ATAR: An Adaptive Thermal-aware Routing Algorithm for Three-Dimensional Network-on-Chip Systems

Ranjita Dash[1], *Student Member, IEEE*, Amartya Majumdar[2], Vinod Pangracious[3], *Member, IEEE*, Ashok Kumar Turuk[4], *Member, IEEE*, and José L. Risco-Martín[5]
[1,2,4] National Institute of Technology, Rourkela,India
[3]American University of Dubai, Dubai, UAE
[5]Complutense University of Madrid, Madrid, Spain

Three-dimensional Network-on-Chip (3D NoC) is gaining popularity among designers due to scalability, higher bandwidth, fault tolerance, and reliability. However, the stacking of multiple dies leads to severe thermal problems due to increase in power density. Unequal traffic distribution across the chip and higher power density result in higher on-chip temperature resulting in performance degradation, increase in leakage power, and circuit failure. In this paper, a novel routing method called Adaptive Thermal-Aware Routing (ATAR) is proposed to alleviate peak on-chip temperature. The proposed technique is simulated using AccessNoxim simulator. The parameters for simulation are taken Global Foundries and Terazon Semiconductors. Analysis of the simulation and experimental results shows, ATAR-based designs make the on-chip traffic and thermal distribution more uniform. Different traffic patterns are used for the validation of simulation and 3D design platforms developed to demonstrate thermal optimization in high performance 3D NoC-based parallel processing systems.

***Index Terms*—Thermal Model, ATAR, 3D VLSI, Network on Chips, Micro Architecture, Network Traffic.**

1. Introduction

THREE dimensional NoC is an on-chip communication subsystem between the Intellectual Property (IP) blocks in a System-on-Chip (SoC). The use of NoCs incorporates heterogeneous cores in a single chip to provide a power-efficient, scalable and robust architecture [2]. 3D NoCs have the benefit of both 3D integration and NoC[4]. The benefits include shorter interconnects, smaller form factor and reduced delay. However, thermal challenges arise in 3D NoCs due to high transistor junction temperature, exacerbated spatial thermal gradients and higher integration density. Due to the aforementioned reasons, designing the worst-case cooling system design is not feasible. Therefore, Runtime Thermal Management (RTM) techniques are proposed for cooling 3D NoCs. A significant portion of chip power is consumed by the on-chip communication system. The power consumption of NoCs would be higher in future generations because of advanced technology and architecture.

Interconnects use more power in comparison to logic blocks due to technology scaling. The interconnect power in future technology nodes are expected to take up 65%-80% of the total chip power[6]. Because of higher performance and finer control, integrating many simple cores gives a better result than integrating a small number of complex cores[7]. This leads to the increase in communication power budget.

In case of communications-centric applications like MIT-Raw chip, [8] is accounted for 40% of the entire chip power. Processors are dominated by on-chip networks in heat generation while executing communication-centric applications. Power density within the router area can be 5.5 times higher than that of the IP [9]. In Intel's 80-core SoC, the power density of NoC routers is almost twofold than the floating point and memory units in the die [10]. The aforementioned reasons make NoC a greater contributor to on-chip heating in comparison to the other units present on the die.

Thermal-aware routing offers an opportunity to control the workload spanning across the whole chip. It is done by a routing method to alleviate hotspots by efficient migration of load to the cooler areas in the chip to attain thermal optimization. There exist many paths between a source-destination node pair. In thermal-aware routing case, thermally hot-spot paths are avoided and the coolest path among the available is chosen to reduce the power density at the nodes along the path to moderate the temperature. This method can be applied by any thermal-aware techniques available in the literature proposed, like floor-planning[11], task mapping, etc.

The contributions of this paper have been summarized as follows:

1) A novel thermal-aware routing method is proposed for uniform diffusion of heat from the 3D NoC, and a distributed adaptive routing based control architecture, ATAR, is developed to execute the proposed approach.
2) ATAR is simulated using AccessNoxim simulator [27]. The parameters considered for simulation are taken from Global Foundries and Terazon Semiconductors.
3) ATAR is implemented to compute and propagate cost in accordance with turn models to avoid deadlock and livelock.
4) Experimental results obtained in this paper are rigorously evaluated through experiments and compared with state-of-the-art techniques using different synthetic traffic scenarios. Results for temperature and performance are compared and discussed.

In Section 2, the literature survey presents the contributions

Manuscript received December 1, 2012; revised August 26, 2015. Corresponding author: Ranjita Dash (email:ranjita.rakhi@gmail.com).

and limitations of various existing thermal-aware routing techniques. Physical implementation of the 3D NoC architecture used is described in Section 3. The proposed algorithm is based on a thermal model, which is described in Section 4, and the router micro-architecture in Section 5. Section 6 explains the proposed work and Section 7 describes the results. The paper is concluded in Section 8.

2. Literature Survey

Thermal modeling has gained a lot of attention in recent years because of greater thermal challenge in current and future very large scale integration (VLSI) systems. Recently, researchers have paid much attention to thermal-aware routing algorithms. Due to high switching activity of routers, there has been a significant increase in overall on-chip temperature. The power consumption of a router depends on its utilization rate and it also leads to increase in temperature. Thermal-aware routing algorithm improves reliability and reduces cooling costs of the chip by distributing the chip temperature across the chip. It ensures that the temperature of the routers is below the threshold temperature. A proficient routing algorithm limits traffic congestion, the formation of hotspots, and packet delay. Network throughput is increased by distributing the network load uniformly among the paths given by the network topology. As thermal distribution is highly correlated with power consumption, routing algorithm plays a great role in reducing the on-chip peak temperature. This is useful when the NoC takes a major share of the total system power, e.g. 40% of total power in the MIT RAW chip [14] and 30% in case of Intel 80-core teraflop chip [10]. The main concern of routing algorithms for 3D NoCs is thermal management, freedom from deadlock, link assignment for each packet, and reliability. Table I shows the major contribution of the authors along with their limitations towards thermal-aware routing.

3. 3D NoC Demonstrator

This section describes the design and implementation of 3D-NoC based many-core chip and its architecture. The 3D many-core chip considered in this paper has 4 layers and each layer has 64 cores arranged in a 2D platform of 8×8. The chip size is about 8 mm $\times$ 8 mm with additional test circuits placed as peripheral sections.

