Q-Thermal: A Q-Learning-Based Thermal-Aware Routing Algorithm for 3-D Network On-Chips

Narges Shahabinejad and Hakem Beitollahi

***Abstract*— The multilayer structure of 3-D network-on-chips (3-D NoC) leads to the unequal thermal conductance between different layers and, as a result, an unbalanced thermal distribution in the chip. This issue leads to the low reliability and performance degradation of 3-D NoCs. To ensure thermal safety, 3-D NoCs require effective cooling methods. Several thermal-aware routing algorithms have been proposed to route packets in a way which helps mitigation of imbalance temperature distribution. To have a thermal-aware path selection and safety thermal distribution, we propose a new thermal-aware routing method based on the Q-learning algorithm (Q-Thermal). In our method, thermal information propagates over the network using ordinary network packets. Thanks to the Q-learning algorithm that keeps always the thermal information of routers up-to-date that causes routers steer packets toward the cooler paths (laterally or vertically). Based on the experimental results, Q-Thermal reduces the heat and temperature of different layers of 3-D NoCs significantly in comparison with the previous state-of-the-art techniques. Q-Thermal leads to a balanced thermal distribution in the stacked layers of the chip and it improves the network performance considerably.**

***Index Terms*— 3-D network-on-chip (3-D NoC), packet routing, Q-learning, Q-routing, thermal management.**

I. Introduction

DUE to the technology scaling, it is possible to integrate thousands of cores in a single chip. Network-on-chip (NoC) has been proposed as an efficient structure to solve communication bottlenecks in the integrated circuits (ICs) [1]. A 3-D network-on-chip (3-D NoC) is an efficient combination of 2-D NoC and 3-D integration technology [2]. 3-D NoCs can achieve lower latency and higher performance due to the shorter connection length, higher bandwidth, and more flexible routing options than 2-D NoCs [3].

Regardless of these evident benefits, 3-D NoCs face with essential thermal challenges [2], [4]. The power density of the chip increases as more dies stacked vertically. Furthermore, as shown in Fig. 1, the longer heat conduction path and the heat sink location cause an unbalanced thermal distribution in the chip [5]. For instance, the farther layers from the heat sink experience more heat and temperature. These thermal drawbacks in 3-D NoCs can increase leakage power, and temperature that impose longer propagation delay, and create several hotspots. This issue leads to performance and reliability degradation [6].

Manuscript received December 1, 2019; revised July 18, 2020; accepted July 25, 2020. Date of publication August 20, 2020; date of current version September 21, 2020. Recommended for publication by Associate Editor N. Sabry upon evaluation of reviewers' comments. *(Corresponding author: Hakem Beitollahi.)*

The authors are with the School of Computer Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran (e-mail: beitollahi@iust.ac.ir).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCPMT.2020.3018176

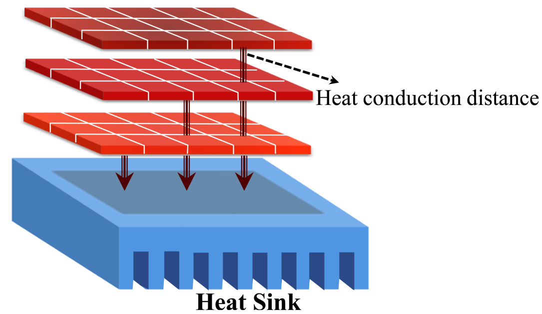


Fig. 1. Thermal conduction paths from different layers in a 3-D mesh.

The temperature of 3-D NoCs is the outcome of both the processing elements and the interconnection network. The power consumption as a result of communications has a significant impact on the overall chip power density due to the high switching activity [7]. On the other hand, a many-core system architecture tends to integrate a large number of simple cores instead of integrating a few complex cores [8]. The result is that the weight of the power consumption due to communications overcomes the same issue due to processing elements.

Cooling techniques in 3-D NoCs locate into two categories: the technology level and the algorithmic/architectural level. In the former case, extra equipment is added to the chip to eliminate hotspots. It is an efficient technique but suffers from high area and manufacturing costs [9]. The latter case proposes a runtime thermal management (RTM) technique to keep the system temperature below a certain thermal limit. In this category, the mentioned overheads are negligible. Hence, many kinds of research are done in this domain.

The thermal management techniques [10] in the algorithmic/architectural level fall into two categories: reactive and proactive. The reactive techniques temporarily either stop or throttle the activity of the overheated routers. In these techniques, for each router, a thermal sensor is embedded into the chip [11]. As soon as the temperature of a router exceeds a thermal threshold, the activity of the router is stopped (or limited) until the router's temperature falls down again [7]. The main challenge of this category is how to route packets in the network with lots of slept (throttled) routers. Without any doubt, the performance of such networks is reduced significantly. In the proactive techniques, creating hotspots and reaching to the critical thermal condition are prevented via subtle and smart routing techniques. In fact, the routing mechanism is done in a way that the traffic load is transferred to the cooler parts and the heat is balanced across the layers of the chip. As a result, none of the routers may reach the thermal threshold and forced to stop their activity. To have an accurate thermal-aware selection, routers should consider the thermal status of the network. Therefore, a thermal-aware routing algorithm needs to propagate thermal information obtained from thermal sensors in the entire network.

This article presents a new proactive thermal management technique for 3-D NoCs based on the Q-learning mechanism. Q-learning is a values-based reinforcement learning algorithm that is employed to make the best possible decision in multi-choices environments. During the system operation, an agent learns its own actions in an interactive environment using a feedback-based technique. Q-learning uses a table, named Q-table, to store and update the reward values of the agent actions. The experimental values help the agent to learn how to act optimally in each state of its operation.

In our method, few bits (e.g., 8 bits) of the header of each ordinary packet (a packet that transfers data between nodes) are assigned to carry the average temperature of routers that the packet traverses them. These values are stored and updated in the routers' Q-table. Consequently, the thermal information is propagated across the chip with no need of learning packets. The routers drive incoming packets to the cooler paths considering their Q-table values and detour them from high-temperature areas at an earlier time. To the best of our knowledge, we utilize Q-learning for the first time to present an efficient thermal-aware routing mechanism in 3-D NoCs. Though Q-learning has been used and updated as the name of Q-routing for congestion control in NoCs [12].

