

# HiPAD : High Performance Adaptive Deflection Router for On Chip Mesh Networks

Simi Zerine Sleeba\*, John Jose<sup>†</sup>, Mini M.G.\*

\*Government Model Engineering College, Kochi, India

<sup>†</sup>Rajagiri School of Engineering and Technology, Kochi, India

Email: simi@mec.ac.in, johnj@rajagiritech.ac.in, mininair@mec.ac.in

**Abstract**—The efficiency of a router in a Network on Chip (NoC) is characterised by good performance, minimum packet latency, area and power. In this paper, we propose an adaptive deflection router for mesh NoC which offers higher speed of operation by reducing the router critical path latency. We propose a single cycle router that uses an intelligent decision making logic to store deflected flits in minimum number of side buffers. Synthesis results of the design show an overall reduction in timing latency compared to a conventional minimally buffered router with two cycle latency. Network simulation of the proposed architecture using synthetic and real application traffic reports that the average flit latency and deflection rate reduces significantly in our design compared to the state-of-the-art single cycle deflection routers.

**Keywords**—Network on Chip; Deflection routing; Single cycle ; Latency reduction

## I. INTRODUCTION

With the growth of computation-intensive applications and the need of low-power, high-performance systems, the number of computing cores in a single-chip has enormously increased. The issues associated with global on-chip wiring are tackled by replacing them with scalable packet switched Networks on Chips(NoC) which offer higher communication bandwidth. In a conventional mesh NoC, each processing core is associated with a switching element called router. Routers are interconnected using links in a well structured topology. The basic data unit in NoC is a packet which is divided into flow control units called flits. The router receives flits coming from north, east, south and west directions and also from the processing element. Cache misses and coherence transactions cause traffic to be initiated into the network. Flits are introduced into the network from the local processing core by the injection process and flits are removed from the network into its destined processing core by the ejection process.

The buffers present in virtual channel(VC) routers consume large amount of power, even though they deliver higher throughput [1]. Researches to improve the energy-efficiency of routers come up with buffer-less and minimally buffered deflection routers which reduce both dynamic and static power. In buffer-less deflection routers, flits which do not get the desired output port get deflected through a freely available output port. At high intensity workloads, bandwidth of buffer-less NoCs shrink substantially due to increased flit deflections leading to degradation in application performance. Performance and

bandwidth of deflection routers can be improved by using a minimum number of buffers in every router.

## II. BACKGROUND AND MOTIVATION

BLESS [2] and CHIPPER [3] are two cycle bufferless deflection routers which offer good performance under low and medium network load. BLESS reports that power consumption decreases by 20-40%, router area on die is reduced by 75%, and implementation complexity also decreases compared to VC routers [2]. MinBD [4] is a minimally buffered router which uses a small side buffer at the output stage to store few of the deflected flits. By this technique, flit deflections are considerably reduced which improves the network performance without significantly affecting power and area. Deflection Based Adaptive Router(DeBAR) [5], uses a minimal central buffer pool to store some of the misrouted flits and exhibits promising results compared to MinBD. In both MinBD and DeBAR router pipelines, flits are injected from core buffer and re-injected from side buffer before output port allocation. This sometimes results in injected flits moving from core buffer to side buffer.

In single cycle deflection routers, the cycle duration is greater than that of two cycle routers, but it enables higher operating frequency for the NoC. In MinBSD (Minimally Buffered Single Cycle Deflection) router, parallelisation of independent router operations like route computation, ejection and flit prioritisation reduce the delay inside router [6]. MinBSD uses a two stage Permutation Deflection Network (PDN), each stage having three permuter blocks to connect input ports and side buffers to output ports. The two side buffers store a minimum number of misrouted flits and one among them also stores flits generated from the local core. Injection from these buffers takes place in all cycles. The 3x2 arbiter reduces the PDN delay significantly; yet it has a few major drawbacks which gives us motivation for further improving the architecture of single cycle deflection routers.