We used six metals of 130 nm process node given by Global Foundries. These are enhanced to accommodate by taking the specification of Tezzaron Semiconductor into account [34]. Very small TSVs are produced by Tezzaron's via-first 3D manufacturing process. These are approximately 1.2 μm wide with 2.5 μm minimum pitch and 6 μm height [35]. The area around the TSV is expanded to accommodate *keep out zones* [35] to make TSVs fit within 5 to 6 standard cell area because the area of a 4-transistor logic gate is supposed to vary from 0.82 μm^2 to 0.20 μm^2 by 2017 [35]. The TSV *keep out zones* are necessary for the 3D design to keep the performance of active device placed near to TSVs. The calculated values of TSV resistance R_{TSV} and capacitance C_{TSV} are $\approx 600\ m\Omega$ and 15 fF, respectively. The wire delay estimation of NoC interconnects of the 3D stacked NoC is derived from the 5 tier layout developed using Tezzaron Process [34]. The TSV delay is evaluated using *eldo* is $\approx$ 28 *pS*. In tier 1, the spatial distribution of TSVs and interconnect switches along with I/Os are configured to optimize the wire delay. The NoC design in layers 1 and layer 4 are performed to optimize the placement of TSVs onto single side only. The design tool developed to test Face-2-Back (F2B) stacking methodology is given by Tezzaron's 3D design platform using via first TSV process. It supports two types of TSV structures: Supper-Contact and Super-Via. In our design we have used Supper-Contact, using Tungsten via fill. In F2B stacking, the tier 1 via-first TSVs have their landing pads on metal 1 and metal 6. The connection between via-first TSVs is made using local interconnection and vias in between adjacent dies. The tier 1 die is thinned down to TSVs first and bonded using the TSV landing pads. These landing pads include *keep-out-zones* uniformly located around them to reduce coupling effects on active devices located around it.

Communication between NoCs in tier 1 and tier 2 is carried out using F2B vias. Any net that connects to F2B via and thus interconnecting circuitry from tier 2 to tier 1, is known as a 3D net. The NoC interconnections are considered to be 3D nets. The individual designs for each die (tier 1 to 4) should contain pins for all nets that cross the vertical interconnection boundary of the 4 tier test chip. The test chip contains 4 tiers with 64 NoCs plus cores in each tier. The interconnecting vias from tier 1, 2, 3, and 4 were manually placed on both dies for F2B stacking. A minimum TSV pitch of 250 μm throughout the entire wafer is considered. This requirement forces to include at least one TSV inside every 250 μm window in our design. According to the Tezzaron 3D process, this is used for planarity of the wafer during chemical and mechanical polishing (CMP) process. In the case of F2B back stacking the tier 1 to 4 is thin down to TSV, that is 6 μm in length before bonding. In certain locations, we manually inserted dummy TSVs before placement to meet this requirement. Figure 1 presents the TSV assignment of tier 1 and 2 dies of a cluster in the 3D stacked mesh-based many-core test chip. The power and ground distribution networks (PDNs) are mainly generated using the strip and ring generation commands in cadence Encounter. Figure 1 shows all 4 layers and TSV points attached with NoCs.

4. Thermal Model

The temperature of a node in an NoC depends on the following two factors.

- Transformation of electrical energy consumed by the node into heat.
- Thermal conduction among the nodes with different temperature.

Accordingly, temperature of a node can be expressed as

$$T_n = T_n^0 + T_{ET} + T_{TC} \tag{1}$$

where T_n^0 represents the initial temperature of node n, T_{ET} and T_{TC} are the temperature generated due to energy transformation and thermal conduction, respectively. T_{ET} is directly proportional to the energy consumed by node *n*. A coefficient θ is used to indicate the relationship between the

TABLE I: Literature Review

Authors	Contributions	Limitations
Ying *et al.*[15]	Selects minimal paths to route packets to vertical links without any extra virtual channel.	*i*) Restriction on the placement of functional blocks; *ii*) low reliability.
Dubois *et al.*[16]	Improvement of the network performance with minimal hardware overload	*i*) Low path diversity; *ii*) poor reliability
Chao *et al.*[17]	*i*) Downward XYZ routing; *ii*) mitigates routing load of layers closer to heat sink; *iii*) improves heat diffusing efficiency	*i*) Hardware overload; *ii*) intra-layer cool paths are not utilized; *iii*) application specific
Lin *et al.*[13]	*i*) Traffic and thermal-aware adaptive routing; *ii*) uses OE turn model to avoid deadlock; *iii*) detour throttled nodes and efficient routing at same logic layer	*i*) Heavy traffic in specific columns; *ii*) unbalanced vertical traffic load
Chao *et al.*[18]	Transport layer assisted routing for NSI-mesh	*i*) Look-up table size grows with the topology size; *ii*) high computational complexity
Salamat *et al.*[19]	*i*) Fault-tolerant routing algorithm to improve reliability in vertically partially 3D NoCs; *ii*) two virtual problem to solve deadlocks	Reassignment of vertical link results in packet detouring and low throughput
Taheri *et al.*[20]	Cool vertical link used for communication	*i*) Low throughput; *ii*) increase in delay
Jiang *et al.*[21]	*i*) Fully adaptive thermal-aware runtime algorithm; *ii*) Uniform distribution of temperature across the network; *iii*) 12-bit register is used to reserve the router state	Increase in hardware cost
Chao *et al.*[22]	*i*) Peak power reduction using bio-inspired ant colony approach; *ii*) uniform distribution of packets to minimize the occurrence of hotspots	*i*) Application specific traffic information not taken into account; *ii*) area and power overhead; *iii*) unsuitable for dynamic networks due to high complexity; *iv*) processor core power is not considered to calculate on-chip temperature
Liu *et al.*[23]	*i*) Dynamic thermal balance routing; *ii*) minimal path is chosen for packet delivery; *iii*) use of virtual channel to avoid deadlock	*i*) Heavy traffic congestion in minimal path region; *ii*) degrades system performance
Xiang *et al.*[24]	Parallel port allocation mechanism within routers to reduce router critical path latency	Lack of buffers prevents efficient handling of multicast, broadcast operations
Kamran *et al.*[25]	Enhanced decision making routing algorithm to avoid congestion in 2D NoC architecture	Cannot tolerate multiple link failures at the same time
Hsin *et al.*[26]	*i*) Predict the future temperature based on the data in the look-up table; *ii*) low computational complexity	*i*) Precision depends on offline profiling; *ii*) large area overhead
Chen *et al.*[28]	*i*) Predict the future temperature based on the sensing results and current workload; *ii*) less area overhead; *iii*) application independent approach	Precision depends on the computing time or initial computing parameters
Lee *et al.*[29]	*i*) Dynamic regulation of buffer depth to avoid congestion in overheated nodes; *ii*) uniform thermal distribution	Performance degradation
Xie *et al.*[30]	Edge routing to mitigate the temperature from central nodes to edge nodes for uniform thermal distribution	Not suitable for complicated hotspot distribution
Rahmani *et al.*[31]	*i*) Thermal-aware routing strategy for bus-mesh hybrid 3D NoC; *ii*) packets are routed to the layers closer to the heat sink	*i*) Limited adaptation; *ii*) traffic increases in the layers closer to the heat sink
Daneshtalab *et al.* [32]	*i*) AntNet routing algorithm for even distribution of packets; *ii*) possibility of hotspots reduced; *iii*) routing path is chosen based on local temperature and throughput	Different traffic scenarios are not taken into account

