

# Traffic Aware Deflection Rerouting Mechanism for Mesh Network on Chip

Simi Zerine Sleeba\*, John Jose<sup>†</sup>, Maurizio Palesi<sup>‡</sup>, Rekha K. James<sup>§</sup> and M.G. Mini<sup>¶</sup>

\* Dept. of Electronics Engineering, Model Engineering College, Cochin, India

<sup>†</sup> Dept. of Computer Science and Engineering, Indian Institute of Technology Guwahati, India

<sup>‡</sup> Dept. of Electrical, Electronic and Computer Engineering, University of Catania, Italy

<sup>§</sup> Division of Electronics Engineering, School of Engineering, CUSAT, India

<sup>¶</sup> Dept. of Electronics Engineering, College of Engineering, Cherthala, India

simi.abie@gmail.com, johnjose@iitg.ac.in, maurizio.palesi@dieei.unict.it, rekhajames@cusat.ac.in, minimg@cectl.ac.in

**Abstract**—In two dimensional mesh Network on Chips (NoC), efficient routing algorithms route majority of the flits through the central routers of the network, whereas routers at the edges and corners experience relatively lesser flit flow. This in turn leads to higher traffic towards central routers than to edge and corner routers. Such uneven traffic distribution causes thermal hot-spots at the center of the chip where the load is high, and reduces the average life-time of the chip. In existing buffer-less deflection routing techniques, load balanced traffic distribution is not considered as a factor during assignment of links to mis-routed flits. Devising deflection routing techniques with greater load balancing capability is a major challenge for efficient thermal management of the chip. This paper proposes an adaptive routing mechanism that can provide a more balanced traffic profile in a deflection router based mesh NoC. Significant number of deflected flits are rerouted towards the edges/corners of the mesh, thereby reducing the load on the central routers. From evaluations, it is seen that the proposed technique reduces traffic variance compared to NoCs using baseline deflection routers. Transient temperature variation studies using Hotspot tool substantiate our findings.

**Index Terms**—Network on Chip, Deflection routing, Traffic rerouting, Traffic variance, Average latency

## I. INTRODUCTION

With the aim of enhancing the performance of processors, multiple computational cores are integrated on a single chip and are termed as Tiled Chip Multiprocessors (TCMP) [1]. Network on Chip (NoC) is widely envisioned as the interconnect of such TCMPs. In a homogeneous TCMP, various tiles are connected using a two dimensional mesh topology where each Processing Element (PE) is connected to a dedicated router and routers are interconnected using links. Data is exchanged between tiles in the form of packets. A packet is further divided into flits (flow control units). Packets generated from a PE make multiple hops through intermediate routers and links and finally reach their destination core. Each router has input/output ports to North, South, East and West directions and also to the local core.

This research is supported in part by Department of Science and Technology (DST), Government of India vide project grant ECR/2016/000212

The first generation NoCs used input buffered routers that use store-and-forward wormhole routing technique [2]. A flit occupies a buffer in the router until it wins arbitration for a productive output port. Buffers play a major role in improving the network performance parameters, but they consume significant amount of chip power. Buffer-less deflection routers are proposed as an alternate method for achieving energy efficient on chip communication [3]. Experiments show that buffer-less routers outperform buffered ones at low to medium injection rate [3]. Due to absence of buffers, flits that fail to occupy productive output ports are deflected through available output ports of the router. In deflection routing, a flit with higher priority is allocated to an output port of its choice. Output ports obtained by other flits are determined by the flit priority, port conflict and port allocation method. Consequently, traffic due to flit deflections may either be towards the center or the edges/corners of the mesh. Majority of the productive flit movements as well as large number of flit deflections occur through the central routers causing traffic imbalance and uneven thermal distribution across the mesh. In this paper, we propose a simple logic unit in the output port allocation stage of deflection routers that reroutes deflected flits away from the center of the mesh and improves the traffic evenness across the network.