Simulation results show that our proposed Q-Thermal technique improves the network performance on average by 32% in comparison with PTB^{3}R [13] and TAAR [5], two well-known techniques in this domain. It can also reduce the standard deviation of temperature distribution by 13% and 28% with respect to PTB^{3}R and TAAR, respectively. In fact, our method balances the thermal distribution across the layers of a 3-D NoC effectively. Our analysis indicates that our method has negligible overheads and it can be considered as a suitable thermal-aware routing mechanism in the next generation of 3-D NoCs.

The rest of this article is organized as follows. In Section II, some related routing algorithms for balancing thermal distribution in the 3-D NoCs are reviewed and discussed. Section III provides background on Q-learning and Q-routing. The Q-Thermal technique is presented in Section IV. Section V shows and discusses the experimental results. Finally, this article is concluded in Section VI.

II. Related Works

In this section, some techniques from both reactive and proactive categories are reviewed.

A. Reactive Techniques

The main duty of researchers in this category is deciding how to route and steer traffic load when there are lots of slept routers. Authors in TAAR [5] have used a topology table for gathering throttling information. Using this table, the routing algorithm detours the traffic load from the throttled routers in the earlier steps. This algorithm also routes packets in the lower layers as cooler paths if it finds them necessary. In TTABR [14], nonminimal beltway paths on the network borders have been intended to mitigate the traffic load in the minimal paths and detours packets from congested areas and throttled nodes. QTTAR [15] improves heat and traffic distribution between layers of a 3-D NoC in two steps. In the first step, it collects all deadlock-free routing directions based on the topology and location of throttled nodes. In the second step, a policy is created based on the Q-function routable strategy to select a final direction (collected from the first step) toward the noncongested region. Taheri *et al.* [16] propose a reactive routing algorithm to dynamically bypass packets from hot zones containing throttled routers. The algorithm proposes two virtual networks to provide path diversity for packet routing in each layer of 3-D NoCs. The path diversity distributes heat generation to reduce the thermal variance.

B. Proactive Techniques

In proactive techniques, researchers try to balance temperature distribution in the network and prevent routers' temperature to exceed a critical thermal point. In some proactive methods, like [7] and [17], routing strategies rely on the network structure (i.e., location of routers). The more distance from the heat sink leads to less traffic load. Chao *et al.* [7] proposes a downward routing scheme to steer packets toward the bottom layer to avoid the throttled nodes. However, downward routing leads to unnecessary traffic movement and also extreme unbalance traffic distribution especially in the bottom layer. On the other hand, some techniques propose a thermal model to predict the thermal behavior of the system [13], [18]. For example, PTB^{3}R [13] has proposed a thermal index named mean time to throttled (MTTT) to control routers' temperature before overheating. In this method, MTTT comes along with the traffic information in TTABR [14] to have a thermal and traffic-aware choice between minimal and nonminimal beltway paths.

Pagracious *et al.* [19] have embedded a control unit in the network nodes to extract the optimal paths between cores. These units proceed parallel to the network activity. The control network has executed the Dijkstra's algorithm to minimize a weighted sum contained alternative parameters that their highest weight is assigned to the temperature. Sivakumar *et al.* [10] believe the thermal challenges of 3-D NoCs are due to increased wire length and routing paths between components. They propose a stochastic-based genetic algorithm to offer minimum wire length and routing paths by replacing the TSV positions. Recently, Cao *et al.* [20] provide an exhaustive survey on optimization techniques to alleviate the energy and thermal issues in the 3-D multiprocessors ICs.

Most of the techniques in the survey are from the proactive type.

III. Background on Q-Learning

Q-learning is a basic form of the reinforcement learning domain. This algorithm learns a policy that helps agents to act optimally in a specific environment. The agent passes several states in its traverse. Each state offers some actions with different costs. The agent tries to improve the quality of its actions in each state using reactions received from the environment.

Suppose that the agent takes action a in state s then, receives reward r, and moves to the next state s'. In this manner, $Q(s, a)$ is the estimated cost of doing action a in state s. Before the training phase, all Q-values are initialized by a predefined amount, which depends on their attributed parameters. These values are updated using the temporal difference rule as follows:

$$Q[s, a] \leftarrow Q[s, a] + \alpha(r + \gamma \, . \max_{\alpha'} Q[s', a'] - Q[s, a]). \quad (1)$$

In this equation, r is the reward of moving from state s to state s'. $\max_{\alpha'} Q[s', a']$ is the maximum reward estimated for future states. α is the learning rate or step size ($0 < \alpha < 1$) which determines how quickly the agent adopts the random changes in the environment. In fully deterministic environments, a learning rate of $\alpha = 1$ gives an optimal solution. In the stochastic scenarios, though assigning a small value to α will offer more control and a more accurate final answer, it will take more attempts to reach the optimal solution. Therefore, we encounter a tradeoff between converging to the best answer and the number of cycles required to converge. Eventually, γ that is called discount factor gives weight to future rewards.

The Q-learning method utilizes a table, named Q-table, to hold states, actions, and their relevant Q-values for each special goal. This table will be used and updated by the agents in their learning process.

In the earlier learning steps, action selection is normally based on random or exploration. By training progress and replacing optimal Q-values with primary ones, decisions are made exploitation or Q-table based.

A. Q-Routing

Q-routing [12] is a Q-learning based network-on-chip routing algorithm used to optimize a specific network parameter in the packet routing process. In each router, a Q-table is located to keep information attained from the learning process.

Fig. 2 shows a typical 2-D NoC based Q-table. Based on the table, the current state is the ID of the current routers. The action state-space indicates the possible choices for the agent to go to the next state. In a 2-D mesh, these actions are the output channels that can transmit network packets to their destinations (north, south, east, and west directions).

The next state-space is the ID of the next routers. The goal column consists of the ID of all network routers as packets destination. Q-values are the network parameters that the learning process aims to control. For each action and destination pair, there is a Q value in an NoC-based Q-table.