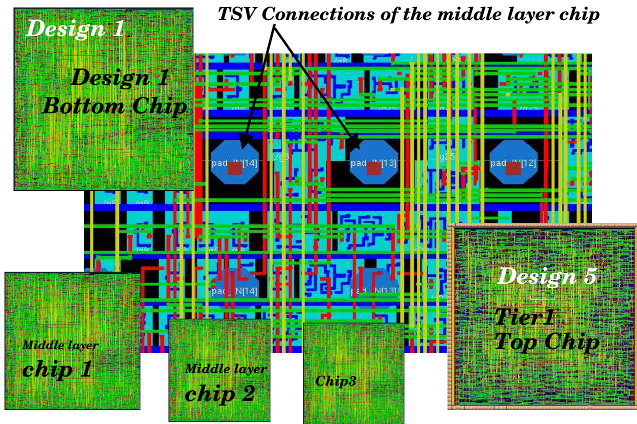


Fig. 1: 3D NoC Chips layers with TSV placement for NoCs

node's temperature and it's energy consumption. T_{ET} can be expressed as $T_{ET} = \theta \times E_{total}$, where E_{total} includes the total energy consumed by the node 'n' during system operation, $\theta = 1$ °C J^{-1} indicates the node temperature will increase 1 °C for every 1 J of energy consumed. T_{TC} can be calculated taking the thermal resistance coefficient into account,

$$T_{TC} = \sum R_{ij} \times \triangle T_{ij} \tag{2}$$

where, R_{ij} is the thermal conductivity between node i and j, and $\triangle T_{ij}$ is the temperature difference between node i and j. Any NoC dissipates its energy on (i) routing and arbitration, (ii) forwarding and receiving of flit, (iii) clock and, (iv) waiting of header flit. Among the mentioned energy spent on a flit, forwarding a flit is the dominant one on the router. It can be formulated as:

$$E_{ff} = E_{br} + E_{cb} + E_{lt} \tag{3}$$

where E_{ff} is the forwarding energy, E_{br} is the energy spent in reading a buffer, E_{cb} is the energy spent on crossbar, E_{lt} is the energy spent on traversing a link. Energy consumed at the IP core, E_{ip}, in the tile is highly affected by its communication energy, E_{comm}, similar to the work in [17]. E_{ip} changes dynamically in accordance to local data transfer, i.e., from/to the local router. Hence, E_{ip} can be calculated as;

$$E_{ip} = \beta \times E_{comm} \tag{4}$$

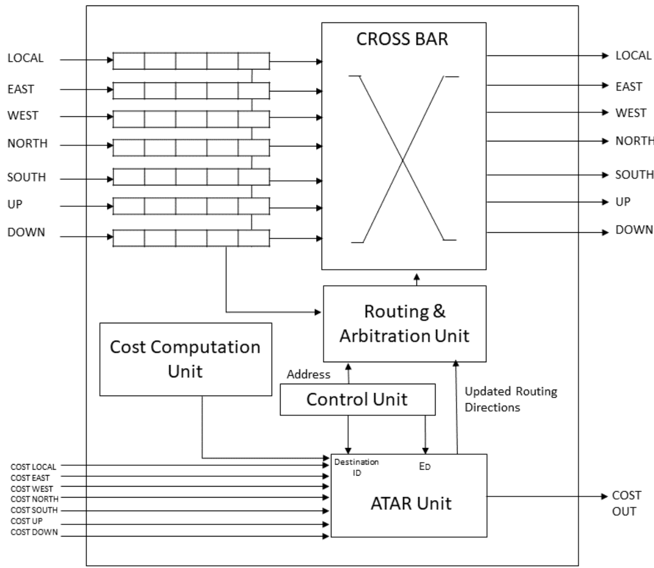


Fig. 2: Router Micro-architecture

where β is the communication to computation power ratio of the IP core. β is computed dynamically by taking cycle-accurate simulation result of individual IP and power profile as given in[33].