## II. RELATED WORK

Most of the NoC routers adopt minimal routing techniques that focus on network performance rather than traffic balancing [4]. Due to restrictions imposed by the routing algorithm, certain regions in the network tend to have more concentration of traffic than the rest, creating an uneven traffic profile. Over the past decade, a wide variety of routing techniques for resolving network congestion have been proposed for NoCs with input buffered routers. Beginning with the Free Buffer Priority (FBP) scheme, the count of free input buffers in downstream routers is taken as a measure for adaptive selection of output ports [5]. BOFAR utilizes the history of buffer occupancy time of flits to determine congestion in downstream routers [6]. Another work introduces an aging-aware adaptive routing algorithm that routes packets along

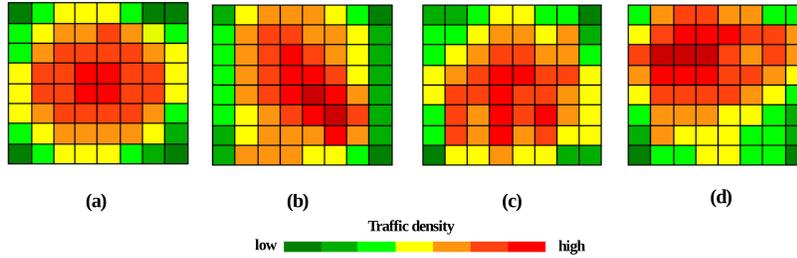


Fig. 1. Traffic Profile Graph for an 8x8 mesh NoC using (a) Uniform (b) Transpose (c) Shuffle (d) Benchmark mix traffic patterns

the paths which experience least congestion and minimum aging stress [7]. In ATDOR, a secondary network is employed for transmitting congestion messages and routing switches dynamically between XY and YX methods [8]. The problem of traffic balancing is not efficiently addressed in any of the above schemes. One major work in this direction is RCA, which uses the congestion information beyond adjacent routers to improve load balancing capability of the network [9]. GCA and GLB are also similar schemes where information on global congestion is the metric used for load balancing [10] [11]. Cool Centers follows an output port selection strategy based on prioritizing ports that route packets away from the center of the mesh [12]. This method is successful in balancing traffic flow in NoCs with input buffered routers.

Deflection routing algorithms exploit the path diversity of an NoC, hence they exhibit an inherent load balancing capability. Buffer-less deflection router BLESS performs sequential allocation of output ports to flits which are sorted in age order [3]. This routing technique reduces flit deflections to a minimum at the expense of longer critical path. CHIPPER exhibits better performance as a result of parallel port allocation and is considered to be the best buffer-less deflection router architecture [13]. A major drawback of CHIPPER is high flit deflection rate and subsequent dissipation of dynamic power. A category of minimally buffered deflection routers derived from CHIPPER reduce flit deflections without significant impact on power and area [14]. The structural limitation of the output port allocator used in all CHIPPER based router architectures causes low priority flits to be deflected randomly through vacant output ports. Hence, additional mechanisms are to be devised for achieving uniform traffic distribution for such NoCs. A recently proposed method mitigates network congestion by deflecting flits away from destination nodes that are identified as hotspots when a flit counter of the router exceeds a preset threshold [15]. A deflection routing technique which attains a uniform traffic distribution throughout the mesh NoC without compromising on the performance parameters is an open challenge.

### III. MOTIVATION

In CHIPPER, productive output ports of input flits are computed using XY method and output port allocation is done using a Permutation Deflection Network (PDN). Simulations

are performed on 8x8 mesh NoC using CHIPPER with various synthetic traffic patterns and SPEC CPU 2006 benchmark mixes. The router wise traffic distribution is recorded for a simulation period of 1 million cycles with flit injection rate close to saturation point of the network. The number of flits passing through each router is used to generate a Traffic Profile Graph (TPG) [12]. Figure 1 shows the TPG for Uniform, Transpose, Shuffle and SPEC CPU 2006 benchmark mix traffic patterns. In a TPG, the 64 routers in an 8x8 mesh NoC are represented by 64 squares placed like an 8x8 matrix. The color of each square represents the amount of flit traffic moving through the corresponding router. This value is referred to as the traffic density of the router. These values are encoded by choosing an appropriate color from the color scale which shows a transition from green to red. A square with deep green color represents a router with minimum traffic passing through it. Similarly, a deep red colored square implies that the corresponding router carries heavy load.

