

Mesh Connected Crossbars: A Novel NoC Topology with Scalable Communication Bandwidth

Arash Tavakkol
Sharif University of Technology
& IPM
Tehran, Iran
arasht@ipm.ir

Reza Moraveji
Shahid-Beheshti University
& IPM
Tehran, Iran
moraveji_r@ipm.ir

Hamid Sarbazi-Azad
Sharif University of Technology
& IPM
Tehran, Iran
azad@ipm.ir

Abstract

Recent studies have revealed that on-chip interconnects neither is wire plentiful nor is bandwidth cheap. Based on the results of these studies, in physical design of Multi-processor System-on-Chip (MPSoCs), both the wiring density constraint and routing of wires are controversial issues, and there is a trade-off between the network bandwidth and wiring limitations. Therefore, in this paper, we introduce a new topology, named Mesh Connected Crossbars (MCC), to enhance the communication bandwidth between processing elements; the proposed topology, also, has significant topological advantages over traditional torus- and mesh-based NoCs. Furthermore, we study the topological properties of MCCs and propose deterministic and fully adaptive deadlock-free routing algorithms in an attempt to evaluate the performance of MCC in different working conditions. The simulation results show that under constant wiring conditions, MCC exhibits higher performance and consumes lower energy in comparison with equivalent Torus or Mesh networks.

1. Introduction

Technology scaling intensified the number of processing elements and memory cores on a single chip and increased their operation speed [15]. Hence, the communication between cores has become a major bottleneck for the performance of such systems [15, 5, 7, 17]. Traditionally, communication between processing elements in Systems-on-

Chip (SoC) was based on buses. However, for large Multi-processor System-on-Chips (MPSoCs) with many processing elements, bus is a bottleneck from performance, scalability, and power dissipation points of view and is not able to support the heavy communication traffic of current applications [12].

The Network-on-Chip (NoC) is a communication-centric interconnection approach which provides a scalable infrastructure to interconnect different IPs and sub-systems in a SoC [5, 7, 17]. Moreover, NoCs can make SoCs more structured, and reusable, and can also improve their performance [5, 8]. However, solutions to overcome performance limitations in NoCs are yet to be presented.

Many topologies with different capabilities have been proposed for NoCs including Mesh [8], Torus [7], Octagon [9], SPIN [4], and BFT [11]. In such cases, one of the main goals is to improve network performance by providing better static topological characteristics such as diameter and average inter-node distance [8]. However, when designing communication architecture, it is vital to consider the effect of physical design constraints such as wire routing, wiring density, and power consumption. The authors of [13] showed that in contrast to normal beliefs, on chip interconnections suffer from certain physical limitations which lead to great performance reduction. According to their results, when we consider these physical design constraints, higher dimensional networks may have serious limitations. Moreover, these constraints cause designers to decline the number of communication channels or wire bandwidth.

In this paper, we consider the concepts of performance enhancement and wiring problems by proposing a new NoC

topology called Mesh Connected Crossbars (*MCC*). *MCC* provides a communication architecture for IP cores with reduced number of outgoing channels and better topological properties with respect to Mesh and Tori networks. This reduction causes simplification of wire routing on the chip and declines the wiring density, which is a quantitative measure of wiring complexity [13]. In addition, under constant wiring density conditions, *MCC* lets us to improve communication performance by increasing the communication channel bandwidth. In *MCC* topology, all nodes are virtually connected based on Diagonal Connected Mesh (*DCM*) graph; that is, all links of each fully connected sub-graph in *DCM* will be replaced with a single crossbar switch in *MCC*. This replacement reduces the number of channels of a single node, and hence, the node degree will be reduced. We will formally define and study the topological properties of *DCM* in section 2. In section 3, a deadlock free adaptive routing algorithm is introduced for *DCM*-based networks and a intuitive proof for deadlock freedom of the algorithm is given. In section 4, we address the design of *MCC* based on *DCM*. Finally, we evaluate the performance of *MCC* in section 5, and conclude the paper in section 6.

2. The *DCM* Topology

Mesh networks have been widely used as the underlying interconnection network structure in parallel computers [14, 16] due to their simplicity, regularity, ease of implementation, and scalability [8]. Two and three-dimensional Mesh and Torus were widely used in commercial massively parallel systems, such as IBM BlueGene/L [6], Cray XT4 [2], and Cray XT3 [1].

