# Minimally Buffered Single-Cycle Deflection Router

Gnaneswara Rao Jonna, John Jose, Rachana Radhakrishnan, Madhu Mutyam
PACE Laboratory, Department of Computer Science and Engineering
Indian Institute of Technology Madras, Chennai, India 600036
Email: {jonna, johnjose, madhu}@cse.iitm.ac.in, rachana.rad@gmail.com

*Abstract*—**With the drift from computation centric designs to communication centric designs in the Chip Multi Processor (CMP) era, the interconnect fabric is gaining more importance. An efficient NoC in terms of power, area and average flit latency has a huge impact on the overall performance of a CMP. In the current work, we propose MinBSD - a minimally buffered, single cycle, deflection router. It incorporates different operations (Injection, Ejection, Preemption, Re-injection) in a single module to handle the traffic effectively and ensures smooth flow of flits through router pipeline. It performs overlapped execution of independent operations. These factors not only make MinBSD to operate in a single cycle but also to reduce the critical path latency resulting in a faster interconnect network. Experimental results show that MinBSD reduces the average flit latency on real work loads, reduces die area and power consumption when compared to the existing state-of-the-art minimally buffered deflection routers.**

## I. Introduction

With the scaling of number of on-chip cores, multihop interconnection networks are emerging as the most cost effective communication framework. Scalability, fault tolerance and better load handling capability make Network on Chip (NoC) a preferred choice over the bus-based inter-core communication system. Even though input buffered Virtual Channel (VC) based router design [1] is a common choice for NoC designs [2], the presence of power hungry buffers is affecting its applicability and scaling. This power inefficiency is a major bottleneck when the number of cores increases. Studies show that NoCs consume 30% to 40% of the chip power [3], [4].

Recent works propose new router models that use backpressure less routing techniques by completely eliminating buffers in the router input ports [5], [6], [7], [8]. As buffers are not used, network power consumption is greatly reduced. In bufferless or minimally buffered routers, port contentions are handled by deflection routing. With real workloads, it is observed that for low injection rate applications, the buffer occupancy is indeed low. When mesh NoCs are considered, for 90% of the time only 25% of the buffers are in use [8]. This indicates over-provisioning of the buffers. When the load on the network is low, due to minimum link contention, backpressure less routing performs well [6]. For low injection rate applications, bufferless router based NoC is an optimal design choice. Under high load these bufferless deflection routers [5], [6] incur significant deflections, leading to early saturation, rise in network activity and poor energy efficiency. Minimally buffered deflection routers address these issues in bufferless routers by buffering a fraction of deflected flits [7], [8].

In all the state-of-the-art minimally buffered routers, a 4x4 Permutation Deflection Network (PDN) is used for parallel port allocation. Separate modules with well coordinated inter-module handshaking are used for flit ejections, injections, preemption, side buffer ejection and reinjection. These features ensure the smooth flow of flits through the deflection router pipeline [7], [8]. Having separate logic modules for each of these operations makes the routers bulky and power hungry. The structural dependency of these modules makes the existing deflection routers to operate in two cycles [7], [8].

978-3-9815370-2-4/DATE14/©2014 EDAA

In the current work, we propose minimally buffered single cycle deflection router (**MinBSD**). It employs an innovative PDN that incorporates the functionalities of injection, preemption, reinjection and side buffer ejection, all in a single module. We show that this reduces the critical path latency of the router pipeline and enables it to operate in a single cycle at a higher frequency. Experimental results on 8x8 mesh network with synthetic traffic show that MinBSD outperforms MinBD [7] in terms of average flit latency but saturates earlier than DeBAR [8]. With real workloads, MinBSD outperforms DeBAR in terms of average flit latency and power consumption by 29% and 8.1%, respectively, on an average, while saving 12.7% of die area.

The remainder of this paper is organized as follows: Section II presents brief overview of works addressing deflection routers and single cycle routers. Section III provides the motivation for the current work. The details of the new router architecture, MinBSD, are dealt in Section IV. SectionV presents the experimental methodology, SectionVI provides the results and analysis and we conclude the paper in Section VII.