From Figure 1, we observe that traffic density is significantly higher at the central locations of the mesh (having more red squares) than at the edges and corners in all the four cases. As seen from Figure 1(b), the uneven traffic distribution is more evident for a network intensive traffic pattern like Transpose than random patterns such as Uniform and Shuffle. Detailed flit flow analysis through the 16 central routers of the 8x8 mesh shows that 77% of it is due to flits moving in productive paths to their respective destinations and the remaining 23% is due to the deflected flits. In CHIPPER, due to unrestricted deflection schemes during port conflicts, all flits other than the highest priority flit may get deflected. For a flit whose productive port is South, assignment of East or West port is equally counter productive. Approximately 15% of these deflected flits in the central routers could be reassigned to vacant ports towards the edges/ corners of the mesh.

The 15% cases observed above show that the traffic density at the center of the mesh network could be reduced by rerouting some of the flits in each router towards the edges of the mesh without increasing the hop. In this paper, a traffic aware deflection routing mechanism that balances the load across the NoC by port reallocation of deflected flits is proposed. The additional router logic does not alter the path of flits traversing through productive directions. Hence

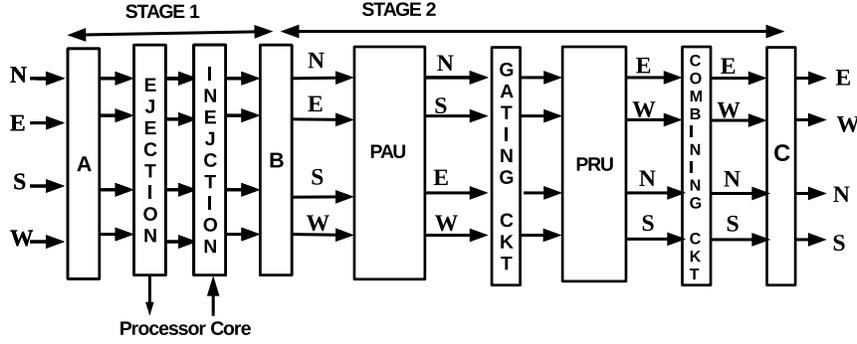


Fig. 2. Two stage pipeline diagram of the proposed router

network performance is not affected due to the proposed traffic balancing mechanism.

#### IV. ROUTER ARCHITECTURE

The router architecture consists of two pipeline stages as shown in Figure 2. The first stage consists of basic input functional blocks viz. ejection and injection units. The second stage consists of the output port allocation stage which is referred to as the Parallel Allocation Unit (PAU) followed by a Port Reallocation Unit (PRU) which is a newly proposed module. Each of the functional blocks in the router pipeline are explained in detail below.

##### A. Ejection and Injection Units

The functional blocks in the first stage of the proposed router are same as that of CHIPPER. The flits destined to the local PE are ejected out of the NoC through the Ejection Unit. This architecture supports ejection of a single flit per cycle. If there are more than one flit with the local core as destination, the flit with the highest priority is ejected while the others are deflected to neighboring routers at the end of the router pipeline. Such flits come back to the same router in subsequent cycles and compete for the ejection port. Injection of new flits from the local core is done by the Injection Unit subject to a vacancy in any of the four internal flit channels. Productive output port for the flits are computed using XY routing algorithm. After passing through the Ejection and Injection units, the flits move to register B at the end of a clock cycle.

##### B. Parallel Allocation Unit (PAU)

The second stage of the router pipeline begins from pipeline register B. The flits in register B are assigned output ports by the PAU on the basis of flit prioritization and port preference of each flit. The function of PAU is similar to that of PDN in CHIPPER. The PAU consists of four permuter blocks (P1, P2, P3 and P4) each having two input ports and two output ports as shown in Figure 3. Flits from the North and East internal flit channels are connected to P1 and that of South and West are connected to P2. For each input permuter block (P1 and P2), the input flit with higher priority is assigned to an

output permuter (P3 or P4) of its choice and the other input flit is assigned to the second output permuter. For example, consider two flits F1 and F2 with F1 coming through the North and F2 coming through the East internal flit channels. Let us assume that the preferred output port for F1 and F2 is South. Then, as per the flit priority ( $F1 > F2$ ), F1 moves from P1 to permuter P3 to which South output port is attached. Automatically, F2 will be deflected to permuter P4 from the output of P1. Mapping of inputs to outputs in P1 and P2 is done simultaneously and the same is followed in P3 and P4. The parallel structure of permuters in PAU reduces the critical path latency of the port allocation stage.