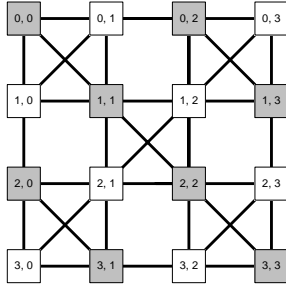


Figure 1. A sample *DCM* network of size 4×4 denoted $DCM_{4 \times 4}$.

Diagonal Connected Mesh (*DCM*) is a topology based on traditional Mesh, which improves its topological characteristics by providing diagonal links between nodes in addition to the usual links in the Mesh. A 4×4 *DCM* is shown in Figure 1. Note that our main goal in proposing *DCM* is to provide a network structure for the Mesh-

Connected Crossbars (*MCC*) network that contains fully connected sub-graphs; hence, we can substitute the interconnects in a sub-graph with a crossbar switch. As will be discussed later in section 4, substituting multiple output links of a single node by only one crossbar switch port helps improve communication bandwidth and alleviates physical wiring limitations.

Like common instances of 2D and 3D Meshes and Tori (to match physical constraints), we focus on 2D and 3D versions of *DCM*. However, it is easy to extend the definition of the *DCM* to any number of dimensions. Moreover, as we will see later, higher dimensional *MCC*s require large crossbar switches (with more than 8 ports) which may reduce the scalability of the proposed topology.

2.1. Notations and Assumptions for *DCM*

We base our definitions on the ordinary Mesh specifications and notation. Hence, in the first step, we will briefly introduce necessary concepts in the Mesh.

Definition 1 . An n -dimensional Mesh, denoted by $M_{k_0 \times k_1 \times \dots \times k_{n-1}}$, consists of a set of nodes $N(M_{k_0 \times k_1 \times \dots \times k_{n-1}}) = \{(a_0, a_1, \dots, a_{n-1}) \mid \forall i : 0 \leq i \leq n-1 \implies 0 \leq a_i \leq k_i - 1\}$ where two nodes $A = (a_0, a_1, \dots, a_{n-1})$ and $B = (b_0, b_1, \dots, b_{n-1})$ are connected by a link if and only if:

$$\text{There exist a unique } j \text{ such that : } (a_j = b_j \pm 1) \quad (1a)$$

$$\forall i : 0 \leq i \leq n-1 \text{ and } i \neq j \implies a_i = b_i \quad (1b)$$

Definition 2 . In a $M_{k_0 \times k_1 \times \dots \times k_{n-1}}$ two nodes $A = (a_0, a_1, \dots, a_{n-1})$ and $B = (b_0, b_1, \dots, b_{n-1})$ are called normal neighbors if their address satisfy Eq. 1. E.g., nodes $(1, 1)$ and $(0, 1)$ in Figure 1 are normal neighbors.

Definition 3 . In a $M_{k_0 \times k_1 \times \dots \times k_{n-1}}$ two nodes $A = (a_0, a_1, \dots, a_{n-1})$ and $B = (b_0, b_1, \dots, b_{n-1})$ are called diagonal neighbors if their address satisfy the following equation:

There exist j and k such that :

$$j \neq k \text{ and } (a_j = b_j \pm 1) \text{ and } (a_k = b_k \pm 1) \quad (2a)$$

$$\forall i : 0 \leq i \leq n-1 \text{ and } i \neq j \text{ and } i \neq k \implies a_i = b_i \text{ or } a_i = b_i \pm 1 \quad (2b)$$

For example nodes a and c in Figure 2(a) and nodes $(0, 0)$ and $(1, 1)$ in Figure 1 are diagonal neighbors.

Definition 4 . A two dimensional *DCM*, denoted by $DCM_{k_0 \times k_1}$, consists of a set of nodes $N(DCM_{k_0 \times k_1}) = \{(a_0, a_1) \mid 0 \leq a_0 \leq k_0 - 1, 0 \leq a_1 \leq k_1 - 1\}$ where two nodes $A = (a_0, a_1)$ and $B = (b_0, b_1)$ are connected

by a horizontal or a vertical link if they are normal neighbors and the conditions in Eq. 1 are satisfied. Additionally, if they are diagonal neighbors, they can be connected by a diagonal link when one of the following conditions are satisfied:

$$\text{if } a_0 \& a_1 \text{ are both even or odd simultaneously} \\ \left\{ \begin{array}{l} a_0 = b_0 + 1 \\ a_1 = b_1 + 1 \end{array} \right\} \text{ or } \left\{ \begin{array}{l} a_0 = b_0 - 1 \\ a_1 = b_1 - 1 \end{array} \right\} \quad (3a)$$

$$\text{if on of } a_0 \& a_1 \text{ is even and the other is odd} \\ \left\{ \begin{array}{l} a_0 = b_0 + 1 \\ a_1 = b_1 - 1 \end{array} \right\} \text{ or } \left\{ \begin{array}{l} a_0 = b_0 - 1 \\ a_1 = b_1 + 1 \end{array} \right\} \quad (3b)$$

Based on these equations we define two types of nodes for the 2D DCM. **Type I** nodes are those whose diagonal connections satisfy Eq. 2a and **Type II** nodes are those whose diagonal connections satisfy Eq. 2b. Figure 1 depicts a $DCM_{4 \times 4}$ network. The nodes highlighted by gray color are of **Type I** and others are of **Type II**. As can be seen in this figure, Eq. 3a and Eq. 3b lead to a Mesh-based network with additional diagonal links compared to the traditional Mesh. However, the mentioned equations do not allow connection to all diagonal neighbors. This restriction is directly related to our goal in defining DCM with the aim of cost reduction. In other words, we will substitute fully connected sub-graphs by a crossbar switch which leads to a reduction in network cost and power consumption, as will be discussed in section 4.

Definition 5 . A three dimensional DCM, denoted by $DCM_{k_0 \times k_1 \times k_2}$, consists of a set of nodes $N(DCM_{k_0 \times k_1 \times k_2}) = \{(a_0, a_1, a_2) \mid 0 \leq a_0 \leq k_0 - 1, 0 \leq a_1 \leq k_1 - 1, 0 \leq a_2 \leq k_2 - 1\}$ where two nodes $A = (a_0, a_1, a_2)$ and $B = (b_0, b_1, b_2)$ are connected by a horizontal or a vertical link if they are normal neighbors and the conditions of Eq. 1 are satisfied. Additionally, if they are diagonal neighbors, they can be connected by a diagonal link inside one the planes $\{d_0 \times d_1, d_0 \times d_2, d_1 \times d_2\}$, where d_i represents the i -th dimension axis, when one the conditions of Eq. 4 are satisfied. Connections of diagonal neighbors in the $d_h \times d_m$ plane are as follows:

$$\text{if both of } a_h \text{ and } a_m \text{ are even} \\ \text{or odd simultaneously} \\ \left\{ \begin{array}{l} a_h = b_h + 1 \\ a_m = b_m + 1 \\ a_l = b_l, l \neq h \text{ and } l \neq m \end{array} \right\} \text{ or } \\ \left\{ \begin{array}{l} a_h = b_h - 1 \\ a_m = b_m - 1 \\ a_l = b_l, l \neq h \text{ and } l \neq m \end{array} \right\} \quad (4a)$$

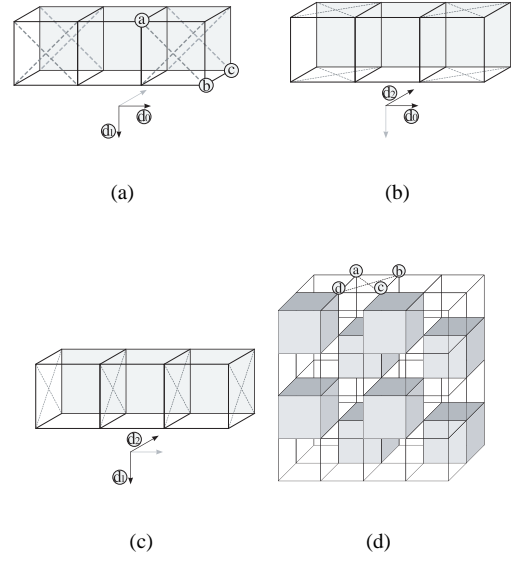


Figure 2. Diagonal links defined for $DCM_{k_0 \times k_1 \times k_2}$. (a) $d_0 \times d_1$ plane (b) $d_0 \times d_1$ plane (c) $d_0 \times d_1$ plane (d) Highlighted cubes depict fully connected sub-graphs with 8 nodes.

