-buffer. This side-buffering reduces average deflection rate, thereby bringing down the unwanted flit movements in the network.

Percentage Miss Rate	Benchmarks
Low MPKI (less than 5)	calculix, gobmk, gromacs, h264ref
Medium MPKI (between 5 and 25)	bwaves, bzip2, gamess, gcc
High MPKI (greater than 25)	hammer, lbm, mcf, leslie3d

Table I. Classification of benchmarks based on cache MPKIs

E. Re-injection Unit (RU)

We observe that only in less than 10% cases, all output ports of DeBAR are full. Under such conditions, to exploit the slot wastage, the newly added RU search among the buffered flits (which are there in side/core-buffers) to find if their desired output ports match with any idle output channels. If found, RU assigns respective idle output channels to each such flits. As in the case of DIU, side-buffer and core-buffer re-injections are given alternate priority in odd and even cycles, respectively to ensure fairness.

V. EXPERIMENTAL METHODOLOGY

We use a cycle-accurate simulator, BookSim 2.0 [13] for the NoC simulation. We modify BookSim to model two-cycle deflection router microarchitectures of MinBD, DeBAR and ADIEU for an 8×8 mesh network. We consider flits with necessary header information to facilitate independent routing as practised in standard deflection routers [5]. Necessary reassembly mechanism is employed for handling out-of-order delivery of flits. The flit channel is 140-bit wide: 128-bit data field and a 12-bit header field. We first consider synthetic workloads for the evaluation of our proposed router design. Average latency, buffer occupancy and deflection rate of flits are collected for each traffic pattern with injection rate varying from zero to saturation.

To evaluate our design with real workloads, SPEC CPU 2006 benchmarks are used, which are classified according to their Misses Per Kilo Instructions (MPKI) on a 64KB L1 cache as shown in Table I. This is to classify the applications to different network injection intensity groups. Based on this network injection intensity, we create 7 workload mixes (M_i s) consisting of SPEC CPU 2006 benchmarks as shown in Table II. Consider mix 1 (M_1); where out of 64 cores that we model, 16 cores run *calculix*, 16 cores run *gobmk*, 16 cores run *gromacs* and last 16 cores run *h264ref* benchmark. Similarly, other workload mixes ($M_2 - M_7$) can also be described.

We run 64 application instances of the respective workload mixes (as mentioned above) in gem5 simulator [14], which models a 64-core CMP setup with CPU cores and 2 levels of cache hierarchy. Each core consists of an out-of-order x86 processing unit with a 64KB, 4-way associative, 32B block, dual ported, unified, private L1 cache and a 32MB, 16-way associative, 64B block, shared distributed L2 cache (i.e. 512KB/core). We create a request packet for each L1 cache miss and feed it to BookSim to model the NoC traffic. Network statistics are collected and analysed. Each

Mix #	SPEC CPU 2006 Benchmarks					
M1	calculix(16)	gobmk(16)	gromacs(16)	h264ref(16)		
M2	bwaves(16)	bzip2(16)	gamess(16)	gcc(16)		
M3	hammer(16)	lbm(16)	mcf(16)	leslie3d(16)		
M4	calculix(16)	gobmk(16)	gamess(16)	gcc(16)		
M5	bwaves(16)	bzip2(16)	mcf(16)	leslie3d(16)		
M6	hammer(16)	lbm(16)	gromacs(16)	h264ref(16)		
M7	calculix(10)	gromacs(10)	bwaves(10)	gamess(10)	hammer(12)	mcf(12)

Table II. Various workload mixes

L1 cache miss creates a 1 flit request packet to the respective core where the shared distributed L2 cache is mapped. Then, the respective core responds with a 4 flit reply packet.

VI. EXPERIMENTAL ANALYSIS

We compare the performance of ADIEU with both DeBAR and MinBD routers, as they are considered the best in the available literature. An analysis is done on average latency, buffer occupancy and deflection rate for both synthetic and real workloads.

A. Effect on Average Flit Latency

Fig. 4 shows a set of injection rate vs average flit latency graphs for MinBD, DeBAR and the proposed ADIEU routers using synthetic traffic patterns. We can see that across all traffic patterns ADIEU shows either same or lower average flit latency than MinBD and DeBAR. Also across all traffic patterns, ADIEU saturates later than MinBD and DeBAR. This makes our proposed ADIEU a better design choice for high injection rate applications.