State-Space			Action-Space		Q-Value		Goal
s	s'		a		Q(s;a)		d
Current-Router	Next-Router		Output-Port		Delay		Destination-Router
Node 4	1	3	South	West			Node 0
Node 4	1	-	South	-		-	Node 1
Node 4	1	5	South	East			Node 2
Node 4	3	-	West	-		-	Node 3
Node 4	4	-	Local	-	-	-	Node 4
Node 4	5	-	East	-		-	Node 5
Node 4	3	7	West	North			Node 6
Node 4	7	-	North	-		-	Node 7
Node 4	5	7	East	North			Node 8

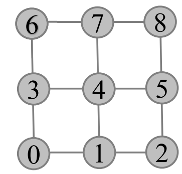


Fig. 2. Typical Q-table of node 4 for a 3×3 NoC mesh.

This amount expresses the estimated cost from the current router to the destination by taking the given action.

In [12], a Q-routing algorithm is applied to send the network packets to the lower traffic areas and prevent congestion in the network. In this method, whenever router x sends a packet to router y, router y sends a learning packet back to router x. The learning packet includes the following items. The node ID, which determines sender of the learning packet, the packet destination ID (d), the packet's waiting time in the neighbor router (as local latency) and global latency which demonstrates the estimated latency from the next node to the destination. The Q-value upgrading is done as follows:

$$Q_x(y, d)_{\text{new}} = Q_x(y, d)_{old} + \alpha(Q_y(z, d) + q_y - Q_x(y, d)_{\text{old}}) \quad (2)$$

where $Q_y(z, d)$ is the global latency. This amount is returned by router y as the most optimal cost, saved in its Q-table, to get to the destination d (i.e., the latency from z as a neighbor of y to d). The q_y parameter is equivalent to the local latency. At the end, router x overwrites the $Q_x(y, d)$ in its Q-table with the newly obtained value from router y. In our implementation, in a 50-50 weighting assignment to old and new information ($\alpha = 0.5$), the algorithm had the best response to dynamically changing network conditions. Lee and Han [15] through empirical experiments found that setting α to 0.6 brings the best response for their Q-learning machine.

IV. Q-Learning-Based Thermal-Aware Algorithm (Q-Thermal)

In this section, we introduce a novel thermal-aware routing method for a 3-D network on chips based on the Q-learning technique. The algorithm is detailed through five parts: 1) how to collect thermal information across the network; 2) how to do thermal-aware lateral routing; 3) how to update thermal information across the network; 4) how to do vertical proactive routing and finally; and 5) how to add a free deadlock mechanism to the thermal-aware routing algorithm.

A. Collecting Thermal Information

One important issue regarding to thermal-aware routing designs is propagating thermal information through the network routers. To do so, our approach inspires the Q-routing scheme as follows.

Action-Space		Q-Value		Goal
a		Q(s;a)		d
Output-Port		Avg-Temp (°C)		Destination-Router
South	West	25	35	Node 0
South	-	40	-	Node 1
South	East	35	40	Node 2
West	-	55	-	Node 3
Local	-	-	-	Node 4
East	-	43	-	Node 5
West	North	36	67	Node 6
North	-	57	-	Node 7
East	North	72	38	Node 8

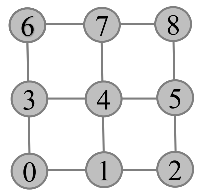


Fig. 3. Thermal-based Q-table of node 4 with imaginary temperatures.

Each router keeps a Q-table containing the goal, the actions, and the Q-value fields. In a 3-D NoC, the goal is the ID of nodes that are located in the same layer as the main router. It represents the packet's destination. The actions specify the lateral directions (north, south, east, and west), which can deliver the network packets into their destinations. Finally, Q-value is the estimated average temperature that a packet experiences from its source to the current router. In the beginning, all Q-values are initialized to the ambient temperature. Fig. 3 shows a thermal-based Q-table with imaginary temperatures.

According to the Q-routing algorithm, updating the Q-values and learning process were being done by the learning packets. Generating and propagating these packets and allocating a discrete virtual channel to them impose the unreasonable costs to the network.

In our method, during a packet transition process, each router calculates the new amount of Q-values and locates it in the packet's header before sending the packet to the next router. This value is the average temperature of the routers covered by the packet in its path from the source router to the current router. Whenever a router receives a packet, it extracts the average temperature from the packet's header to upgrade its Q-values relative to that path. In fact, the upgraded Q-values determines the average temperature from the packet's source s to the current node. Indeed, this information would be useful for the packets which will pass the opposite path of the current packet (from the current node to node s). Consequently, the learning packets are omitted. Hence, the values equivalent to the receiving node ID, the destination node ID, and the local latency in the main Q-routing algorithm are dropped from the network traffic. Moreover, the virtual channel capacity is used for the main network packets' transmission.

B. Thermal-Aware Lateral Routing

Suppose that packet p is transferring from source s to destination d. A router r receives the packet and checks the lateral output channels, which can carry the packet to the node correspond to node d in the current layer. If there were two output channels, the router makes the decision based on its Q-table. Therefore, the direction with the smaller Q-values (lower temperature) is chosen as the optimal output channel. If at least one of the Q-values has the initial temperature, it means that the learning process is still in the earlier stages. So the output will be selected randomly.

Algorithm 1 illustrates the pseudocode of the thermal-aware lateral routing technique in the Q-Thermal method. Table I shows the list of important abbreviations, which have been used in Algorithms 1 and 2.

TABLE I
LIST OF ABBREVIATIONS USED IN THE ALGORITHMS

Abbreviations	Comments
$Init\_Temp$	Initial average temperature
$(Nr\_out\_port)_{(r)}$	Number of output ports of router r
$Q\_table_{(r)}$(K)	Q-table of router r that extracts information of the row related to Src/Dest K
Channel i	Input channel i or output channel i of router r extracted from Q-table
Avg_Temp_i	The average temperature related to channel i
Pkt_Avg_Temp	The average temperature is carrying by the packet
$Local\_Temp$	The current router's temperature
Pkt_Hop_Num	The number of routers passed by the packet

C. Updating Thermal Information

One of the important router obligations in our approach is updating thermal information in its Q-table. Suppose that, router r receives packet p from input i. Router r extracts the average temperature (Pkt_Avg_Temp) from the packet's header, which denotes the average temperature packet p has visited during its path from its source to router r. Then the router saves Pkt_Avg_Temp as Q-values relevant to direction i for goal s. So that, if packets with destination s leave router r from output i, they will expect to experience that average temperature. Moreover, before sending the packet, the router should add its temperature to Pkt_Avg_Temp for updating the Q-table of the next router. Algorithm 2 shows the pseudocode of updating the thermal information in the proposed method. As the algorithm shows, the average temperature of a path is updated as follows:

$$\begin{aligned}\text{Pkt\_Avg\_Temp(new)} = &\left(\frac{\text{Pkt\_Hop\_Num} - 1}{\text{Pkt\_Hop\_Num}}\right)\\ &\times \text{Pkt\_Avg\_Temp(old)}\\ &+ \frac{1}{\text{Pkt\_Hop\_Num}} \times \text{Local\_Temp}\end{aligned} \tag{3}$$

where Pkt_Hop_Num represents the number of routers passed by the packet through its path, Pkt_Avg_Temp_old is the average temperature perceived by the packet until its prior state and Local_Temp shows the temperature of the current router.