$$\text{if on of } a_h \text{ and } a_m \text{ is even} \\ \text{and the other one is odd} \\ \left\{ \begin{array}{l} a_h = b_h + 1 \\ a_m = b_m - 1 \\ a_l = b_l, l \neq h \text{ and } l \neq m \end{array} \right\} \text{ or } \\ \left\{ \begin{array}{l} a_h = b_h - 1 \\ a_m = b_m + 1 \\ a_l = b_l, l \neq h \text{ and } l \neq m \end{array} \right\} \quad (4b)$$

Figure 2 shows diagonal connections of this type in different planes of a $DCM_{4 \times 2 \times 2}$. E.g., nodes a and b in Figure 2(a) are both in the $d_0 \times d_1$ plane.

Further to this type of diagonal links we can define new diagonal links which connect diagonal neighbors in two different planes (such as nodes a and c in Figure 2(a)).

Assume A and B are diagonal neighbors in two different planes; there exists a diagonal link between them if and only if all of the following paths can be established using the links defined in Eqs. 1 and 4:

$$\begin{aligned} A = a_2 a_1 a_0 &\rightarrow a_2 (a_1 \pm 1) (a_0 \pm 1) \\ &\rightarrow (a_2 \pm 1) (a_1 \pm 1) (a_0 \pm 1) = b_2 b_1 b_0 = B \\ A = a_2 a_1 a_0 &\rightarrow (a_2 \pm 1) (a_1 \pm 1) a_0 \\ &\rightarrow (a_2 \pm 1) (a_1 \pm 1) (a_0 \pm 1) = b_2 b_1 b_0 = B \\ A = a_2 a_1 a_0 &\rightarrow (a_2 \pm 1) a_1 (a_0 \pm 1) \\ &\rightarrow (a_2 \pm 1) (a_1 \pm 1) (a_0 \pm 1) = b_2 b_1 b_0 = B \end{aligned}$$

Similar to the two-dimensional case, the aforementioned definitions for $DCM_{k_0 \times k_1 \times k_2}$ links, restrict the connection of possible diagonal links due to the concerns regarding a cost effective design. Hence, only a portion of all possible fully-connected sub-graphs will be produced. Figure 2(d) shows the fully connected sub-graphs consisting 8 nodes in a $DCM_{5 \times 5 \times 3}$.

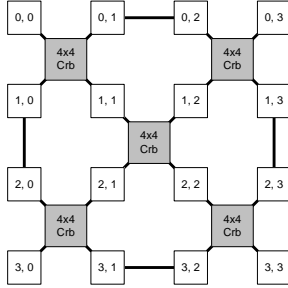


Figure 3. A sample MCC network of size 4×4 , denoted $MCC_{4 \times 4}$, which is constructed by substituting links of each fully connected sub-graph in DCM with a single crossbar switch.

2.2. Topological Properties

In this section we will present topological properties of DCM and MCC which are commonly used to measure and compare the static network performance of a system.

Node Degree: The node degree (N_D) is defined as the number of physical channels emanating from a node [8]. This attribute shows the node's I/O complexity. The N_D of a node in $DCM_{k_0 \times k_1}$, depends on the node's position in the network structure and can get these values: $N_D(DCM_{k_0 \times k_1}) \in \{3, 4, 6\}$. For example, in Figure 3, we have $N_D(0, 0) = 3$, $N_D(0, 1) = 4$, and $N_D(1, 1) = 6$.

For two-dimensional Mesh we have $N_D(Mesh_{k_0 \times k_1}) \in \{2, 3, 4\}$, and for two-dimensional Torus network we have $N_D(Torus_{k_0 \times k_1}) = 4$ [8]. Therefore, the I/O complexity of a sample node in DCM is more than that in Mesh and Torus. However, we will replace the fully-connected sub-graphs with crossbar switches to implement the MCC, which leads to better node degree in comparison with Mesh and Torus (Figure 3). In this case, there will be a great reduction in the number of emanated physical channels from network nodes and $N_D(MCC_{k_0 \times k_1}) \in \{2, 3\}$. As will be discussed further in section 4, this reduction has a positive impact on simplifying the I/O complexity of the network nodes and allows us to enhance the communication bandwidth.