### A. Problem of structural isolation

As we can see from Figure 1, flits are not always free to move to their desired direction in a single cycle because all input ports are not structurally connected to all output ports. Due to this problem of structural isolation, many of the flits are sent to side buffer or core buffer instead of its desired output port. It occurs in cases when a flit needs to take a left or right turn. Consider a flit coming

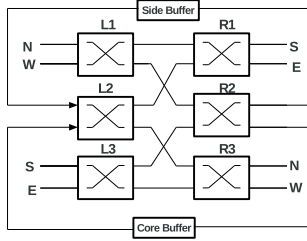


Figure 1: PDN structure of MinBSD [6]

from the north input link, and wishes to move to the west direction. In the first level of arbiters, the flit chooses the second link of the L1 arbiter and reaches the input of the R2 arbiter. Then it finally moves out of the output links either into the core buffer or side buffer. In the next cycle the flit again gets injected into the same router through the L2 arbiter of the first stage and moves into the R3 arbiter in the second stage.

### B. Delay due to extra arbiter stage

The flits that are to be buffered need to go through two levels of arbiters. This causes an extra delay in router, which in turn reduces the running frequency of the network. If it is possible to select the flits to be buffered before they enter the PDN stage, such flits can be moved out into the buffers directly, rather than arbitrating through the permutation network.

### C. Penalisation of high priority flits

In each arbiter of the PDN, flit with higher priority is given preference in the choice of output port. As shown in Figure 1, there are no direct paths connecting the L1 and R3 arbiters. Similarly there are no paths between L3 and R1 arbiters. Hence a high priority flit which chooses a desired output in the first stage of the PDN may end up in a side buffer or core buffer. As explained in the previous example, such flits are penalised and they are delayed by an additional cycle before re-entering into the PDN, inspite of having high priority. We experimentally analysed that about 23% of the high priority flits targeting for a desired output port connected to R1 or R3 arbiters, gets buffered by moving into the R2 arbiter of the second stage.

Latency of a flit through the network in cycles is given by the expression,

$$L = CBT + SBOT + HOPS$$

where CBT- Wait time in Core Buffer before Injection, SBOT- Side Buffer Occupancy Time and HOPS - Total number of hops from source to destination (some of the hops will be deflections). By optimising each of these terms, flit latency can be reduced and higher performance can be achieved. We simulate an 8x8 mesh network using MinBSD routers and fraction the flits into various bins based on the values of SBOT and overall latency. We observe that SBOT of 43% of the flits in uniform traffic is 1 or zero cycles, 30% of the flits is 2 to 3 cycles and rest

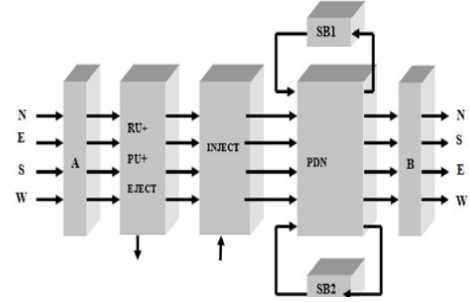


Figure 2: HiPAD router architecture

of the flits consume more time in side buffers. Because of this, the total latency of majority of flits is also greater than 10 cycles. We understand that the problem is due to the structural isolation problem discussed earlier.

In this paper, we propose HiPAD, a high performance adaptive deflection router with a structurally connected PDN which avoids unnecessary buffering of high priority flits. It also incorporates an intelligent decision making logic prior to the PDN which eliminates some flit deflections by buffering them in advance.