The golden flit prioritization scheme used in CHIPPER for livelock avoidance is utilized here. The golden flit obtains desirable output ports in all the routers in its path to reach its destination without any deflection. The priority is then passed on to another flit in transit. We implement this priority mechanism so as to have a fair comparison between CHIPPER and the proposed traffic aware deflection routing method.

##### C. Port Reallocation Unit (PRU)

Flits from the PAU are connected to the PRU by a gating circuit as shown in Figure 2. The function of PRU is to reassign deflected flits to vacant output ports of a router. The main motive behind this is to assign a port to an already deflected flit so that the flit moves away from the center of the mesh. As shown in Figure 3, a flit, F from an output line of the PAU enters the PRU only if all the following three conditions are satisfied.

- (1) F is not assigned to a productive port
- (2) The port assigned to F by the PAU will take it to a router R that is farther from the edges/corners of the mesh than the current router C (ie. C is relatively closer to the center of the mesh than R)
- (3) There exists an idle output port in the current router C that will take F to a router R1 that is closer to the edges/corners of the mesh than C.

In short, the router R1 to which a flit is rerouted should be such that the minimum number of hops to reach the

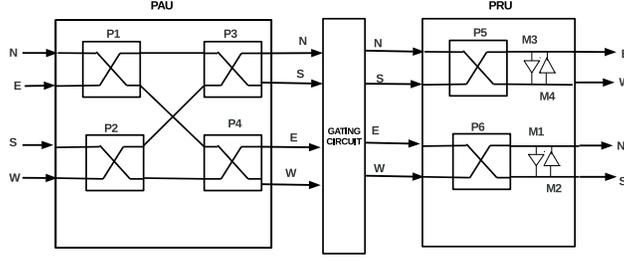


Fig. 3. Structure of Parallel Allocation Unit (PAU) and Port Reallocation Unit (PRU)

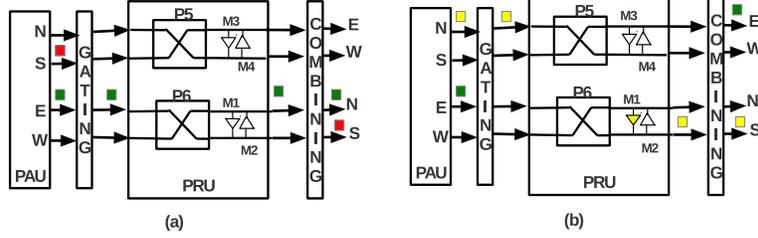


Fig. 4. Examples for port reallocation for router 50 in an 8x8 mesh network

edges/corners of the mesh from R1 is lesser than that of C and R. This conditional check and subsequent selective forwarding are done by a gating circuit.

#### Algorithm 1 Algorithm for PRU

Inputs : P5(Northin), P5(Southin), P6(Eastin), P6(Westin)  
Outputs: P5(Eastout), P5(Westout), P6(Northout), P6(Southout)

```

If flit in P5(Northin) or P5(Southin)
  If flit in P5(Northin) and Enabled(P5(Eastout))
    Assign P5(Eastout) := P5(Northin)
  Else If flit in P5(Northin) and Enabled(P5(Westout))
    Assign P5(Westout) := P5(Northin)
  If flit in P5(Southin) and Enabled(P5(Westout))
    Assign P5(Westout) := P5(Southin)
  Else If flit in P5(Southin) and Enabled(P5(Eastout))
    Assign P5(Eastout) := P5(Southin)
If flit in P6(Eastin) or P6(Westin)
  If flit in P6(Eastin) and Enabled(P6(Northout))
    Assign P6(Northout) := P6(Eastin)
  Else If flit in P6(Eastin) and Enabled(P6(Southout))
    Assign P6(Southout) := P6(Eastin)
  If flit in P6(Westin) and Enabled(P6(Southout))
    Assign P6(Southout) := P6(Westin)
  Else If flit in P6(Westin) and Enabled(P6(Northout))
    Assign P6(Northout) := P6(Westin)

```

The PRU consists of two permuter blocks (P5, P6), each having two inputs and two outputs. P5 reallocates flits from the North and South output lines of the PAU to the East or West ports of the router if there is a flit F belonging to any of the above mentioned conditions. Similarly, P6 connects East and West outputs of PAU to North and South output ports. Algorithm 1 gives the rules for rerouting using PRU. Reallocation of flits between North and South or East and West directions are enabled using multiplexers (M1, M2, M3, M4) between output lines of P5 and P6 as shown in Figure 3. If there are no flits that satisfy the above conditions, the gating circuit bypasses the flits over the PRU by keeping the timing

constraints. The combining circuit multiplexes output lines of the PRU with corresponding output lines from the PAU.