The mentioned points are also correct for three-dimensional DCM and MCC. This means that

the $MCC_{k_0 \times k_1 \times k_2}$ with crossbar switches has $N_D(MCC_{k_0 \times k_1 \times k_2}) \in \{1, 2\}$, while three-dimensional Mesh and Torus have $N_D(Mesh_{k_0 \times k_1 \times k_2}) \in \{3, 4, 5, 6\}$ and $N_D(Torus_{k_0 \times k_1 \times k_2}) = 6$.

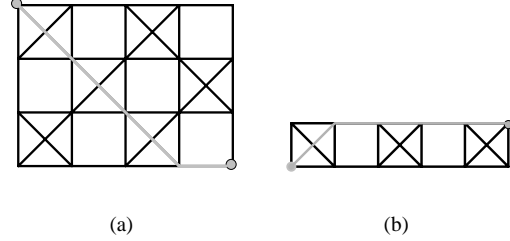


Figure 4. (a) A $DCM_{4 \times 4}$, with diameter = 4. The farthest nodes and the shortest path between them is depicted. (b) A sample case when traversing the path between two farthest nodes. All but one traversed links are non-diagonal.

Diameter: the diameter (D) of a network is the maximum internodes distance, i.e. the maximum number of links that should be traversed to send a message to all nodes along the shortest path. The smaller the diameter of a network, the less time it takes to send a message from one node to the farthest node. For a two-dimensional Mesh network we have $D(Mesh_{k_0 \times k_1}) = k_0 + k_1 - 2$ which corresponds to sending a message from one corner of the network, say $(0, 0)$, to another corner, say $(k_0 - 1, k_1 - 1)$. In the same way, this parameter becomes $D(Torus_{k_0 \times k_1}) = \lfloor k_0/2 \rfloor \lceil k_1/2 \rceil$ for the Torus network.

In a $DCM_{k_0 \times k_1}$, which has some additional diagonal links, compared to ordinary Mesh, the maximum distance between two nodes can be declined by traversing a diagonal link instead of traversing two separate links in two different dimensions. Figure 4(a) shows a sample traversed path in DCM to move from one corner, $(k_0 - 1, k_1 - 1) = (3, 4)$, to another, $(0, 0)$. As depicted in this figure, we can reach a diagonal link just by moving one step left and resuming our path on diagonal links. The diameter of this topology is equal to: $D(DCM_{k_0 \times k_1}) = \max(k_0, k_1) - 1$. It is clear that this relation is correct and for now we rely on this short explanation. Also, the same method can be used to achieve the results for three-dimensional cases as $D(DCM_{k_0 \times k_1 \times k_2}) = \max(k_0, k_1, k_2) - 1$. These values show that the maximum distance is declined in DCM in comparison to Mesh. Moreover, the ratio of this reduction increases as the network size increases. Hypothetically, this diameter reduction leads to better performance in the network.

In addition, the diameter of DCM and Torus challenge

one another and depend on diverse factors such as the number of dimensions and their sizes. For example, in two-dimensional topologies with the same number of nodes in each dimension, the diameter of these two topologies is almost the same.

The mentioned points are also correct for *MCC*. This is because changing normal links with a single crossbar switch does not affect the possibility of moving from a node to its diagonal neighbor. However, in realistic situations there may be a contention between multiple input ports of a crossbar to access a specific output port. This effect is closely related to routing function and network traffic load, which are not considered when theoretically analyzing the diameter.

3. DiaR: A Deadlock Free Routing Algorithm for Diagonal Mesh Connected Topology

In this section, we will propose a deadlock free routing algorithm for *DCM* topology which can be used in *MCC* as well. We call this routing algorithm **DiaR**. The basic idea for DiaR is the same as Duato's routing algorithm [8]. This means that DiaR has two classes for routing, one of which is FAR (Fully Adaptive Routing), its first routing class, and the other is deadlock avoidance class (its second class). The FAR algorithm can route packets through minimal path to anywhere in the network, without any constraint. Hence, we will only propose a deterministic, minimal path, deadlock and livelock free routing algorithm for the second class. Our proposed routing algorithm is for two and three dimensional *DCM* networks. For brevity, we will just present the routing algorithm for two-dimensional *DCM* and call it 2D-Det. We can easily extend the idea of 2D-Det to define the 3D-Det, etc. Figure 5 shows the pseudo code for 2D-Det. The mechanism of 2D-Det is similar to dimension ordered algorithms in Mesh and Torus networks [8], in which, first the distance vector of the current and destination node address is calculated and then it is tried to reduce the offset of dimensions to zero, according to a pre-defined precedence. As an exemplification, the traditional deterministic routing for Mesh (XY-Routing) first tries to reduce the offset of dimension X to zero and then reduces the offset of dimension Y to zero [8].