Using Eqs. 1-4, the temperature of the node 'n' can be expressed as,

$$T_{node} = T_n^0 + \theta \times E_{total} + \sum R_{ij} \times \triangle T_{ij} \tag{5}$$

5. Router Micro-architecture

Figure 2 illustrates the router architecture for adaptive thermal-aware routing. The proposed router micro-architecture is modification of 2D router architecture [23] to support the 3D NoC. The following are the addition to the architecture of [23] to support 3D NoC:

1) **Two channels:** Two channels: (i) *Up*, and (ii) *Down* are added to support the transmission of packets to the upper and lower layers respectively.
2) **Cost computation unit:** It provides local cost to the ATAR computational unit. The local cost is computed using four parameters: (i) path length, (ii) link workload, (iii) next router's queue length, and (iv) next router's temperature. These parameters are detailed in Section 6. The local cost and the costs generating from upward routers are propagated to all downstream routers. Destination address is scanned with the help of a synchronous counter. It provides an address reference to the routing table present inside the router to update the decision related to the destination block.
3) **ATAR unit:** This unit computes the cost of the path. ATAR units are connected using a dynamic network. In case of a dynamic network, Load on network changes over time. The optimal path computation needs minimum steps to estimate and equate the costs originated from upstream routers. This unit checks network status to make routing decisions. It involves two steps:
   a) **Routing function**: It evaluates the selected output channels towards which a packet can be dispatched to arrive at the destination.
   b) **Selection function**: It examines the channels availability status by checking the reservation table, and selects one output channel from the admissible set of channels provided by the routing function. Then, the ATAR unit enables the selected direction signal.
4) **Routing and arbitration unit:** The routing algorithm is implemented by this unit. It sets the necessary switches in the crossbar based on the routing decision and forwards the incoming messages to an output link. The arbitration unit resolves the situation when multiple input channels request access to the same output channel.
5) **Control unit:** The control unit block is comprised of combinational and sequential circuits, implemented using logic gates and a synchronous counter. The counter retrieves the address reference from the routing table. The ATAR unit uses this reference to update the destination decisions in the routing table.
6) **Crossbar:** This is a collection of switches arranged such that input and output lines form interconnecting lines. A connection can be established by by closing the corresponding switches at the necessary intersection.

The optimal routing direction is calculated by taking two parameters into account: (i) availability of the channel, and (ii) cost incurred if the packet follows the given direction. If there is a single available channel, irrespective of the cost, the router sends the packet in this direction, as waiting for other channels to be released will add extra delay. If multiple directions are available, then the router will send the packets in the direction having minimum cost. The cost function is described in Eqn. 8. In case of non-availability of any output channel, the router has to wait until a channel is released.

6. Adaptive Thermal Aware Routing Algorithm

The proposed ATAR algorithm is described in this section for thermal-aware routing. ATAR approach attempts to optimize the following four parameters:

1) **Path length (L):** The effective number of hops toured by a packet to arrive its destination. The path length between a source FU at (x_1, y_1, z_1) and destination FU at (x_2, y_2, z_2) is given by:

$$L_P = 10 \mid (x_1 - x_2) \mid + 10 \mid y_1 - y_2 \mid + 3 \mid z_1 - z_2 \mid \tag{6}$$

where $x_i \in \{1, 2, ...m\}$ is the row, $y_i \in \{1, 2, ...n\}$ is the column and $z_i \in \{1, 2, ...p\}$ is the layer, where the FU is situated; *m* and *n* are the number of FUs along the length and width of each layer, respectively, and *p* is the number of layers in the 3D NoC. Rows are numbered from the southernmost row, columns are numbered from the westernmost column, and layers are numbered from the bottommost layer. For an $8 \times 8 \times 4$ homogeneous mesh architecture, the value of $m = n = 8$ and $p = 4$.

TABLE II: Simulation Parameters

Parameter	Value
Packet size	$2 \sim 10$ flits
Buffer size	16 flits
Simulation time	10^4 cycles
Traffic pattern	Random, shuffle, transpose

2) **Link workload (W):** Link load is the sum of the load of all channels through the links:

$$W_L = \sum_{\forall link \in LINK_n} C_{link}(n) \quad (7)$$

where LINK_n = {E, W, N, S, U, D, Local}, and it presents the links in the direction of East (E), West (W), North (N), South (S), Up (U), Down (D), and Local.

3) **Next router's queue length (Q):** Queue length of the routers on the path to destination. A router with smaller queue length is given more priority as the next hop for the packet. The cost computation unit monitors and maintains the queue lengths of all the routers.

4) **Next router's temperature (T):** Temperature of the routers on the path to destination. A router with lower temperature is given higher preference as the next hop router. Router temperature is gathered from the thermal analyzer and the temperature sensors embedded in the cost computation unit.

We have considered the NoC as a connected, simple, non-planar, symmetric, directed graph. If there exists a directed edge from FU A to FU B then there also exists a directed edge from FU B to FU A. A weighted sum approach is used to assign the cost to each edge. The cost of an edge depends on the parameters ⟨L, T, Q, W⟩. The cost of an edge i is given by:

$$WC_i = \alpha_1 L_i + \alpha_2 T_i + \alpha_3 Q_i + \alpha_4 W_i \quad (8)$$

where $i = 1, 2, ..., n.$ $\alpha_1, \alpha_2, \alpha_3$ *and* α_4 are the weights assigned to L, T, Q, and W, respectively, such that $\Sigma_j \alpha_j = 1$ and $\alpha_j \neq 0, \forall j \in \{1, 2, 3, 4\}$. The proposed ATAR algorithm is stated in Algorithm 1.

Algorithm 1 takes the following arguments: n_{src} (the source FU), n_{dest} (the destination FU), *V* (cost matrix), and *P* (array of FUs to traverse to reach n_{dest}). The cost matrix *V* gives the cost of visiting an FU from any other FU. Initially, if two FUs are not neighbors, the corresponding value is set to infinity.

Assuming a multi-source single destination, the cost computed from the upstream routers is used by each ATAR unit as input. It then adds the cost of each edge to its neighbors and transmits the computed cost to the downstream router. The cost of each edge is the weighted sum cost of the four parameters (i.e., L, T, Q, and W) and hence, the shortest path to the destination is the one with the minimal value of the weighted sum of these four parameters.

The packet from the current FU can be redirected to any of its neighbors. That is the packet can be transmitted in all directions: East, West, North, South, Up, Down, and Local. However, this can cause a deadlock in the 3D mesh architecture. Deadlock can be prevented by blocking some of these directions a packet can take from the current FU. This blockage prevents the formation of waiting cycles. Therefore, the routing algorithm should consider only those possible directions in which packets can be forwarded while determining the optimal path from a source FU to a destination FU.