## III. THE PROPOSED ROUTER

The architecture of the proposed single cycle HiPAD router is shown in Figure 2. We briefly explain each of the functional units used. Initially the flits undergo routing, ejection and prioritization. These units receive flits arriving at the router through pipeline register A. Routing unit does the route computation for incoming flits based on the position of the source and destination router in the mesh. This states the possible directions a flit can move to reach its destination. Ejection unit supports one ejection port per router. In case of multiple flits to be ejected in a router, one among them is selected randomly for ejection. Others continue to move in the router pipeline. Prioritization unit prioritizes the flits based on the number of hops remaining to reach the destination. Flit with least number of hops to destination is given highest priority. After these functions are complete, flits are forwarded to the Injection unit that injects flits from core buffer into the network through a free input channel. Further, the PDN arbitrates the flits and forwards them to the output links of the router. The PDN consists of two stages each stage having two 2x2 crossbars. It allocates the output ports to the flits using the information obtained from routing and prioritization units. When the flits reach the port allocator unit, the highest priority flit is always forwarded to the desired output port whereas other flits get deflected through the remaining port.

The Permutation Deflection Network mainly has two parts: the decision making units and the port allocator as shown in Figure 3. The decision making unit decides which flits are to be injected into the port allocator. The decisions are taken based on the rules given in Algorithm 1.

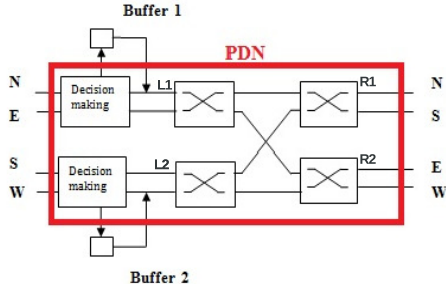


Figure 3: PDN structure of HiPAD

The main contributions in this router architecture are:

**No structural isolation:** The 2x2 arbiter stages used in the PDN addresses the structural drawback of MinBSD. In HiPAD, a flit that comes through the input link is free to move in any output direction by passing through the two stages of the permutation network. There are direct paths among all the arbiters as shown in Figure 3. Since the flits with higher priority can reach their desired direction without being buffered; the overall latency of the network reduces significantly.

**Parallelisation of Independent Operations:** The routing, ejection and prioritization operations are independent of each other. Hence these three operations can be performed in parallel which reduces the critical path delay inside the router.

**Injection from side buffers in all cycles:** The flits stored in the side buffers are injected in every cycle into the PDN. This prevents the starvation of flits in side buffers. When buffered flits are re-injected into the PDN, they are given higher priority than flits coming from input links. Thus it is ensured that the flits which have already undergone buffering do not suffer anymore. At a time six flits are taken care of at the input of PDN, 4 flits from the coming from the input links and 2 flits coming from the side buffers. Since injection from buffers occur in every cycles, only a minimum side-buffer size is required. We choose each side buffer to be of size one. Thus this method makes the proposed architecture more energy and area efficient.

**Single cycle operation:** The entire operation within the router takes place in a single cycle. This reduces the flit latency in terms of number of cycles. Since all the operations are performed in a single cycle, the router delay increases slightly. Hence the operating frequency of the network using this algorithm will be lesser than its counterparts having a two cycle operation.

#### IV. EXPERIMENTAL METHODOLOGY

We modify the traditional VC based Booksim [7] to model the HiPAD router as mentioned in Section 2 and compare its performance with that of MinBSD and MinBD. We consider 1-flit packets. Since deflection routers route each flit independently, every flit contains necessary control information needed for routing. The flit channel is 140-bit wide: 128-bit data field and 12-

---

#### Algorithm 1 Algorithm for Decision Making

---

**If no flit present in Side Buffer1**

```
{If (North flit is present) and (East flit is present)
{If (North flit wants R1) and (East Flit wants R2)
{Pass North flit to L1, Pass East flit to L1.}}
{Else If (North flit wants R1 ) and (East Flit wants R1
)}
{Pass North flit to L1, Store East flit in Side Buffer1.}}
```