Fig. 7 shows percentage reduction in average flit latency of DeBAR and ADIEU with respect to MinBD for various SPEC CPU 2006 benchmark mixes. We can see that for all the mixes ADIEU shows a reduction in average flit latency than DeBAR. Significant reduction in latency can be seen for high injection rate mixes like M_3 and M_5 .

B. Effect on Average Buffer Occupancy

Buffer occupancy of a flit in side-buffered deflection routers refers to the number of cycles spent by a flit in side-buffers in its entire lifetime. It gives the waiting time of the flit in the side-buffers until it gets re-injected into the router pipeline. Average buffer occupancy, B_{occ} is given by,

$$B_{occ} = \frac{\sum_{i=1}^N b_i}{N} \quad (1)$$

where b_i is the total number of cycles a flit stays in side-buffers of all routers in its path to destination and N is the total number of injected flits.

In DeBAR, to reduce the deflection rate of flits, one of the flits that are assigned a non-productive port is moved to the side-buffer. If a buffered flit gets delayed in re-injecting into the router pipeline, it can increase average buffer occupancy. An increase in either deflection rate or buffer occupancy can lead to increase in the overall flit latency. Since we give the highest priority to the re-injected flits from side-buffer,

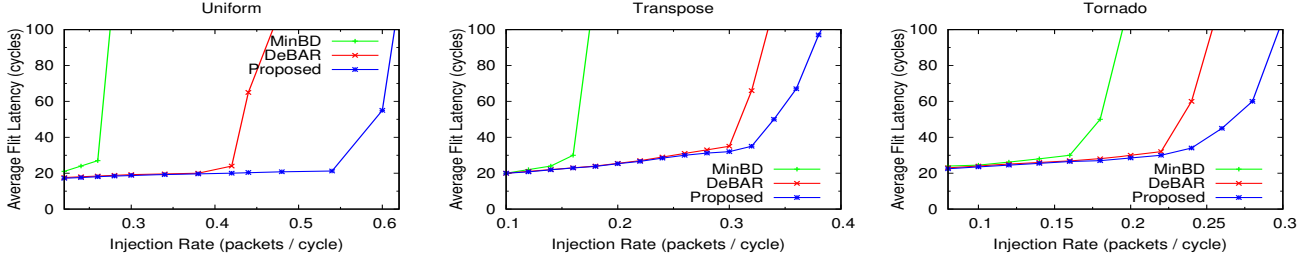


Fig. 4. Comparison of average flit latency for various synthetic traffic patterns in 8×8 mesh network.

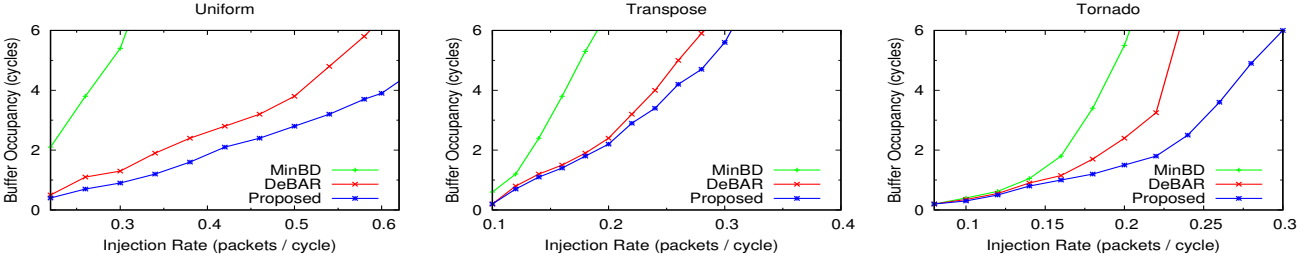


Fig. 5. Comparison of average flit buffer occupancy for various synthetic traffic patterns in 8×8 mesh network.

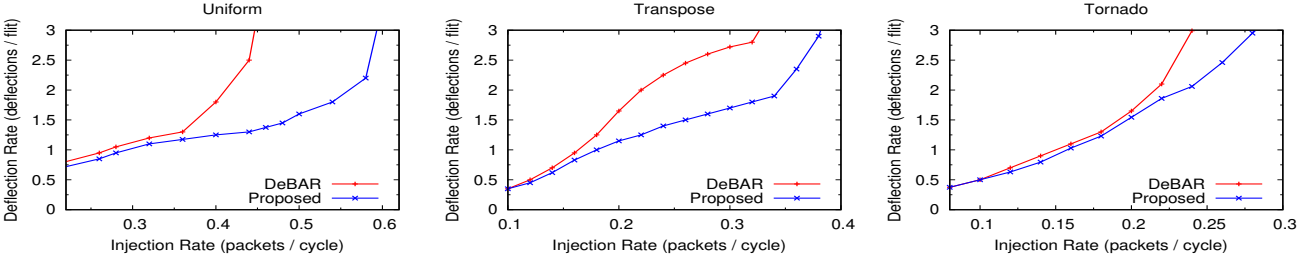


Fig. 6. Comparison of average flit deflection rate for various synthetic traffic patterns in 8×8 mesh network.