The functionality of PRU is explained in detail with an example. Consider an 8x8 mesh topology with routers numbered from 0 (bottom left) to 63 (top right). Consider router number 50 whose South and East output ports are directed towards the center of the mesh whereas the North and West ports are towards the edges. Figure 4(a) shows PAU and PRU of this router. We assume that there are two flits at the output of PAU. The PAU assigns the East port to the green flit and South port to the red flit. Assume that East is a non-productive direction for the green flit. Hence it is ready to be deflected. At the same time, North and West ports are empty. Figure 4(a) shows that the green flit is rerouted towards the vacant North port by the permuter P6 in the PRU. Since the South port is a productive port for the red flit, it bypasses the PRU. Forwarding of green flit to the PRU and bypassing of the red flit over the PRU is done by the gating circuit by proper condition check. Figure 4(b) shows an example of a deflected flit being reassigned from North port to South port by enabling multiplexer M1. In this example, permuters P5 and P6 of the PRU are not used.

## V. EXPERIMENTAL METHODOLOGY

We use an open source NoC simulator, Booksim 2.0 [16] to model the deflection router based NoC with the proposed architecture. The router pipeline is modeled with two cycle delay and the routing algorithm is implemented as described in Section IV. To compare the results, the basic CHIPPER based NoC is also modeled in Booksim. Simulations are conducted using typical synthetic traffic patterns as well as network traces generated by running multi-programmed workload mixes from

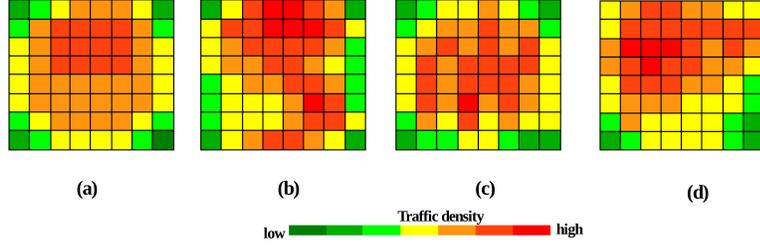


Fig. 5. Traffic Profile Graph for an 8x8 mesh NoC with traffic aware routing using (a) Uniform (b) Transpose (c) Shuffle (d) SPEC CPU 2006 benchmark mix traffic patterns.

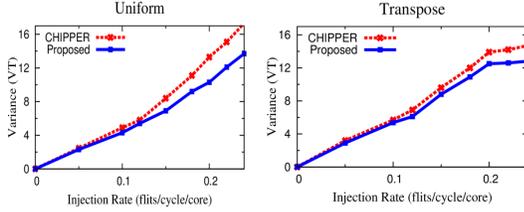


Fig. 6. Traffic Variance for Synthetic Traffic Patterns

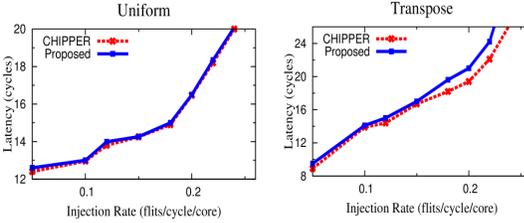


Fig. 7. Average Latency for Synthetic Traffic Patterns

SPEC CPU 2006 benchmark suite [17] on Gem5 simulator [18].

## VI. EXPERIMENTAL RESULTS AND ANALYSIS

The Traffic Profile Graph for Uniform, Transpose, Shuffle and benchmark application traffic patterns for an 8x8 mesh NoC with the proposed method is depicted in Figure 5. Compared to Figure 1, we see that the port reallocation strategy used in the proposed method helps to reduce the traffic density at the center of the mesh by rerouting them to the edge/corner routers. From the Figure 5, we see that squares at the center change from red to orange and yellow colors. Similarly, the color of squares at the edges change from deep green to pale green or yellow.