However, there is a major difference between 2D-Det and the mentioned routings. In 2D-Det, it is sometimes possible to have a simultaneous reduction in both X and Y offsets. As mentioned previously, **Type I** nodes have additional X^-Y^- and X^+Y^+ diagonal channels in comparison with traditional two-dimensional Mesh nodes. Hence, 2D-Det can route the messages through these channels when both $X_{difference}$ and $Y_{difference}$ are positive or both negative. On the other hand, in **Type II** nodes, 2D-Det can route the messages through X^-Y^+ and X^+Y^- channels if either $X_{difference}$ or $Y_{difference}$ is positive and the other is

negative, and vice versa. Despite this concurrent reduction, there is still an ordering in offset reduction. This means that we can only reduce the offset of Y when the X dimension offset is zero.

Inputs :

current node address: $(X_{current}, Y_{current})$

destination node address: $(X_{destination}, Y_{destination})$

Output : The selected output physical channel

$X_{difference} := X_{destination} - X_{current};$

$Y_{difference} := Y_{destination} - Y_{current};$

if $(X_{difference} = 0)$ and $(Y_{difference} = 0)$

 return *EjectionChannel*;

if $((X_{current}$ is even) and $(Y_{current}$ is even))

 or $((X_{current}$ is odd) and $(Y_{current}$ is odd))

 { if $(X_{difference} > 0)$

 {if $(Y_{difference} > 0)$ return X^+Y^+ ;
 else return X^+ ; }

 else if $(X_{difference} < 0)$

 {if $(Y_{difference} < 0)$ return X^-Y^- ;
 else return X^- ; }

 else if $(X_{difference} = 0)$

 {if $(Y_{difference} > 0)$ return Y^+ ;
 else return Y^- ; }

 }

if $((X_{current}$ is even) and $(Y_{current}$ is odd))

 or $((X_{current}$ is odd) and $(Y_{current}$ is even))

 { if $(X_{difference} > 0)$

 {if $(Y_{difference} < 0)$ return X^+Y^- ;
 else return X^+ ; }

 else if $(X_{difference} < 0)$

 {if $(Y_{difference} > 0)$ return X^-Y^+ ;
 else return X^- ; }

 else if $(X_{difference} = 0)$

 {if $(Y_{difference} > 0)$ return Y^+ ;
 else return Y^- ; }

 }

Figure 5. Deterministic routing algorithm for two-dimensional DCM.

To give an intuition that 2D-Det is deadlock-free, we can use the well known turn model introduced by Glass and Ni [10]. As mentioned in [10], a routing algorithm can prevent deadlock by prohibiting certain turns in the network. In our case, by investigating the *DCM* structure we can identify 12 types of none zero and none 180° turns [10] which can be used to form 8 basic cycles as shown in Figure 6(a). Since the mechanism of 2D-Det is similar to dimension ordered, it does not allow any turn occurrence when source channel of the turn is of Y direction. This limitation prohibits certain turns. Such turns are highlighted by a gray color in Figure

6(b). As shown in this figure, after prohibiting the mentioned turns, all 8 basic cycles are broken and consequently, there is no possibility for deadlock occurrence.

The formal proof of deadlock freeness of the 2D-Det is provided in [18].

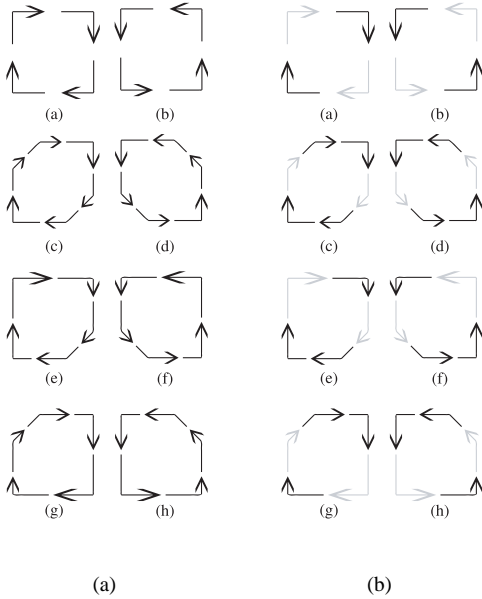


Figure 6. (a) Basic cycles in the network which are formed by all types of possible none zero and none 180° turns. (b) The highlighted turns are prohibited ones by 2D-Det. All basic cycles have been broken.