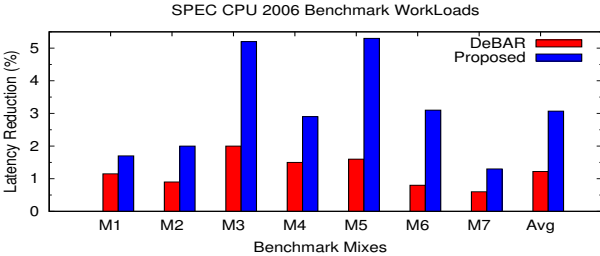


Fig. 7. Percentage reduction in average flit latency.

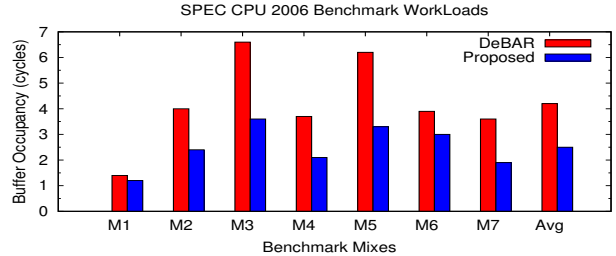


Fig. 8. Reduction in average flit buffer occupancy.

they will always get their desired output ports. Thus the re-injected flits will not go to side-buffer on the same router or get deflected away. Hence using ADIEU, we expect an overall reduction in the average buffer occupancy of flits.

Fig. 5 shows the buffer occupancy comparison for MinBD, DeBAR and ADIEU designs. At higher injection rate, ADIEU design has significantly lower buffer occupancy in all traffic patterns. This result shows the drawback in DeBAR design due to starvation of flits in side-buffer. By giving highest priority to the re-injected flits, we avoid this starvation scenario, thereby reducing flit latency.

Fig. 8 shows the average buffer occupancy of DeBAR

and ADIEU designs for SPEC CPU 2006 benchmark mixes. Since the respective values for MinBD is very high, we are not plotting them in Fig. 6, Fig. 8 and Fig. 9. Here we focus more on how much improvement we attain with respect to DeBAR design. We can see that for all the mixes ADIEU shows a reduction in buffer occupancy of flits than DeBAR. A significant difference in buffer occupancy can be seen at high injection rate mixes M_3 and M_5 .

C. Effect on Deflection Rate

Deflection rate is defined as the number of non-productive hops a flit takes on an average to reach its destination. Deflection rate comparison between DeBAR and ADIEU

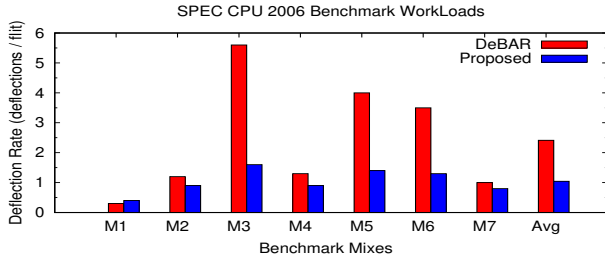


Fig. 9. Reduction in average flit deflection rate.

for synthetic traffic is shown in Fig. 6. It is clear that for all the traffic patterns, ADIEU achieves lower deflection rate as compared to DeBAR. The difference is more evident at higher injection rates. This is due to the better priority scheme used in ADIEU and the re-injection unit that increases the chance for flits to get their desired output port.

Fig. 9 shows the deflection rate of DeBAR and ADIEU designs for SPEC CPU 2006 benchmark mixes. We can see that for all the mixes (except M_1) ADIEU shows a reduction in deflection rate of flits than DeBAR. Lower the deflection rate; lower will be the network activity and hence lower dynamic power dissipation through the links.

Our simulations show that by using ADIEU router, there is a reduction of 11.5% in dynamic power with respect to DeBAR due to lower buffer occupancy and lower deflection rate.