## II. Related work

The concept of bufferless deflection router architecture is discussed in [16] and a detailed implementation and analysis is presented in BLESS router [5]. Sequential port prioritization and allocation in BLESS is replaced by a parallel port allocation mechanism to reduce the critical path latency in CHIPPER [6]. Fallin et al. [7] propose to use side buffer to accommodate a fraction of deflected flits in the MinBD router, reducing the network activity and the delay in packet delivery significantly. DeBAR [8] outperforms MinBD in terms of average flit latency by introducing better priority mechanism, hybrid ejection, parallel execution of independent operations.

While there have been prior works on single cycle router architecture, their design methodologies and targets are different. Nguyen et al. [14] propose a speculation based single cycle buffered router architecture. Amit Kumar et al. [15] propose a non-speculation based single cycle buffered router pipeline. Hayenga et al. [13] propose a bufferless adaptive single cycle router architecture, namely, SCARAB. Few router architectures are based on combination of both buffered and bufferless routing [17], [18]. While these techniques make the design power efficient, area requirements remain the same.

## III. Motivation

With the existing minimally buffered router architectures, we observe that a flit spends two cycles in a router for every traversal between the routers in its lifetime apart from the buffered cycles. We target to *reduce the lifetime of the flit by reducing the router delay*. Further, reducing the critical path latency will enable us to operate our design faster. In all the router designs, cycle time is limited by the critical path. We target to *reduce the critical path latency* of the router.

All the two cycle routers need three levels of pipeline registers per router, which add to the critical path latency, also consume significant power and area. We target to *reduce the levels of pipeline registers* to two. Inclusion of more logic blocks makes the current router architectures smarter than their predecessors, but make them highly complex. As a result they consume more area and power. We target *to make the router architecture less complex, area and power efficient*.

In this work, we propose a single cycle router, namely, MinBSD, which meets all the above targets without compromising on performance.

## IV. MinBSD : Minimally Buffered Single cycle Deflection Router

Fig. 1 shows the MinBSD pipeline architecture. The detailed operation of each unit is explained below:

***Routing Unit (RU)*:** RU computes the productive port for each incoming flit. It extracts the destination (Dx, Dy) address field from each flit. Compares it with current node (Rx, Ry) address and determines the productive port.
If {Dx = Rx and Dy $\neq$ Ry} or {Dy = Ry and Dx $\neq$ Ry}, the productive port is along X or Y direction, respectively.
If {Dx = Rx and Dy = Ry}, it is locally destined and an eject flag is set, which is used by the Eject Unit later in the pipeline.
Else,{{if Xdiff (Dx-Rx) $>$ Ydiff (Dy-Ry) then the productive port is along the X direction, else it is along the Y direction}. If the productive port is along the X direction and Xdiff $>$ 0, then it is East else West. Similarly for Y direction, it is North or South}. The flit may get the productive port assigned or deflected, which is decided in PDN unit. This information is local to each router and reset when the flit leaves the router.

***Prioritization Unit (PU)*:** PU performs the priority computation for each incoming flit based on hops-to-destination, which ensures fairness and progress in flit movement. It extracts the destination address field from each flit, determines hops-to-destination (|Dx-Rx| + |Dy-Ry|) and assigns a priority value. We consider Priority 0 - at most 2 hops to destination; Priority 1 - at least 3 hops and at most 4 hops to destination; Priority 2 - at least 5 hops to destination. Flits closer to their destination are given higher priority.

***Permutation Deflection Network (PDN)*:** In MinBSD, we replace the conventional 4x4 PDN used in MinBD and DeBAR with a 6x6 PDN. It is used for parallel allocation of the ports. Depending on the location of a router in the network, we design three PDNs as shown in Fig. 2: Edge PDN, Corner PDN and Center PDN for edge routers, corner routers and all other remaining routers in the network, respectively.

*Center PDN* has 6 arbiters (L1, L2, L3, R1, R2, R3) arranged in two stages as 3x2 arbiters with 6 inputs and 6 outputs as shown in Fig. 2a. Out of 6 inputs, 4 inputs are from the neighbors, one input is from the Core Buffer (CB) and the other input is from the Side Buffer (SB). Out of 6 outputs, 4 outputs go to the neighbors, one output goes to the Splitter (SP) and the other output goes to the SB.