**Flit present in Side Buffer1:**

```
{If (North flit is present) and (East flit is present)
{If (North flit wants R1) and (East Flit wants R2) and
(Side Buffer flit wants R1)
{Move North flit to Side Buffer1 , Pass Side buffer flit
to L1 through north input link , Pass East flit to L1.}}
{Else If (North flit wants R1) and (East Flit wants R2)
and (Side Buffer flit wants R2)
{Move East flit to Side Buffer1 , Pass Side buffer flit to
L1 through east input link, Pass North flit to L1.}}
{Else If ( (North flit is present) and (East flit is not
present)
{Pass Side buffer flit to L1 through east input link, Pass
North flit to L1.}}
```

---

bit header field. We use necessary reassembly mechanism for handling out-of-order delivery of flits. We simulate mesh network of different sizes with different standard synthetic traffic patterns like uniform, bit-complement, transpose, tornado, shuffle and neighbor and observe the average values of packet latency, deflection rate and throughput. We analyze the performance of HiPAD in comparison with MinBSD using real applications mixes from SPEC CPU2006 benchmark suite [8]. Using a 64-core CMP simulator, we generate 7 multiprogrammed workload mixes (M1 to M7 in Figure 7 )and provide the network events from this as traffic to the NoC simulator.

#### V. EXPERIMENTAL RESULTS AND ANALYSIS

**Average Flit Latency:** Higher latency of flits in the network increases the stall time of applications resulting in their throttling and poor application level performance. Hence for better performance, the average flit latency should be minimum. Figure 4 shows the average flit latencies of HiPAD, MinBSD and MinBD routers for four typical synthetic traffic patterns on 8x8 mesh network. Flit latencies are calculated in unit time by multiplying latency in cycles with time for one cycle (which is different for MinBD, MinBSD and HiPAD). HiPAD exhibits the least average flit latency for all values of injection rates and network saturation point is extended by proposed architecture by 33% compared to MinBSD and 6.6% compared to MinBD. Minimisation of SBOT using intelligent logic for buffering and deflection accounts for the reduction in average latencies. From Figure 6, we see that using HiPAD router, SBOT of 93% of the flits is 1 cycle or less and 72% of flits have a latency less than or equal to 10 cycles. As shown

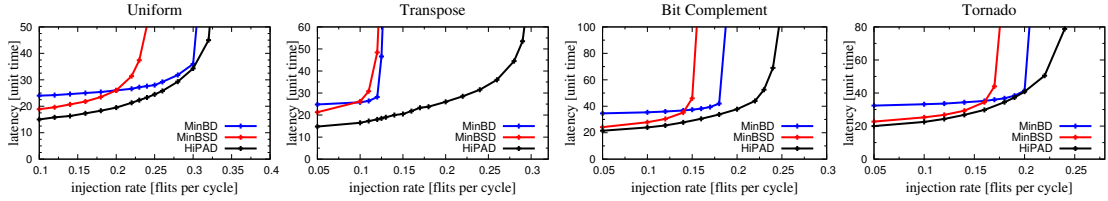


Figure 4: Average flit latency comparison under various synthetic traffic patterns in  $8 \times 8$  mesh network.

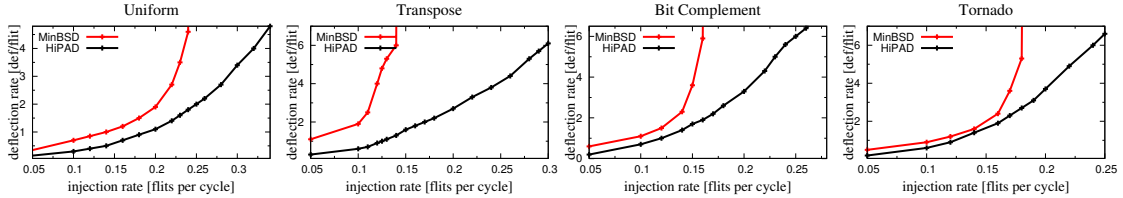


Figure 5: Average flit deflection comparison under various synthetic traffic patterns in  $8 \times 8$  mesh network.