Algorithm 1 *Adaptive Thermal Aware Routing (ATAR) Algorithm*

1: **Define:**
 n$_{src}$**:** source node, **n**$_{dest}$**:** destination node
 V[][]: cost matrix; initially set to infinite (INF)
 P: computed path list, denoting the nodes in the shortest path from n_{src} to n_{dest}
 visited[]: boolean array denoting nodes that are already visited; 1 denotes nodes that are already visited and 0 denotes nodes yet to be visited.
 dir: set of directions, **par:** parallel operation
 getAvailableDirections(n): returns set of all available directions from node n
 m.getEdge(k): returns the edge from node m along direction k
 e.next: denotes the node for which the edge 'e' is incoming
 minCostNode(V, n$_{src}$**):** returns the node with minimum current cost value
 m, k, e, cost: temporary variables
2: **procedure** pathCost(n_{src}, n_{dest}, V, P)
3: **if** $n_{src} = n_{dest}$ **then**
4: **set** V[n_{src}][n_{dest}] ← 0, P[n_{dest}] ← P[n_{src}]
5: **else**
6: **set** m ← n_{src}, visited[] ← **FALSE**
7: **while** m ≠ n_{dest} **do**
8:   **set** dir ← $getAvailableDirections$(m)
9:   **par for all** k ∈ dir **do**
10:    **set** e ← m.$getEdge$(k)
11:    **set** cost ← $\alpha_1 \cdot e.T + \alpha_2 \cdot e.L + \alpha_3 \cdot e.Q + \alpha_4 \cdot e.W$
12:    **if** V[n_{src}][m] + cost < V[n_{src}][e.next] **then**
13:     **set** V[n_{src}][e.next] ← cost, P[e.next] ← m
14:    **end if**
15:   **end par for all**
16:   **set** visited[m] ← **TRUE**
17:   **set** m ← $minCostNode$(V, n_{src}, visited)
18: **end while**
19: **end if**
20: **return** [V[n_{src}][n_{dest}], P]

Given two FUs, n_{src} and n_{dest}, the algorithm first checks whether the two FUs are same. If so, the cost matrix *V* is updated (cost is set to zero), the path list *P* is updated and returned in lines 3-5. Otherwise, all the possible directions are computed from the current FU as shown in line 10. The ATAR unit considers only the candidate neighbors returned by the *getAvailableDirections()* procedure. In lines 11-13, the corresponding edges are obtained for each available direction, and the weighted sum cost is calculated. If the calculated cost is less than the current cost, the cost matrix is updated in lines 14-15. The previous node of the current neighbor in the shortest path from n_{src} is also set in line 16. The current FU,

m, is marked as visited in line 19, and its value is updated to the next unvisited FU with the least computed cost from the source FU in line 20. The nodes marked visited will never be checked again for cost computation or direction selection. The process is repeated until the current FU is same as the destination FU. In line 23, the path list and the total cost from the source FU to the current FU is then returned. In order to reduce the delay caused by the loop in the algorithm, parallel architecture is used to reduce the computation delay.

To avoid deadlock and livelock, 2D odd-even turn model [36] is enhanced to support 3D NoC in ATAR. It analyzes directions in which packets can turn in the network in order to break the cycle. It does this by prohibiting various turns for odd and even layers in the network. The rules are stated as follows:

1) West-North and East-North turns are restricted in odd layers.
2) South-West and South-East turns are blocked in even layers.
3) XY-down turns are restricted in an odd-XY plane and up-XY turns are prohibited in an even XY-plane.

7. Results and Discussion

In this section, we outline the details of simulation performed to compare ATAR with existing routing algorithms for 3D NoCs.

A. Simulation Set-up and Tools

For evaluating the performance, we have considered a $8 \times 8 \times 4$ 3D NoC. Simulations were carried out using AccessNoxim simulator. Three types of traffic patterns: transpose1, bit reversal, and random available in AccessNoxim simulator were considered. In transpose1, if the source node is at location *(i, j, k)* then the destination will be at the location *(m − i, n − j, p − k)* for a $m \times n \times p$ 3D NoC. For a $8 \times 8 \times 4$ 3D NoC considered for simulation, the destination will be at the location *(8−i, 8−j, 4−k)*. In random, the source and destination nodes are selected randomly. In bit reversal, the bit pattern of the source and destination addresses are reversed. For example if a_0, a_1, a_2, a_3 are the bit pattern of source address then the destination address has a bit pattern a_3, a_2, a_1, a_0. We have simulated the existing thermal-aware 3D routing algorithms such as *DTBR* [23] and *TTAR* [13], and thermal-aware fully adaptive routing [37] with our proposed ATAR algorithm. The rate of packet injection was varied from 0.02 to 0.22 flits/cycle/node at an interval of 0.02 flits/cycle/node. Each iteration was run for 200000 cycles with a warmup of 10000 cycles. Warmup cycles allow stabilization of network.

B. Thermal Results

Simulations were run until the on-chip temperature stabilizes. The thermal results are shown in Figure 3 for packet injection rate *(PIR)* of 0.02 flits/cycle for random traffic pattern. *PIR* $(0 < PIR \leq 1)$ is the rate at which packets are injected into the network. An *PIR* of *0.02* means (packets/cycle/node) i.e., each node sends 0.02 packets every clock cycle. The instant at which a packet is injected depends on the distribution of the interval times. The thermal figures show a drop in on-chip peak temperature of ATAR to thermal-aware fully adaptive routing, *DTBR*, and *TTAR* is about 44 K, 33 K, and 37 K, respectively. This implies ATAR achieves better thermal optimization compared to other routing algorithms considered for comparison. To guarantee the accuracy of results, the simulation at each PIR has been repeated a number of times.

Figure 3a shows that the fully adaptive routing has a high-temperature thermal profile since it does not take into account the temperature parameters during routing. *DTBR* was originally proposed for 2D NoCs. Therefore, while mapping it to 3D NoCs, DTBR is not able to cope up with the 3D NoC architecture, as shown in Figure 3b. In the case of *TTAR*, the temperature imbalance occurs as a consequence of network traffic being routed through high-temperature regions with low congestion. Figure 3c also illustrates the high degree of thermal coupling between stacked dies. ATAR gives better thermal profile in comparison to the other routing algorithms taken into consideration. Figure 3d illustrates the thermal profile of ATAR at PIR *0.02*. At higher PIR, the peak on-chip temperature of the layers increases about *11 K* (for PIR = 0.1) and *17 K* (for PIR = 0.22) with respect to the thermal map of ATAR at PIR = 0.02.