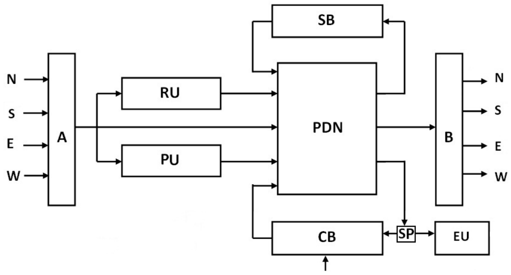


Fig. 1: MinBSD Pipeline Architecture. A and B - pipeline registers, RU - Routing Unit, PU - Prioritization Unit, SB - Side Buffer, CB - Core Buffer, PDN - Permutation Deflection Network, SP - Splitter, EU - Eject Unit.

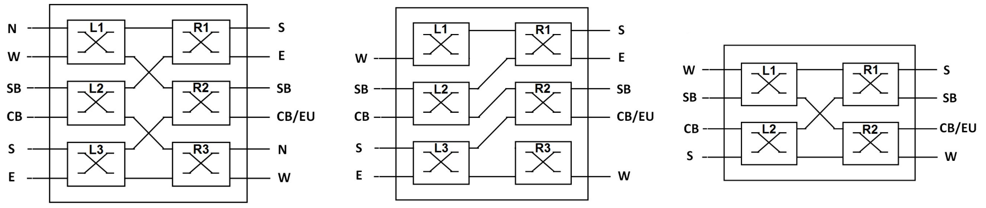
PDN uses the priority and productive port information of flits computed by PU and RU. At each arbiter the highest priority flit is assigned to its productive port and the other flit is assigned to the left-out port. Hence the highest priority flit (i.e., with the least number of hops to destination) always gets the desired port. SB and CB are directly connected to arbiter L2 as shown in Fig. 2a, so they can inject flits in every cycle. A flit reaching SP with eject flag as 0 is directed to CB and given highest priority in the next cycle.

We now illustrate the working of PDN using an example. Consider a flit coming from North with the desired port as South. If it wins at both L1 and R1, it gets South port (refer to Fig. 2a). If it wins at L1 but loses at R1, it may get South port or deflected to East port based on the desired port of the other contender at R1. If it loses at L1 but wins at R2, it is buffered in SB and reinjected from SB in the next cycle. If it loses at both L1 and R2, it may be buffered in SB or CB based on the desired port of the other contender at R2, and reinjected in the next cycle. Observe that all the input-output port pairs do not have a direct path. Instead, some have to be buffered in-between. There is no path between buffer to buffer, which avoids buffer to buffer flit movement cycles.

*Edge PDN* has 6 arbiters arranged in two stages as 3x2 arbiters but with 5 inputs and 5 outputs as shown in Fig. 2b. We inject flits from CB in odd cycles and from SB in even cycles into arbiter L2. In any cycle only one input is available at arbiter L2. The input at arbiter L2 reaches arbiter R2 if the flit has to be ejected (with highest priority) else it reaches arbiter R1, which prevents buffer to buffer flit movement cycles.

*Corner PDN* has 4 arbiters arranged in two stages as 2x2 arbiters with 4 inputs and 4 outputs as shown in Fig. 2c. We inject flits from CB in odd cycles and from SB in even cycles into arbiters L2 and L1, respectively, which prevents buffer to buffer flit movement cycles (when a flit other than locally injected ones reaches CB, it is given the highest priority in the next cycle that avoids CB to SB movement. So there is a SB to CB movement but no CB to SB movement).

From the working of PDN, we can observe that there is no possibility of cyclic movement of flits within the router that avoids internal deadlock and live-lock in MinBSD. While moving from one router to another, a flit either gets its desired



(a) Center PDN

(b) Edge PDN - North

(c) Corner PDN - North East

Fig. 2: PDN Architecture.

port or deflected, which avoids cyclic dependency stalls among the routers. Our prioritization mechanism ensures that there is no cyclic movement of flits among the routers. Hence, MinBSD is both deadlock free and live-lock free.