When the temperature of routers is recorded into Q-tables via departed packets, the routers steer the packets of the opposite direction toward the cooler paths. Consequently, the Q-table of the routers on the cooler paths is updated regularly as they receive packets. However, the Q-table of routers on the hot paths is not updated as they do not receive packets. After a while of time, the temperature of the hot routers is degraded and the hot routers become cooler. A possible concern is that the Q-table of those routers is not upgraded. We believe this issue is not a challenge because when packets are steered toward the cooler paths, finally the

Algorithm 1 Pseudocode of the Proposed Lateral Path Selection Toward Destination D in Router r

```
INPUT: r = local id, Init_Temp = initial average temperature of router
OUTPUT: Output_Selection = The selected output port
Begin
/*Router r receives a new packet*/
S: Source address of the packet;
D: Destination address of the packet;
/* Optimal Path Selection*/

if C = D then Reach destination router;
    return; /*The packet has reached the destination. End of the algorithm*/

else if (Nr_out_port)_(r) to D = 1 then
    Output_Selection = Q_table_(r)(D) (Channel1);
/*If there is only one output toward the destination, the destination locates either in the same row or in the same column. That single output is selected. */

else if (Q_table_(r)(D) (Avg_Temp1) = Q_table_(r)(D) (Avg_Temp2) or (Q_table_(r)(D) (Avg_Temp1) = Init_Temp) or (Q_table_(r)(D) (Avg_Temp2) = Init_Temp) then
    Output_Selection = select random between Q_table_(r)(D) (Channel1) and Q_table_(r)(D) (Channel2);
/*If the temperature value for both output paths is the same; or one of the output paths has the initial temperature value, select the path randomly */

else if Q_table_(r)(D) (Avg_Temp1) < Q_table_(r)(D)) (Avg_Temp2) then
    Output_Selection = Q_table_(r)(D) (Channel1); /*If the temperature value of output channel 1 is less than output channel 2; channel 1 is selected */

else Output_Selection = Q_table_(r)(D) (Channel2); /*Otherwise, channel 2 is selected */

end if

End
```

Algorithm 2 Pseudocode of the Thermal Information Updating of q-Table of Router r

```
INPUT: r = local id, Local_Temp = local router temperature
Begin
/*Router r receives a new packet from S and extracts the Packet Average Temperature from the packet's header*/
S: The Source address of the packet;
Pkt_Avg_Temp: Packet Average Temperature;
Pkt_Hop_Num: Packet Hop Number;
Input_Channel:Packet Input Channel;
/* Q-table Upgrading*/

if Input_Channel = Q_table_(r)(S) (Channel1) then Avg_Temp_old = Q_table_(r)(S) (Avg_Temp1);
/*The previous temperature stored in the table is retrieved as Avg_Temp_old. */
Q_table_(r)(S) (Avg_Temp1) = Avg_Temp_old + 0.5(Pkt_Avg_Temp − Avg_Temp_old);
/*If the packet comes from input channel 1, the new temperature should be updated for channel 1 (current input, future output). The temperature is updated according to q-learning relation. */

else if Input_Channel = Q_table_(r)(S) (Channel2) then Avg_Temp_old = Q_table_(r)(S) (Avg_Temp2);
Q_table_(r)(S) (Avg_Temp2) = Avg_Temp_old + 0.5(Pkt_Avg_Temp − Avg_Temp_old);
/*The same procedure for channel 2 */

end if
/* Packet Average Temperature Upgrading*/
Pkt_Avg_Temp = ((Pkt_Hop_Num − 1)/Pkt_Hop_Num) × Pkt_Avg_Temp + (1/Pkt_Hop_Num) × Local_Temp; /* The temperature of the current router is added to the averaged temperature stored in the packet.*/

End
```

temperature of the routers on those paths becomes more than the recorded temperature for the previous hot routers. Hence, next time, traffic is routed via those paths that had been tagged as hot paths, and their Q-tables are upgraded.

The next issue is that, different paths are traversed via packets of different sources and destinations. As a result, the Q-table of the routers on short paths is upgraded faster. This issue assists the technique to have better selections when packets approach their destinations.

D. Vertical Proactive Routing

In a 3-D NoC, the location of the heat sink is normally in the bottom layer. As a result, it increases the temperature variance in different layers. So, the temperature of routers in the higher layer rises more rapidly because of their distance from the heat sink. This proves the importance of vertical thermal control by the thermal-aware routings through TSVs.

In our method, the information obtained from Q-tables helps the routers to prevent transmitting traffic load to the overheated areas. If the Q-values for a special destination that indicate temperature, exceed over the half of threshold temperature, it means that some routers in the minimal path of the current router to that destination may be close to the emergency state. In such situations, the routing method selects between the current layer and a lower layer (through a TSV) which possibly has a better thermal condition. The illustrated example in Fig. 4 shows how a network packet finds its way to the destination through a thermally safe path and detours the routers which are getting near to the threshold temperature. The network packet may move to one or two downward layers via TSVs and then move laterally toward the destination and finally move to upward layers via TSVs until reaching the destination. The probability of selecting the lower layer via a TSV increases linearly by the average temperature increment. If the average temperature rises up to the temperature threshold, the relative router sends packets to the lower layer by certain.