C. Performance Results

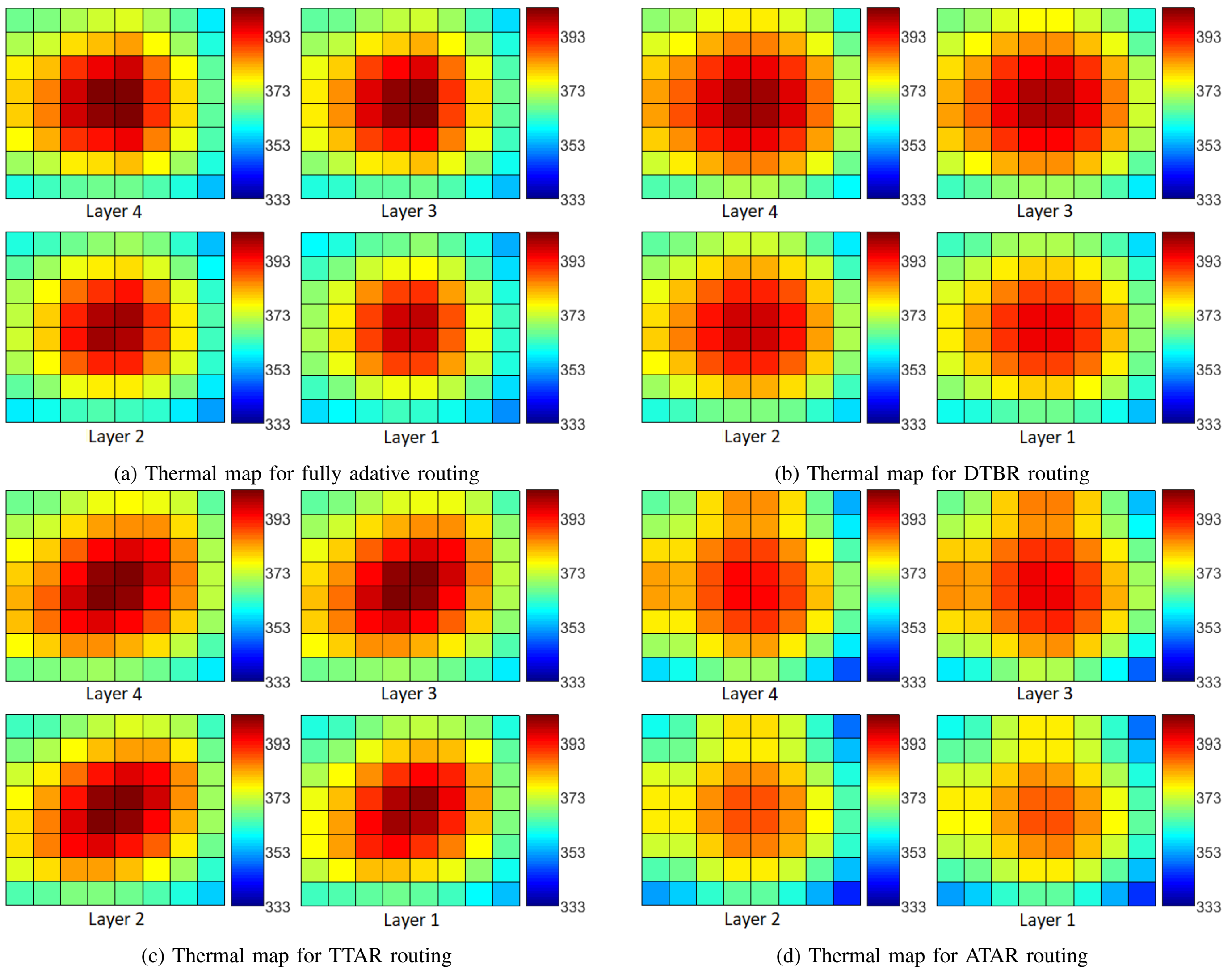
For analyzing the performance, we have considered the average packet delay. The comparison of average packet delay for different traffic pattern is shown in Figure 4a, Figure 4b, and Figure 4c. It is observed that the delay is less in ATAR for all the considered pattern.

In case of fully adaptive routing, there are many paths for a packet to follow, leading to a greater path diversity. This causes an increased delay in the fully adaptive routing. DTBR was originally proposed for 2D NoCs. However, scaling it for 3D NoCs increases the latency and temperature. TTAR is a congestion-aware routing strategy, which offers significantly low latency because it can avoid high traffic regions. However, it results in routing of packets close to interconnect-independent thermal hotspots. This causes the temperature to rise.

In ATAR, routing of packets is partially-adaptive in nature, which considers four different parameters for packet routing. ATAR also takes into account the queue length of various routers and link workload. This ensures that packets remain on a less congested path. The comparison among fully adaptive routing, DTBR, TTAR, and ATAR is shown in Figure 4a, Figure 4b, and Figure 4c for bit-reversal, random, and transpose1 traffic patterns, respectively.

8. Conclusion

NoC has been extensively used to handle the perplexing communication for future many-core systems. In this paper, an adaptive thermal-aware routing algorithm is developed to distribute the traffic more uniformly across the chip for uniform thermal distribution in the chip. ATAR makes the decision based on the weighted sum approach by taking



(a) Thermal map for fully adative routing

(b) Thermal map for DTBR routing

(c) Thermal map for TTAR routing

(d) Thermal map for ATAR routing

Fig. 3: Thermal maps with random traffic

input parameters like path length, next router temperature, next router queue length and link workload into account. Distributed ATAR units are coupled using a dynamic network to regulate the routing workload and results in minimization of thermal hotspots. In ATAR, cost computation and direction selection takes place in accordance with a deadlock-free turn model. Performance evaluation is carried out by using a cycle-accurate simulator taking different traffic into account. We have simulated the proposed ATAR algorithm along with three existing algorithms namely Fully adaptive, DTBR, and TTAR. From the analysis of simulation results, it was found that ATAR performs better in terms of thermal management and average packet delay. This work tackles a major problem in future 3D many-core systems with an efficient thermal-aware solution.