### A. Traffic Variance

In order to measure the traffic load and uniformity of traffic distribution across various routers in an NoC, we introduce a parameter known as traffic variance which is calculated using the formula,

$$\text{Traffic Variance, } V_T = \frac{\sum_{i=1}^{64} \text{mod}(A_v - T_i)}{64} \quad (1)$$

where

$$\text{Average, } A_v = \frac{\sum_{i=1}^{64} (T_i)}{64}$$

$T_i$  is the traffic density of the  $i^{\text{th}}$  square in the TPG.

In an NoC, lower value of traffic variance signifies higher uniformity in traffic distribution. Using Equation 1, the traffic variance for 8x8 mesh NoCs using CHIPPER and the proposed router for typical synthetic traffic patterns are calculated for various network injection rates. Figure 6 shows the graphs for uniform and transpose traffic patterns. The proposed re-routing scheme confirms lower variance compared to CHIPPER for all the cases. The lowest variance shown is 26% for uniform pattern at saturation injection rate which is approximately 0.2. Transpose pattern represents network intensive traffic with specific packet destinations. Under transpose pattern, very few flits satisfy all the three conditions for port reallocation. This accounts for the minor reduction in variance compared to uniform traffic.

Figure 7 shows the comparison of average latency for uniform and transpose traffic patterns for 8x8 mesh NoC using CHIPPER and proposed router. As flits progressing in productive paths are unaffected by the traffic re-routing mechanism, there is only negligible increase of 0.05% in average latency for the proposed mechanism. We also find that the deflections per flit reduces up to 8% since edge routers are lightly loaded and flits encounter lesser port conflicts in them.

### B. Real Applications

Applications from the SPEC CPU 2006 benchmark suite are categorised as low, medium or high MPKI (misses per kilo instructions) based on the rate at which they inject packets into the network. Table I lists the applications from each category which are used in our simulations. Each simulation uses network traces generated from a 64-core TCMP running one SPEC benchmark application per core as given in the Table. Figure 8 shows the graph of normalised variance and latency of the proposed NoC with respect to CHIPPER for various benchmark mixes. Although the proposed method delivers promising results for all applications, significant reduction in

	Category	Benchmark Applications
Low MPKI	C1	calculix, h264ref, gromacs, gobmk
Medium MPKI	C2	gcc, bwaves, bzip2
High MPKI	C3	leslie3d, hmmer

TABLE I

APPLICATIONS OF VARIOUS NETWORK INJECTION INTENSITY IN SPEC CPU BENCHMARK SUITE

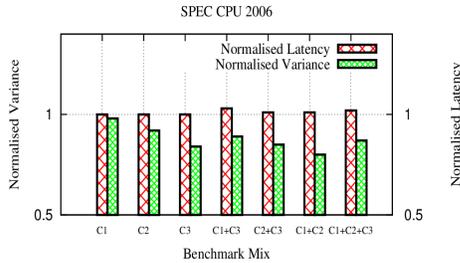


Fig. 8. Normalised variance and latency w.r.t. CHIPPER for 8x8 mesh NoC using mixes from SPEC CPU 2006 benchmark suite

traffic variance is noted for high MPKI applications. Since they inject more number of flits into the network, the probability of deflections in the routers also increases. Deflected flits that satisfy the conditions for reallocation are forwarded to the edges/corner routers of the NoC, due to which traffic variance is low. It is also observed that the average latency of the proposed technique is equivalent to the normal value of CHIPPER for all application mixes.

### C. Thermal Analysis

Thermal distribution across the NoC is analyzed using Hotspot 6.0 tool [19]. Dynamic power dissipation of various routers in an 8x8 mesh NoC due to varying load is extracted by modeling our router architecture in Orion 2.0, a power estimation tool [20]. Using these power traces obtained from Orion, Hotspot estimates the transient temperature variation due to the flit flow load across the 64 routers of the NoC. Simulation results confirm that there is a temperature reduction of upto 3°K in the 16 central routers for real workloads using our proposed scheme compared to that of baseline CHIPPER.

### D. Hardware Synthesis

We implement Verilog models of the proposed router and CHIPPER and synthesize using Synopsys Design Compiler with 65nm CMOS library. Router delay is the time taken by a flit to move from its input to output port through various functional units. The first stage of CHIPPER and the proposed router have the same delay because of similar functional units in both architectures. The output stage of CHIPPER consists mainly of a port allocator whereas the proposed router includes a port allocator (PAU) followed by a rerouting logic (PRU). Since both the architectures use the same port allocation logic, additional delay of 18% due to the PRU occurs in the output stage of the proposed router. Area and static power consumed by the control logic of our router are 4% and 7% higher than CHIPPER. The hardware overhead and the extended critical path of the proposed technique are justified by the significant

reduction in traffic variance that it promises. However our router pipeline frequency is same as that of CHIPPER as latency of the first stage dominates over the second stage.