E. Deadlock Freedom

One of the most important challenges in an NoC routing is cyclic dependence among the packets which causes a deadlock in the network. On the other hand, the average

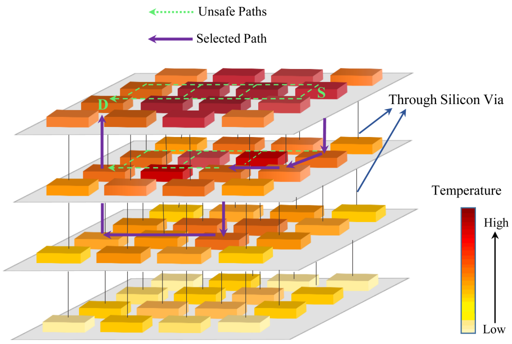


Fig. 4. Example of vertical proactive routing.

temperature calculated for a path will be used by the reverse path. Therefore, both paths should be able to be used by the packets without causing a deadlock in the network.

The virtual channels are one of the usual approaches for deadlock avoiding and performance improvement. Our proposed routing method has utilized two virtual channels. The first one is used for the packets whose sources' y dimension is smaller than their destinations' y dimension. Therefore, north-left, north-right, left-north, and right-north turns never happen for those packets. Moreover, the other virtual channel is used for the packets whose sources' y dimension is larger than their destinations' y dimension. So that south-left, south-right, left-south, and right-south turns will not be done. For preventing deadlock between network layers, up-lateral turns are avoided. Fig. 5 shows the packet transmission in both virtual channels and symmetrical paths for a typical source and destination.

F. Hardware Realization

As shown in Fig. 4, a fully connected mesh of a 3-D NoC topology is considered, in which every face-to-face routers in two adjacent layers are connected through a TSV. In fact, in a typical 3-D NoC mesh, which is our target topology, each router is connected at most to six neighboring routers (north, south, west, east, top, and bottom). Coskun *et al.* [21] has shown that, since total TSVs occupy a limited portion of the interface material area, they have a negligible effect on the silicon die's temperature.

The hardware realization of Q-Thermal is designed based on the flowchart presented in Fig. 6. Fig. 7 illustrates the abstract-level of the router's architecture in Q-Thermal that utilizes Q-table, Q-table updater, and thermal analysis in its architecture. The abstract level of the thermal analysis section is depicted in the figure.

G. Summary of Q-Thermal

In summary, the packets that are traversed for different destinations record the temperature of their paths in the next routers. This recorded temperature tells the routers if a packet is traversed in the opposite direction, it may experience this temperature. After a while, many packets are transferred in many paths. As a result, the temperature of many different paths from any point to any other point is stabled and recorded. Consequently, routers can drive packets to the cooler paths based on the recorded temperatures and avoid creating hotspots

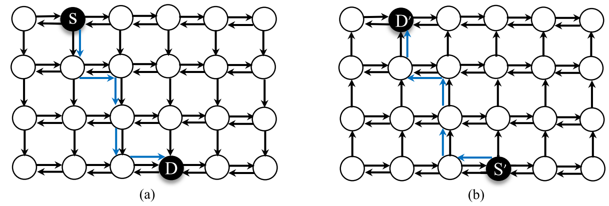


Fig. 5. (a) Traversed path in the first VC for pair (S, D). (b) Corresponding path in the second VC for pair (S', D').

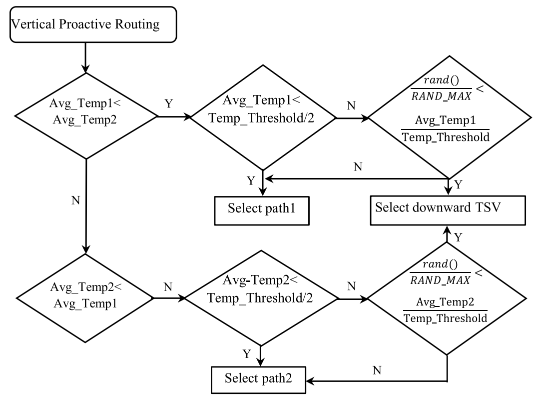


Fig. 6. Operational flowchart of vertical proactive routing.

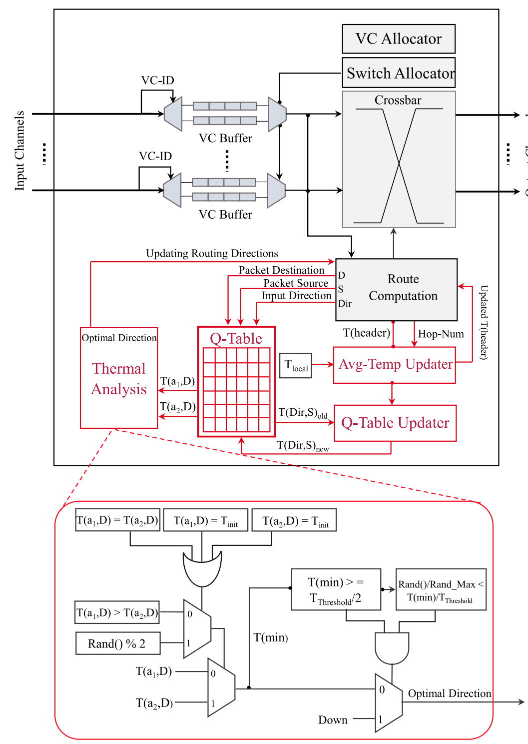
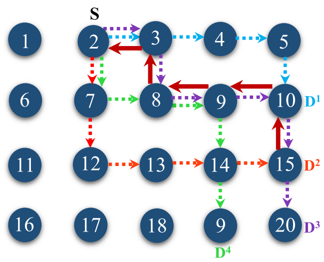


Fig. 7. Architecture of the router in the abstract level.

in the chip. Fig. 8 illustrates our idea in a 2-D NoC including a partial part of Q-tables. Illustrating the idea in a 3-D NoC may complex the figure and we omit it to represent. According to Fig. 8, some network packets are sent from source S to multiple destinations. These packets update the Q-table of their passing routers for goal S. Suppose that, router 15



Node	Optimal Out	Optimal Temp	Goal
3	West	Temp(2)	2

Node	Optimal Out	Optimal Temp	Goal
8	North	Avg_Tmp (2,3)	2

Node	Optimal Out	Optimal Temp	Goal
9	West	Avg_Tmp (2,3,8) OR Avg_Tmp (2,7,8)	2

Node	Optimal Out	Optimal Temp	Goal
10	West	Avg_Tmp (2,3,8,9)	2

Node	Optimal Out	Optimal Temp	Goal
15	North	Avg_Tmp (2,3,8,9,10)	2

Fig. 8. Main idea of the Q-Thermal technique in a 2-D NoC within a partial part of the Q-tables.

intends to send a packet to destination 2, that is S. The packet will employ the Q-table's information gathered from previous packets to take the most optimal decision in its transition.