References

[1] B. Hoefflinger, *ITRS: The International Technology Roadmap for Semiconductors in Chips 2020*, Springer, 2011, pp. 161–174.

[2] , L. Benini and G. De Micheli, *Networks on Chips: A New SoC Paradigm*, Computer, vol. 35, no. 1, pp. 70–78, IEEE, 2002.

[3] W. J. Dally and B. Towles, *Route Packets, Not Wires: On-Chip Interconnection Networks*, Design Automation Conference, 2001. Proceedings. IEEE, 2001, pp. 684–689.

[4] A. Y. Weldezion, M. Grange, D. Pamunuwa, Z. Lu, A. Jantsch, R. Weerasekera, and H. Tenhunen, *Scalability of Network-on-Chip communication architecture for 3-D meshes*, 3rd ACM/IEEE International Symposium on. NoCS 2009. IEEE, 2009, pp. 114–123.

[5] H. S. Bennett, *International Technology Roadmap for Semiconductors Radio Frequency and Analog/Mixed-Signal technologies*, 2014 INTERNATIONAL TECHNOLOGY ROADMAP, 2017.

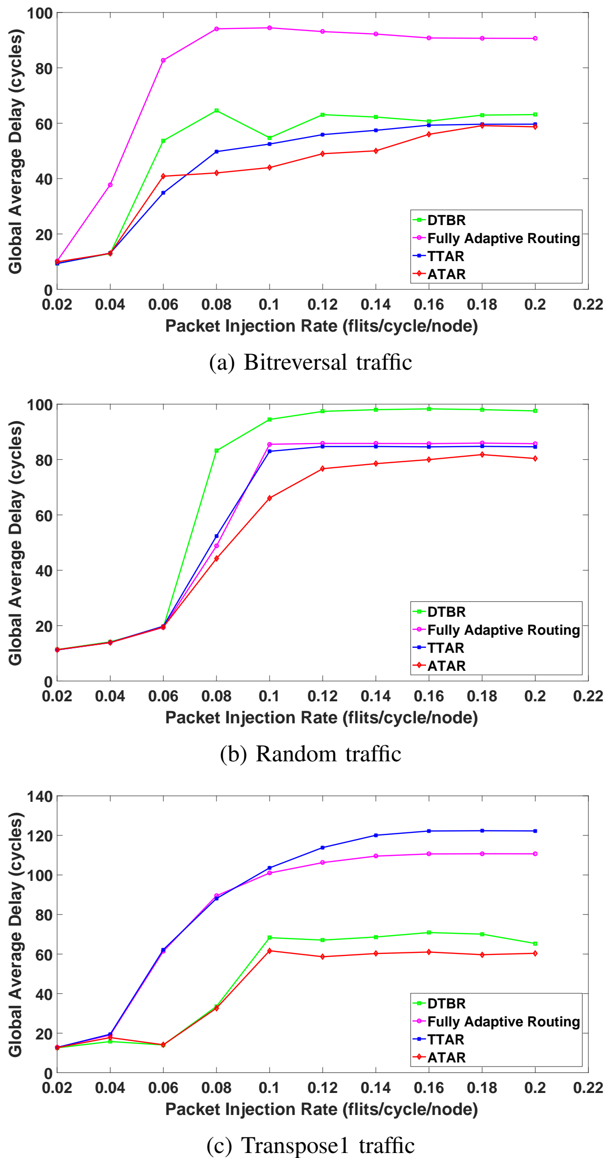
[6] M. Sadri, A. Bartolini, and L. Benini, *Single-Chip Cloud Computer Thermal Model*, Thermal Investigations of ICs and Systems (THERMINIC), 17th International Workshop on. IEEE, 2011, pp. 1–6.

[7] S. Borkar, *Thousand Core Chips: A Technology Perspective*, Proceedings of the 44th annual Design Automation Conference. ACM, 2007, pp. 746–749.

[8] L. Shang, L.-S. Peh, A. Kumar, and N. K. Jha, *Thermal Modeling, Characterization and Management of on-Chip Networks*, Proceedings of the 37th annual IEEE/ACM International Symposium on Microarchitecture. IEEE Computer Society, 2004, pp. 67–78.

[9] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, *A Network on Chip Architecture and Design Methodology*, VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on. IEEE, 2002, pp. 117–124.

[10] S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain et al., *An 80-Tile Sub-100w Teraflops Processor in 65-nm CMOS*, IEEE Journal of solid-state circuits, vol. 43,



(a) Bitreversal traffic

(b) Random traffic

(c) Transpose1 traffic

Fig. 4: Delay comparisons between various routing techniques with different traffic patterns

no. 1, pp. 29–41, 2008.

[11] R. Dash, J. L. Risco-Martin, A. K. Turuk, V. Pangracious, J. L. Ayala, and A. Majumdar, *A Bio-Inspired Hybrid Thermal Management Approach for Three-Dimensional Network-on-Chip Systems*, IEEE Transactions on NanoBioscience, 2017.

[12] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, *Temperature-aware Microarchitecture*, Computer Architecture, 2003. Proceedings. 30th Annual International Symposium on. IEEE, 2003, pp. 2–13.

[13] S. Lin, T.-C. Yin, H.-Y. Wang, and A.-Y. Wu, *Traffic and Thermal-aware Routing for Throttled Three-dimensional Network-on-Chip Systems*, VLSI Design, Automation and Test (VLSI-DAT), 2011 International Symposium on. IEEE, 2011, pp. 1–4.

[14] M. B. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffman, P. Johnson, J.-W. Lee, W. Lee et al., *The Raw Microprocessor: A Computational Fabric for Software Circuits and General-purpose Programs*, IEEE micro, vol. 22, no. 2, pp. 25–35, 2002.

[15] H. Ying, A. Jaiswal, and K. Hofmann, *Deadlock-free Routing Algorithms for 3-Dimension Networks-on-Chip with Reduced Vertical Channel Density Topologies*, High Performance Computing and Simulation (HPCS), 2012 International Conference on. IEEE, 2012, pp. 268–274.