## VII. CONCLUSION

An adaptive deflection router which helps to achieve a uniform traffic distribution within the mesh NoC is proposed. A logic block is introduced into the router pipeline that performs output port reallocation of flits which are assigned to unproductive directions towards the center of the mesh. The merit of the proposed traffic aware routing mechanism and the corresponding thermal variation effect are quantitatively justified by the reduction in traffic variance parameter in comparison with a basic deflection router based NoC.

## REFERENCES

- [1] J. Balfour and W. J. Dally, "Design tradeoffs for tiled cmp on-chip networks," in *Annual International Conference on Supercomputing*, 2006, pp. 187–198.
- [2] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. USA: Morgan Kaufmann Publishers Inc., 2003.
- [3] T. Moscibroda and O. Mutlu, "A Case for Bufferless Routing in On-Chip Networks," in *ISCA*, 2009, pp. 196–207.
- [4] G. Chiu, "The Odd-Even Turn Model for Adaptive Routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, pp. 729–738, 2000.
- [5] W. Dally, "Virtual-Channel Flow Control," *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, no. 2, pp. 194–205, 1992.
- [6] J. Jose et. al, "BOFAR : Buffer Occupancy Factor based Adaptive Router for mesh NoC," in *International workshop on Network on Chip Architectures*, 2011, pp. 23–28.
- [7] K. Bhardwaj et. al, "Towards Graceful Aging Degradation in NoCs through an Adaptive Routing Algorithm," in *Design Automation Conference*, 2012, pp. 382–391.
- [8] R. Manevich et al., "A cost-effective centralized adaptive routing for networks-on-chip," in *DSD*, 2011, pp. 39–46.
- [9] P. Gratz et. al, "Regional Congestion Awareness for Load Balance in Networks-on-Chip," in *International Symposium on High-Performance Computer Architecture*, 2008, pp. 203–215.
- [10] M. Ramakrishna et. al, "GCA: Global Congestion Awareness for Load Balance in Networks-on-Chip," in *NoCS*, 2013, pp. 21–24.
- [11] M. Ebrahimi et. al, "GLB - Efficient Global Load Balancing Method for Moderating Congestion in On-Chip Networks," in *International Symposium on Reconfigurable Communication-centric Systems-on-Chip*, 2012, pp. 1–5.
- [12] J. Jose et. al, "An Energy Efficient Load Balancing Selection Strategy for Adaptive NoC Routers," in *International workshop on Network on Chip Architectures*, 2014, pp. 31–36.
- [13] C. Fallin et al., "CHIPPER: A Low Complexity Bufferless Deflection Router," in *HPCA*, 2011, pp. 144–155.
- [14] C. Fallin, G. Nazario et al., "MinBD: Minimally-Buffered Deflection Routing for Energy-Efficient Interconnect," in *NOCS*, 2012, pp. 1–10.
- [15] R. Raj R.S. et. al, "Implementation and analysis of hotspot mitigation in mesh nocs by cost-effective deflection routing technique," in *VLSI-SoC*, 2017, pp. 1–6.
- [16] N. Jiang et al., "Booksim 2.0 User's Guide." <http://nocs.stanford.edu>.
- [17] J. Henning., "SPEC CPU 2006 Benchmark Descriptions," in *SIGARCH Computer Architecture News*, 2006.
- [18] N. Binkert et al., "The gem5 simulator," *SIGARCH Computer Architecture News*, vol. 39, no. 2, pp. 1–7, 2011.
- [19] S. Sharifi et al., "Hybrid dynamic energy and thermal management in heterogeneous embedded multiprocessor socs," in *ASP-DAC*, 2010, pp. 873–78.
- [20] A. B. Kahng et al., "Orion 2.0: A Fast and Accurate NoC Power and Area Model for Early Stage Design Space Exploration." *IEEE Transactions on VLSI*, vol. 20, no. 1, pp. 191–196, 2012.