***Side Buffer (SB) and Core Buffer (CB)*:** We have two buffer segments in MinBSD, Side Buffer (SB) and Core Buffer (CB). SB is used to store the preempted flits out of the arbiter R2. Flits in SB are injected in every cycle into PDN. CB is used to store the flits from the local node and the flits through the eject channel of arbiter R2 whose eject flag is not set. Flits in CB are injected in every cycle into PDN. Though increase in the size of buffer segments yields marginal performance improvement, it occupies more area and consumes more power. We keep the size of buffer segments same as that of DeBAR [8] for evaluation purposes. No flit is buffered in a router more than once in consecutive cycles because our design has no direct path from buffer to buffer (except in the Corner PDN).

***Splitter (SP)*:** SP forwards any flit out of the eject channel of arbiter R2 in PDN either to Eject Unit (EU) or to CB based on whether or not the eject flag is set.

***Ejection Unit (EU)*:** EU forwards a flit received to the eject port. It has only one ejection port per router, unlike DeBAR and MinBD, where dual ejection ports are considered. We prefer single ejection over dual ejection as dual ejection is required in less than 12% of the cases on an average [8].

The major aspects of MinBSD are *parallel execution* and *simple design*. Overlapping of operations results in reduced critical path latency. We perform desired port computation and prioritization of flits in parallel. We also perform ejection of a flit in parallel with link traversal of flits. We directly inject flits in every cycle from CB and SB into PDN and buffer back the lesser priority flits in every cycle directly. This avoids the need for a separate injection unit, preemption unit, reinjection unit and buffer ejection unit when compared to MinBD and DeBAR. All these design optimizations make our architecture simpler, faster, low power consuming and area efficient when compared to earlier techniques.

## V. Experimental Methodology

We model the two-cycle deflection router micro architecture explained in CHIPPER by modifying the VC based NoC simulator Booksim [2]. Every flit has the routing information attached in the header, which allows independent routing of flits within a packet and proper reassembly mechanism is used for handling out-of-order delivered flits (it is the common standard in deflection routers). The flit channel is 140-bit wide: 128-bit data field and 12-bit header field. We modify this baseline deflection router architecture to model MinBD, DeBAR and MinBSD for performing experimental analysis.

We use Multi2Sim [9] to model the necessary setup to work with real workloads. We develop the Verilog model for our architecture and synthesize using Synopsys Design Compiler with 65nm CMOS library to compute the pipeline latency, area and power. We compare the performance of MinBSD with DeBAR and MinBD based on unit time instead of cycles. The cycle times are different (MinBSD:DeBAR = 3:4 = 1.5ns:2ns) because the critical path delay of MinBSD is 26% less than that of DeBAR and MinBD.

***Real Workloads*:** We analyze the performance of MinBSD in comparision with DeBAR using real workloads (multiprogrammed- SPEC CPU2006 [10] benchmark mixes and multithreaded SPLASH2 [11], PARSEC2.1 [12] benchmarks). We use Multi2Sim to model 64-core CMP setup (64 cores, each core has an out-of-order x86 processing unit with 64KB, 4-way set associative, 32 byte block, private L1 cache and a 512KB, 16-way set associative, 64 byte block, shared distributed L2 cache). We consider similar setup with slight modifications to run multithreaded workloads to generate enough traffic (64-core CMP with 60 processing cores with private L1 cache and 4 shared distributed L2 cache cores at the 4 corners of the network). We run 60 threads of each benchmark at a time, one thread on each processor core. After fast forwarding sufficiently, we collect the network events and provide this as traffic to the NoC simulator.

***Synthetic traffic*:** We analyze performance of MinBSD against DeBAR and MinBD using standard synthetic traffic patterns: *uniform, transpose, tornado, bit-complement, bit-reverse, shuffle and neighbor*. For evaluation purpose, we consider 8x8 mesh network. We collect the average latency values by varying the injection rate from zero to saturation.