V. Experimental Results

This section evaluates the performance of our proposed method and compares it with previous state-of-the-art methods, PTB^{3}R [13] and TAAR [5] as a proactive and reactive technique, respectively. Simulations are carried out through the traffic-thermal co-simulation platform, which uses PAT-Noxim [22] to simulate a 3-D NoC platform within routers' thermal sensors, respectively.

For all experiments, the 3-D NoC is an $8 \times 8 \times 4$ mesh, which consists of 256 tiles. For each router, the channel depth of each buffer is 8 flits, and to avoid deadlock, we use two virtual channels in each router's port. The size of each packet is 8 flits, wormhole flow control is adopted, and random arbitration is used for the switch schedule. The ambient temperature is set to 25 °C. Results are captured after 1 000 000 cycles and a warm-up period of 10 000 clock cycles.

A. Analysis of Thermal Distribution

Fig. 9 depicts the thermal propagation profile of Q-Thermal, TAAR, and PTB^{3}R under uniform traffic pattern. It can be seen that temperature is more evenly distributed between layers in the proposed Q-Thermal than other schemes. TAAR moves the traffic load to downward layers whenever encounters a throttled node; otherwise, the traffic load is steered laterally in the same layer. PTB^{3}R has defined the MTTT as a proactive index for selecting a cooler path. Since each router has access to MTTT of its neighbors, the index can merely be helpful for one packet hop. Additionally, PTB^{3}R does not take advantage of downward routing as cooler paths and routings mainly is performed in the source layers which makes the situation worse particularly for top layers. But, Q-Thermal constantly keeps track of the routers' temperature which helps to eliminate the traffic load from potentially overheated regions and hotspots formation. Moreover, Q-Thermal considers local and global thermal information of the network in the routing decisions. As a result, whenever, Q-Thermal found out that a region becomes overheated soon, it moves the traffic load to the downward layer. Our analysis shows that Q-Thermal improves the standard deviation of thermal distribution nearby 28% and 13% in comparison with TAAR and PTB^{3}R, respectively.

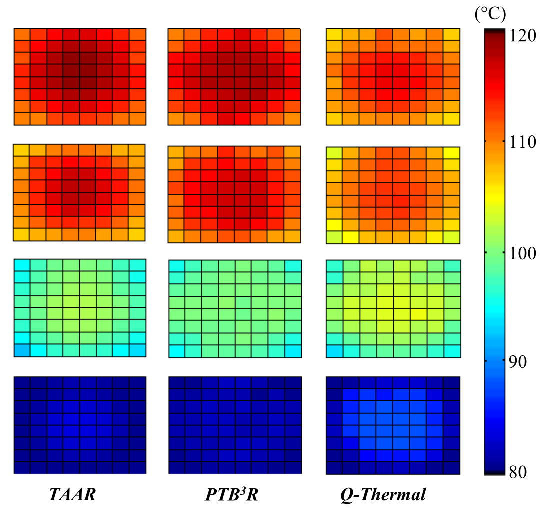


Fig. 9. Thermal distribution of Q-Thermal vs. TAAR and PTB^{3}R.

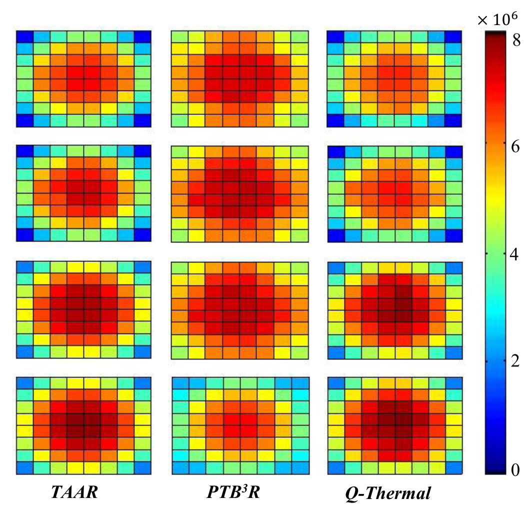


Fig. 10. STLD of Q-Thermal versus TAAR and PTB^{3}R.

B. Analysis of Traffic Load Distribution

The statistical traffic load distribution (STLD) has been utilized to analyze the network loading. Fig. 10 shows the STLD comparison of three techniques over different layers. PTB^{3}R has a more balanced inter and intra-layer traffic distribution than other two techniques due to exploiting beltway paths as potentially cooler paths and avoiding downward routing. Two other schemes steer more traffic load toward the downward layers closer to the heat sink. As the figure illustrates, Q-Thermal has experienced slightly denser traffic in lower layers than TAAR because in TAAR, downward paths are alternatives for the time of encountering hotspots; while Q-Thermal gradually steers traffic to the layers closer to the heat sink in the early steps of routers overheating. Noteworthy, employing an additional virtual channel in Q-Thermal would satisfy the required routing density in the lower layers.

C. Analysis of Network Performance

The network performance (in terms of latency) is carried out for Q-Thermal, TAAR, and PTB^{3}R via measuring the

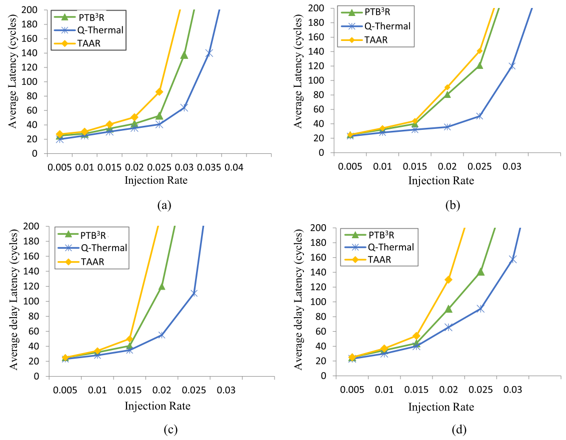


Fig. 11. Average latency of Q-Thermal versus TAAR and PTB^{3}R. (a) Uniform traffic pattern. (b) Transpose traffic pattern. (c) Hotspot traffic pattern. (d) Bit-Reversal traffic pattern.