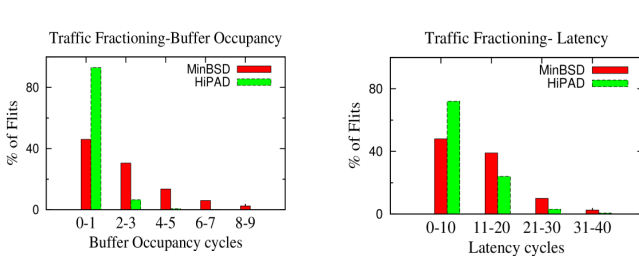


Figure 6: Traffic Fractioning

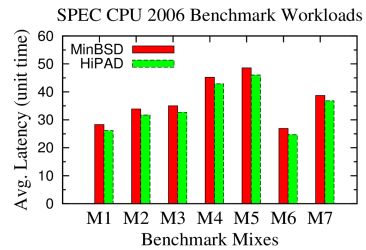


Figure 7: Average latency for real applications.

in Figure 7, HiPAD shows an average reduction of 8% compared to MinBSD for real application workloads also.

**Average Deflection Rate:** Injection rate versus deflection rate plot in Figure 5 shows that the deflection rate of the proposed work is lower than that of MinBSD for various injection rates. If the deflection for a packet is less, it reaches its destination faster leading to lower latency. But sometimes the waiting time of flits in the buffers may also lead to increased latency. Since in this work, one among the flits asking for the same target arbiter is stored in the side buffer due to intelligent decision making, the deflection of the flits is reduced.

#### A. Router Pipeline Latency, Static Power and Area

We implement Verilog models of all the three routers mentioned above and synthesize using Synopsis design compiler in 65nm technology to compute their pipeline latencies. Router delay can be defined as the total time a flit takes to move from the input links of the router to its output links. The synthesis report shows that the router delay of the proposed technique is 14% more than that of MinBSD whereas it is 38% lesser than that of MinBD.

## VI. CONCLUSION

In this work, we analyzed the existing state-of-the-art deflection routers and identified few drawbacks occurring due to the positioning of various units in their pipeline. We noted some of the drawbacks with the design of the single cycle deflection router, MinBSD and proposed a new router architecture. The unnecessary flit movements from

an input link to the buffer through an extra arbiter within the router is removed. This concept helps in achieving higher performance by reducing average flit latencies. An intelligent logic is introduced before output port arbitration to decide on which flit is to be injected into the PDN. Hence HiPAD router can be employed to achieve higher performance for mesh NoCs.

## REFERENCES

- [1] W. Dally, "Virtual-channel flow control," *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, no. 2, pp. 194–205, 1992.
- [2] T. Moscibroda and O. Mutlu, "A case for bufferless routing in on-chip networks," in *ISCA*, 2009, pp. 196–207.
- [3] C. Fallin et al., "CHIPPER: A low complexity bufferless deflection router," in *HPCA*, 2011, pp. 144–155.
- [4] C. Fallin, G. Nazario, X. Yu, K. Chang, R. Ausavarungrun, and O. Mutlu, "MinBD: Minimally-Buffered Deflection Routing for Energy-Efficient Interconnect," in *NOCS*, 2012, pp. 1–10.
- [5] J. Jose et al., "DeBAR: Deflection based adaptive router with minimal buffering," in *DATE*, 2013, pp. 1583–1588.
- [6] G. R. Joanna, J. Jose, R. Radhakrishnan, and M. Mutyam, "MinBSD: Minimally Buffered Single Cycle Deflection Router," in *DATE*, 2014, pp. 1–4.
- [7] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. USA: Morgan Kaufmann Publishers Inc., 2003.
- [8] "SPEC2006 CPU benchmark suite," <http://www.spec.org>.