## VI. Results and Analysis

***Average flit latency*:** Fig. 3 shows average flit latency of MinBSD, MinBD and DeBAR routers for different synthetic traffic patterns on 8x8 mesh network. From all the graphs, we can observe that MinBSD performs better than MinBD in all the cases and better than DeBAR at low to medium injection rates. MinBSD has low average flit latency than DeBAR and MinBD at low to medium injection rates. It extends the saturation injection with respect to MinBD but saturates earlier than DeBAR. This is because, at low to medium injection rates the contention at each arbiter in PDN is low and the percentage of flits buffered is also low, as a result we gain the advantage of single cycle router at low and medium injection rates. As the injection rate increases, more flits are injected into the network, which in-turn increase contention at each arbiter in PDN and the flits are forced to buffer in every cycle, leading to high congestion and early saturation. In case of DeBAR, flits are ejected, preempted, injected and buffered selectively, so that congestion reduces and injection saturation occurs lately.

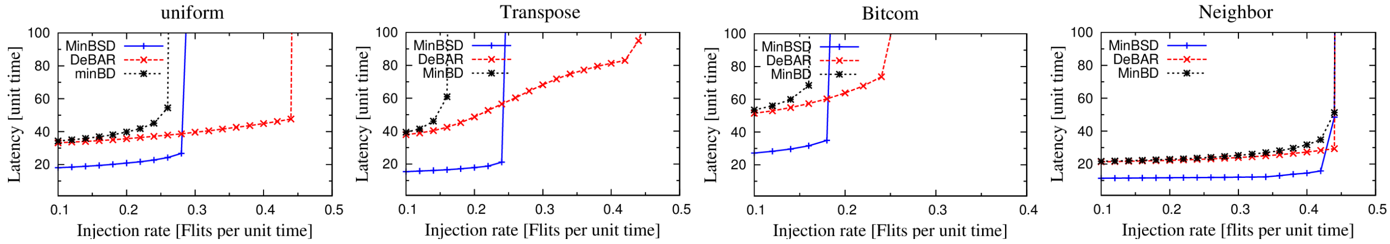


Fig. 3: Latency comparison under various synthetic traffic patterns on 8x8 mesh network.

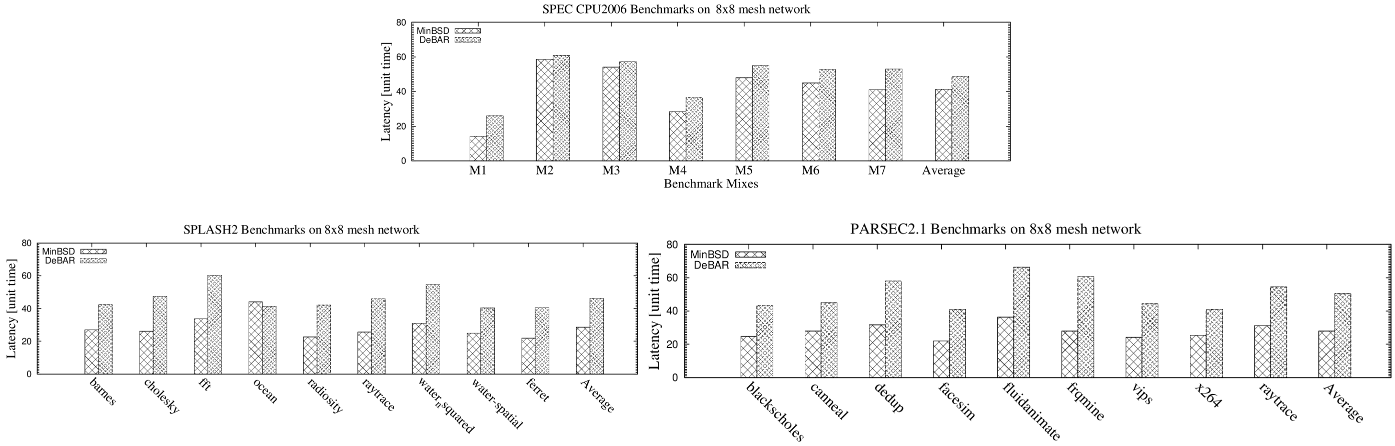


Fig. 4: Latency comparison under various benchmarks on 8x8 mesh network.