D. Effect on Router Critical Path, Area, and Power

We implement DeBAR and ADIEU in Verilog and synthesise using Synopsys Design Compiler with 65nm cell library to obtain timing delay. In ADIEU, due to the removal of routing and priority units, and the addition of RU, the latency of stage 2 is unchanged with respect to DeBAR. The latency of stage 2 dominates over stage 1 in both DeBAR and ADIEU. We experimentally find that ADIEU can be operated at the same frequency as that of DeBAR.

We compute the area and power estimates of DeBAR and ADIEU using Orion 2.0 [15]. We assume 65nm technology for a NoC operating at 1GHz frequency with an inter-router link delay of 1 cycle. Due to the presence of RU and additional circuits in ADIEU, we incur an area overhead of 2.5% and a static power overhead of 3.8% with respect to DeBAR. Nevertheless, the performance gained with ADIEU is much more significant than this negligible overhead.

VII. CONCLUSION

By identifying the performance limitations in existing baseline models including MinBD and DeBAR, we proposed ADIEU, an adaptive deflection router microarchitecture with minimal side-buffering. ADIEU's superior design is based on enhancements proposed in primitive DeBAR design to improve overall system performance. The modification in priority scheme and the inclusion of an extra re-injection logic facilitate all possible opportunities for idle flits to

move out of the router. ADIEU microarchitecture stands above both MinBD and DeBAR, in terms of better overall average latency, buffer occupancy and deflection rate. All these enhancements and optimisations make ADIEU an ideal implementation choice for minimally buffered NoC routers.

ACKNOWLEDGMENT

This research is supported in part by Department of Science and Technology (DST), Government of India vide project grant ECR/2016/000212. The authors would like to thank R&D Section, IIT Guwahati for the SuG grant given to do this work.

REFERENCES

- [1] W. J. Dally and B. P. Towels, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004.
- [2] Y. Hoskote *et al.*, "A 5-GHz Mesh Interconnect for a Teraflops Processor," *IEEE Micro*, vol. 27, no. 5, pp. 51–61, 2007.
- [3] M. B. Taylor *et al.*, "Evaluation of the Raw Microprocessor: An Exposed-Wire-Delay Architecture for ILP and Streams," in *International Symposium on Computer Architecture (ISCA)*, pp. 2–13, 2004.
- [4] C. Fallin *et al.*, "CHIPPER: A Low-complexity Bufferless Deflection Router," in *International Symposium on High Performance Computer Architecture (HPCA)*, pp. 144–155, 2011.
- [5] C. Fallin *et al.*, "MinBD: Minimally-Buffered Deflection Routing for Energy-Efficient Interconnect," in *International Symposium on Networks-on-Chip (NOCS)*, pp. 1–10, 2012.
- [6] J. Jose *et al.*, "DeBAR: Deflection Based Adaptive Router With Minimal Buffering," in *Design, Automation and Test in Europe (DATE)*, pp. 1583–1588, 2013.
- [7] R. James *et al.*, "Smart Port Allocation for Adaptive NoC Routers," in *International Conference on VLSI Design (VLSI-D)*, pp. 475–480, 2015.
- [8] S. A. R. Jafri *et al.*, "Adaptive Flow Control for Robust Performance and Energy," in *International Symposium on Microarchitecture (MICRO)*, pp. 433–444, 2010.
- [9] G. Kim *et al.*, "FlexiBuffer: Reducing Leakage Power in On-Chip Network Routers," in *Design Automation Conference (DAC)*, pp. 936–941, 2011.
- [10] E. Nilsson *et al.*, "Load Distribution with the Proximity Congestion Awareness in a Network on Chip," in *Design, Automation and Test in Europe (DATE)*, pp. 1126–1127, 2003.
- [11] T. Moscibroda and O. Mutlu, "A Case for Bufferless Routing in On-Chip Networks," in *International Symposium on Computer Architecture (ISCA)*, pp. 196–207, 2009.
- [12] B. Nayak *et al.*, "SLIDER: Smart Late Injection Deflection Router for Mesh NoCs," in *International Conference on Computer Design (ICCD)*, pp. 377–383, 2013.
- [13] N. Jiang *et al.*, "A Detailed and Flexible Cycle-Accurate Network-on-Chip Simulator," in *International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 86–96, 2013.
- [14] N. Binkert *et al.*, "The gem5 Simulator," *SIGARCH Computer Architecture News (CAN)*, vol. 39, no. 2, pp. 1–7, 2011.
- [15] A. B. Kahng *et al.*, "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration," in *Design, Automation Test in Europe (DATE)*, pp. 423–428, 2009.