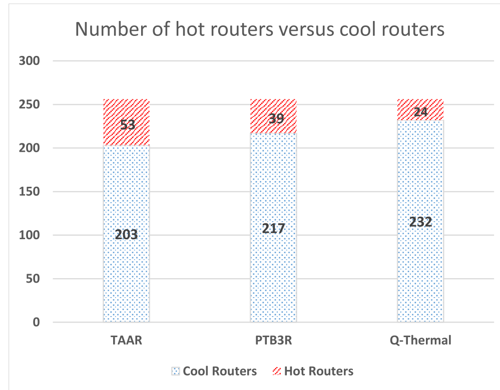


Fig. 12. Number of hotspots for Q-Thermal, TAAR and PTB^{3}R.

TABLE II
COMPARISON OF AREA AND TOTAL POWER CONSUMPTION

Routing Algorithm	Area(μm)	Total power (mw)
$TAAR$	152.861	129
PTB^3R	147.857	126
Q-Thermal	164.175	132

average latency of the network packets under different packet injection rates for four different traffic patterns. The simulated traffic patterns are uniform, transpose, bit-reversal, and hotspot traffic. In the uniform traffic pattern, the source and destination of the packets are selected randomly. In the transpose traffic pattern, a node (i, j) only sends packets to a node $(N-1-j, N-1-i)$, where N is the size of the mesh. In the bit-reversal traffic pattern, the bit pattern of the source and destination addresses is the reversal and, finally, in the hotspot traffic pattern, some nodes are chosen as hotspots receiving an extra portion of the traffic. In this experiment, for the hotspot traffic pattern, we select two nodes (at the centers of layers 1 and 3) as hotspots, with hotspot percentage of $H = 10\%$ (10% traffic more than other nodes).

Fig. 11 depicts the average latency for different packet injection rate under four traffic patterns. Q-Thermal yields less network latency regardless of the type of traffic pattern. TAAR imposes additional latency as it has periodic reconfiguration stages for topology table updating and routability check for the upcoming packets. However, in Q-Thermal, q-tables are updated parallel with network ordinary activities. Besides, to avoid deadlock, TAAR utilizes a two-step routing, one between the source and a cascaded node, and next between the cascaded node and the destination. Meanwhile, TAAR uses the "store and forward" policy in which some packets may need to be stored in the cascaded nodes' buffer for a while until the nodes' switches become free.

All three algorithms use nonminimal paths to avoid sending packets to high thermal paths (or paths with throttled nodes). In this case, Q-Thermal and TAAR send packets to the lower layers causing the route between source and destination to get farther. In this case, PTB^{3}R sends packets via the beltway path. In the worst case, when both source and destination are in the top layer of our experimental platform, a maximum of six hops are walked in both Q-Thermal and TAAR. However, the maximum and the average number of walked hops in PTB^{3}R are more than both Q-Thermal and TAAR.

In addition, employing two VCs improves the routability and adaptivity of Q-Thermal, which in turn decreases the network latency and elevates throughput. The results show that Q-Thermal has experienced an average of 32% improvement over PTB^{3}R and TAAR.

D. Number of Thermal Hot Spots

In this experiment, we measure the number of hotspots (hot routers) for Q-Thermal, TAAR, and PTB^{3}R over 256 routers of the experimental platform. A router with the temperature above 85 °C (the threshold temperature) considered as a hotspot. Fig. 12 illustrates the number of hot routers versus cool routers for all techniques. As can be seen, Q-Thermal reduces the number of hotspots on average by 38% and 54% in comparison with TAAR and PTB^{3}R, respectively, which in turn improves the network availability by 6% and 11% compared to TAAR and PTB^{3}R, respectively.

E. Overhead Costs

To assess the area overhead and power consumption, ORION 2.0, a power-area simulator for interconnection networks, has been utilized. For synthesis, the UMC 90-nm technology, 1-GHz operating frequency, and the supply voltage of 1 V are used. Table II shows that Q-Thermal increases the layout area by 7% and 11% in comparison with TAAR and PTB^{3}R, respectively. This extra cost has arisen mainly due to the Q-table area of routers and the additional virtual channel. Table II also shows that the total power consumption (the sum of both dynamic and static power) of Q-Thermal is 2% and 4% more than TAAR and PTB^{3}R platforms, respectively.

All three algorithms have more and less the same power consumption. TAAR uses topology table, periodic reconfigurable, 'store and forward' policy, and routability check for both source and destination. In PTB^{3}R, MTTT is calculated and sent to all neighbors routers; a suitable route (minimal or nonminimal) is selected for all packets and many packets take a couple of extra hops to get their destinations.

In Q-Thermal, q-tables are used. These issues indicate that power consumption in all three techniques is rather comparable. Moreover, thanks to eliminating overheated regions, the static buffer power of additional VC has been alleviated finely by the reduction of the network total delay and drop the leakage power. Nevertheless, considering the balanced temperature distribution and the performance gained by Q-Thermal, we think that the further costs of our algorithm are acceptable.

VI. Conclusion

This article proposes a thermal-aware Q-learning-based routing algorithm (Q-Thermal) for 3-D NoCs. The main goal of Q-Thermal is to proactively balance thermal distribution over different layers using the thermal information stored in the router's Q-tables. The packets from a source to a destination record temperature of different routers in their path. The recorded temperatures guide routers to steer the packets, traversing the opposite side, to the cooler regions. Our method achieves better balanced thermal distribution, a better standard deviation of the thermal distribution, better network performance, and finally creating fewer hotspots in a chip in comparison with previous works. However, Q-Thermal increases the area slightly with respect to previous works. Balancing the pros and cons of our method indicates that our technique can be considered as a new routing technique to cool down 3-D NoCs.

References

[1] Y. Jin, E. J. Kim, and T. M. Pinkston, "Communication-aware globally-coordinated on-chip networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 2, pp. 242–254, Feb. 2012.