Fig. 4 shows average latency of MinBSD with DeBAR for different workloads from SPEC CPU2006, PARSEC2.1 and SPLASH2. From the graphs, we can observe that MinBSD outperforms DeBAR in all the cases. Past analysis [8] indicate that for real workloads DeBAR outperforms MinBD, so in our work we compare our technique with DeBAR alone. As real benchmark workloads have low to medium injection rates, few number of flits are buffered or deflected. Compared to DeBAR, our single-cycle router reduces average flit latency by 29%.

***Power, Area and Critical path latency***: The design optimizations employed in MinBSD enable us to develop a simple, low complex, single cycle router that runs at a higher speed. In MinBSD the critical path latency is reduced by 26%, the total power consumption is reduced by 8.1% and the die area required is reduced by 12.7% when compared to DeBAR.

## VII. Conclusion

We proposed a novel router architecture MinBSD for mesh NoCs. It replaces traditional two-cycle deflection router architecture with simple, low complex, high speed (reduced cycle time), single cycle router architecture. Experimental results showed that MinBSD is area, power and performance (average flit latency on real workloads) efficient. In general, real world applications (represented by real workloads) have low to medium injection rates. To handle them efficiently we do not need a highly complex router architecture. Rather a simple router architecture like MinBSD with area, power and performance benefits will be sufficient.

## References

[1] W.Dally , Virtual-Channel Flow Control, *IEEE Transactions on Parallel and Distributed Systems*: vol.3, pp. 194–205, 1992.

[2] W.Dally and B.Towles, *Principles and Practices of Interconnection Networks*: Morgan Kaufmann Publishers Inc., USA, 2003.

[3] Y.Hoskote et al., A 5-GHz Mesh Interconnect for a Teraflops Processor: *IEEE Micro*, vol. 27, no. 5, pp. 51–61, 2007.

[4] M.B.Taylor et al., Evaluation of the Raw Microprocessor: An Exposed-Wire-Delay Architecture for ILP and Streams: *ISCA*, 2004.

[5] T.Moscibroda and O.Mutlu, A case for bufferless routing in on-chip networks: *ISCA*, pp. 196–207, 2009.

[6] C. Fallin et al., CHIPPER: A Low complexity Bufferless Deflection Router: *HPCA*, pp. 144–155, 2011.

[7] C. Fallin et al., MinBD: Minimally-buffered deflection routing for energy-efficient interconnect: *NOCS*, pp. 1–10, 2012.

[8] JohnJose et al., DeBAR: Deflection Based Adaptive Router With Minimal Buffering: *DATE13,* pp. 1583–1588, 2013.

[9] R.Ubal et al., Multi2sim: A simulation framework to evaluate multicore-multithreaded processors: *SBAC-PAD,* pp. 62–68, 2007.

[10] SPEC2006 CPU benchmark suite: http://www.spec.org.

[11] S.C.Woo et al., The splash-2 programs: characterization and methodological considerations: *ISCA,* pp. 24–36, 1995.

[12] C.Bienia et al., The parsec benchmark suite: characterization and architectural implications: *PACT,* pp. 72–81, 2008.

[13] M.Hayenga et al., SCARAB: A single cycle adaptive routing and bufferless network: *MICRO,* pp. 244–254, 2009.

[14] ST Nguyen et al., A Low Cost Single-Cycle Router Based on Virtual Output Queuing for On-chip Networks: *DSD,* pp. 60–67, 2010.

[15] A Kumar et al., A 4.6Tbits/s 3.6GHz single-cycle NoC router with a novel switch allocator in 65nm CMOS: *ICCD,* pp. 63–70, 2007.

[16] E. Nilsson et al., Load distribution with the proximity congestion awareness in a network-on-chip: *DATE,* pp. 1126–1127, 2003.

[17] S.A.R.Jafri et al., Adaptive flow control for robust performance and energy: *MICRO,* pp. 433–444, 2010.

[18] G. Kim et al., Flexibuffer : Reducing leakage power in on-chip network routers: *DAC,* pp. 936–941, 2011.