[16] F. Dubois, A. Sheibanyrad, F. Petrot, and M. Bahmani, *Elevator-first: A Deadlock-free Distributed Routing Algorithm for Vertically Partially Connected 3D-NoCs*, IEEE Transactions on Computers, vol. 62, no. 3, pp. 609–615, 2013.

[17] C.-H. Chao, K.-Y. Jheng, H.-Y. Wang, J.-C. Wu, and A.-Y. Wu, *Traffic and Thermal-aware Run-time Thermal Management Scheme for 3D NoC Systems*, Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on. IEEE, 2010, pp. 223–230.

[18] C.-H. Chao, T.-C. Yin, S.-Y. Lin, and A.-Y. Wu, *Transport Layer Assisted Routing for Non-stationary Irregular Mesh of Thermal-aware 3D Network-on-Chip Systems*, SOC Conference (SOCC), 2011 IEEE International. IEEE, 2011, pp. 284–289.

[19] R. Salamat, M. Khayambashi, M. Ebrahimi, and N. Bagherzadeh, *A Resilient Routing Algorithm with Formal Reliability Analysis for Partially Connected 3D-NoCs*, IEEE Transactions on Computers, vol. 65, no. 11, pp. 3265–3279, 2016.

[20] E. Taheri, A. Patooghy, and K. Mohammadi, *Cool elevator: A Thermal-aware Routing Algorithm for Partially Connected 3D NoCs*, Computer and Knowledge Engineering (ICCKE), 2016 6th International Conference on. IEEE, 2016, pp. 111–116.

[21] X. Jiang, X. Lei, L. Zeng, and T. Watanabe, *High Performance Fully Adaptive Runtime Thermal-aware Routing Algorithm for 3D NoC*, International MultiConference of Engineers and Computer Scientists. Springer, 2016, pp. 31–48.

[22] C.-H. Chao, K.-C. Chen, and A.-Y. Wu, *Routing-based Traffic Migration and Buffer Allocation Schemes for 3-D Network-on-Chip Systems with Thermal Limit*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 11, pp. 2118–2131, 2013.

[23] F. Liu, H. Gu, and Y. Yang, *DTBR: A Dynamic Thermal-balance Routing Algorithm for Network-on-Chip*, Computers & Electrical Engineering, vol. 38, no. 2, pp. 270–281, 2012.

[24] X. Jiang, X. Lei, L. Zeng, and T. Watanabe, *High Performance Virtual Channel Based Fully Adaptive Thermal-aware Routing for 3D NoC*, Quality Electronic Design (ISQED), 2017 18th International Symposium on. IEEE, 2017, pp. 289–295.

[25] P. Lotfi-Kamran, A.-M. Rahmani, M. Daneshtalab, A. Afzali-Kusha, and Z. Navabi, *EDXY–A Low Cost Congestion-aware Routing Algorithm for Network-on-Chips*, Journal of Systems Architecture, vol. 56, no. 7, pp. 256–264, 2010.

[26] H.-K. Hsin, E.-J. Chang, K.-Y. Su, and A.-Y. A. Wu, *Ant Colony Optimization-based Adaptive Network-on-Chip Routing Framework using Network Information Region*, IEEE Transactions on Computers, vol. 64, no. 8, pp. 2119–2131, 2015.

[27] AccessNoxim 0.3, http://access.ee.ntu.edu.tw/noxim/index.html

[28] Y.-Y. Chen, E.-J. Chang, H.-K. Hsin, K.-C. J. Chen, and A.-Y. A. Wu, *Path-diversity-aware Fault-tolerant Routing Algorithm for Network-on-Chip Systems*, IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 3, pp. 838–849, 2017.

[29] Y.-S. Lee, H.-K. Hsin, K.-C. Chen, E.-J. Chang, and A.-Y. A. Wu, *Thermal-aware Dynamic Buffer Allocation for Proactive Routing Algorithm on 3D Network-on-Chip Systems*, VLSI Design, Automation and Test (VLSI-DAT), 2014 International Symposium on. IEEE, 2014, pp. 1–4.

[30] M. Xie, M. Yin, S. Li, and Y. Li, *A Study on Thermal Impacts of Routing Algorithms for 3D Mesh-based Network-on-Chip*, IET, 2012.

[31] A.-M. Rahmani, K. R. Vaddina, K. Latif, P. Liljeberg, J. Plosila, and H. Tenhunen, *Design and Management of High-performance, Reliable and Thermal-aware 3D Networks-on-Chip*, IET circuits, devices & systems, vol. 6, no. 5, pp. 308–321, 2012.

[32] M. Daneshtalab, A. Sobhani, A. Afzali-Kusha, O. Fatemi, and Z. Navabi, *NoC Hotspot Minimization Using Antnet Dynamic Routing Algorithm*, Application-specific Systems, Architectures and Processors, 2006. ASAP'06. International Conference on. IEEE, 2006, pp. 33–38.

[33] A. B. Kahng, B. Lin, and S. Nath, *Orion3.0: A Comprehensive NoC Router Estimation Tool*, IEEE Embedded Systems Letters, vol. 7, no. 2, pp. 41–45, 2015.

[34] V. Pangracious, E. Amouri, H. Mehrez, and Z. Marrakchi, *Physical Design Exploration of 3D Tree-based FPGA Architecture*, Proceedings of the 23rd ACM international conference on Great lakes symposium on VLSI. ACM, 2013, pp. 335–336.

[35] L. Wilson, *International Technology Roadmap for Semiconductors (ITRS)," Semiconductor Industry Association*, 2013.

[36] G.-M. Chiu, *The Odd-Even Turn Model for Adaptive Routing*, IEEE Transactions on parallel and distributed systems, vol. 11, no. 7, pp. 729–738, 2000.

[37] J. Wang, H. Gu, Y. Yang, and K. Wang, *An Energy and Buffer-Aware Fully Adaptive Routing Algorithm for Network-on-Chip*, Microelectronics Journal, vol. 44, no. 2, pp. 137–144, 2013.