[2] B. Black *et al.*, "Die stacking (3D) microarchitecture," in *Proc. 39th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2006, pp. 469–479.

[3] B. S. Feero and P. P. Pande, "Networks-on-chip in a three-dimensional environment: A performance evaluation," *IEEE Trans. Comput.*, vol. 58, no. 1, pp. 32–45, Jan. 2009.

[4] K. Kang, J. Kim, S. Yoo, and C.-M. Kyung, "Runtime power management of 3-D multi-core architectures under peak power and temperature constraints," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 6, pp. 905–918, Jun. 2011.

[5] K.-C. Chen, S.-Y. Lin, H.-S. Hung, and A.-Y.-A. Wu, "Topology-aware adaptive routing for nonstationary irregular mesh in throttled 3D NoC systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 10, pp. 2109–2120, Oct. 2013.

[6] I. Yeo, C. C. Liu, and E. J. Kim, "Predictive dynamic thermal management for multicore systems," in *Proc. 45th Annu. Conf. Design Autom. (DAC)*, Jun. 2008, pp. 734–739.

[7] 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," in *Proc. 4th ACM/IEEE Int. Symp. Netw. Chip*, May 2010, pp. 223–230.

[8] S. Borkar, "Thousand core chips: A technology perspective," in *Proc. 44th Annu. Design Automat. Conf.*, Jun. 2007, pp. 746–749.

[9] K.-C.-J. Chen, C.-H. Chao, and A.-Y.-A. Wu, "Thermal-aware 3D network-on-chip (3D NoC) designs: Routing algorithms and thermal managements," *IEEE Circuits Syst. Mag.*, vol. 15, no. 4, pp. 45–69, 4th Quart., 2015.

[10] P. Sivakumar, K. Pandiaraj, and K. JeyaPrakash, "Optimization of thermal aware multilevel routing for 3D IC," *Anal. Integr. Circuits Signal Process.*, vol. 103, no. 1, pp. 131–142, Apr. 2020.

[11] A. Bartolini, M. Cacciari, A. Tilli, and L. Benini, "Thermal and energy management of high-performance multicores: Distributed and self-calibrating model-predictive controller," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 170–183, Jan. 2013.

[12] F. Farahnakian, M. Ebrahimi, M. Daneshtalab, P. Liljeberg, and J. Plosila, "Q-learning based congestion-aware routing algorithm for on-chip network," in *Proc. IEEE 2nd Int. Conf. Netw. Embedded Syst. Enterprise Appl.*, Dec. 2011, pp. 1–7.

[13] C.-C. Kuo, K.-C. Chen, E.-J. Chang, and A.-Y. Wu, "Proactive thermal-budget-based beltway routing algorithm for thermal-aware 3D NoC systems," in *Proc. Int. Symp. Syst. Chip (SoC)*, Oct. 2013, pp. 1–4.

[14] K. C. Chen, C.-C. Kuo, H. S. Hung, and A. Y. A. Wu, "Traffic-and thermal-aware adaptive beltway routing for three dimensional network-on-chip systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 1660–1663.

[15] S. C. Lee and T. H. Han, "Q-function-based traffic- and thermal-aware adaptive routing for 3D network-on-chip," *Electronics*, vol. 9, no. 3, p. 392, Feb. 2020.

[16] E. Taheri, K. Mohammadi, and A. Patooghy, "ON–OFF: A reactive routing algorithm for dynamic thermal management in 3d nocs," *IET Comput. Digit. Techn.*, vol. 13, no. 1, pp. 11–19, 2019.

[17] J. Wang, H. Gu, K. Wang, Y. Yang, and K. Wang, "DRTL: A heat-balanced deadlock-free routing algorithm for 3D topology network-on-chip," *Microprocessors Microsyst.*, vol. 45, pp. 95–104, Aug. 2016.

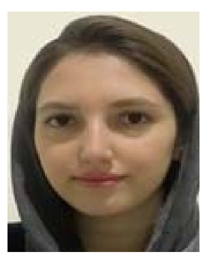
[18] R. Dash, A. Majumdar, V. Pangracious, A. K. Turuk, and J. L. Risco-Martin, "ATAR: An adaptive thermal-aware routing algorithm for 3-D network-on-chip systems," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 8, no. 12, pp. 2122–2129, Dec. 2018.

[19] V. Pagracious, R. Dash, and A. K. Turuk, "3D-cool: Design and development of adaptive thermal-aware three-dimensional NoC-based multiprocessor chip," in *Proc. IEEE Symp. Low-Power High-Speed Chips (COOL CHIPS)*, Apr. 2018, pp. 1–3.

[20] K. Cao, J. Zhou, T. Wei, M. Chen, S. Hu, and K. Li, "A survey of optimization techniques for thermal-aware 3D processors," *J. Syst. Archit.*, vol. 97, pp. 397–415, Aug. 2019.

[21] A. K. Coskun, J. L. Ayala, D. Atienza, T. S. Rosing, and Y. Leblebici, "Dynamic thermal management in 3D multicore architectures," in *Proc. Design, Autom. Test Eur. Conf. Exhib.*, Apr. 2009, pp. 1410–1415.

[22] A. Norollah, D. Derafshi, H. Beitollahi, and A. Patooghy, "PAT-Noxim: A precise power & thermal cycle-accurate NoC simulator," in *Proc. 31st IEEE Int. Syst. Chip Conf. (SOCC)*, Arlington, VA, USA, Sep. 2018, pp. 163–168.



Narges Shahabinejad received the B.S. degree in computer engineering from Shahid Bahonar University, Kerman, Iran, in 2004, and the M.Sc. degree in computer engineering from the Iran University of Science and Technology, Tehran, Iran, in 2019.

Her research interests include computer architecture, low-power design, and hardware accelerators.

Hakem Beitollahi received the B.S. degree in computer engineering from the University of Tehran, Tehran, Iran, in 2002, the M.S. degree from the Sharif University of Technology, Tehran, in 2005, and the Ph.D. degree from the Katholieke Universiteit Leuven, Leuven, Belgium, in 2012.

He is currently an Assistant Professor with the Department of Computer Engineering, Iran University of Science and Technology, Tehran. His research interests include real-time systems, fault tolerance and dependability, reconfigurable computing, logic design, and hardware accelerators.