4. Simplification of Wiring and Bandwidth Enhancement

As mentioned in section 2, the proposed *DCM* topology is used to connect nodes of the *MCC* network. We first define a two dimensional *MCC*.

Definition 6 . A two dimensional *MCC*, denoted by $MCC(k_0, k_1)$ or $MCC_{k_0 \times k_1}$, consists of a set of nodes $N(MCC_{k_0 \times k_1}) = \{(a_0, a_1) \mid 0 \leq a_0 \leq k_0 - 1, 0 \leq a_1 \leq k_1 - 1\}$, in which all nodes are connected using the rules defined in Definition 4 but with a difference in the implementation method, in that all links connecting nodes of each fully-connected sub-graph of size 4 are replaced with a single crossbar switch of size 4×4 .

Figure 3 shows the result of this replacement on the links of the $DCM_{4 \times 4}$ of Figure 1. All of the possible direct

connections which were available between nodes of *DCM* could also be established in *MCC* network. The difference is that some of the mentioned connections should be established through a crossbar switch, instead of a direct link. This substitution leads to a drawback; contention will occur if more than one node tries to connect to a single node through a crossbar. As an example, when nodes (1, 1) and (0, 1) simultaneously try to connect to node (0, 0), one of them should wait till the end of the connection of the other. This effect will degrade network performance, especially in situations where the probability of contention is high i.e. when traffic load of the network is high around the switch. We will discuss this effect when we propose experimental results in section 5.

It is clear that the mentioned substitution reduces node degree in the network with respect to Mesh. To illustrate, in $Mesh_{4 \times 4}$ we have $N_{D, Mesh_{4 \times 4}}(0, 0) = 2$, $N_{D, Mesh_{4 \times 4}}(0, 1) = 3$, and $N_{D, Mesh_{4 \times 4}}(1, 1) = 4$; in $MCC_{4 \times 4}$ these values are reduced to $N_{D, MCC_{4 \times 4}}(0, 0) = 1$, $N_{D, MCC_{4 \times 4}}(0, 1) = 2$, and $N_{D, DCM_{4 \times 4}}(1, 1) = 3$. Consequently, most of the space reserved for wiring links in Mesh is freed, and the wiring complexity will be reduced. In addition, we can use these free spaces to enhance the bandwidth of the remaining links of the network- bandwidth is defined as the number of bits that can be transferred in parallel in one cycle of communication. To illustrate, if the bandwidth of the network is 128 bits, for each bidirectional link there would be 2×128 outgoing wires for each side of the link. Hence in a $DCM_{4 \times 4}$, there are $3 \times 2 \times 128$, $4 \times 2 \times 128$, and $6 \times 2 \times 128$ outgoing wires in nodes (0, 0), (0, 1), and (1, 1) respectively. In addition, in a $Mesh_{4 \times 4}$ there are $2 \times 2 \times 128$, $3 \times 2 \times 128$, and $4 \times 2 \times 128$ outgoing wires on nodes (0, 0), (0, 1), and (1, 1) respectively. But in $MCC_{4 \times 4}$, these values are reduced to 2×128 , $2 \times 2 \times 128$, and $2 \times 2 \times 128$. In a nutshell, the mentioned substitution declines the wiring complexity and releases space which can be used to enhance the bandwidth of the remaining links. This enhancement improves the performance and can also compensate for the degradation effect of the crossbar switch. Our experimental results proposed in section 5, verify the correctness of this claim.

Definition 7 . A three dimensional *MCC*, denoted by $MCC(k_0, k_1, k_2)$ or MCC_{k_0, k_1, k_2} , consists of a set of nodes $N(MCC_{k_0 \times k_1 \times k_2}) = \{(a_0, a_1, a_2) \mid 0 \leq a_0 \leq k_0 - 1, 0 \leq a_1 \leq k_1 - 1, 0 \leq a_2 \leq k_2 - 1\}$, in which all nodes are connected using the rules defined in Definition 5 but with a difference in implementation, in that all links connecting nodes of a fully-connected sub-graph of size 4 or 8 are replaced with a single crossbar switch of size 4×4 or 8×8 respectively.

To substitute normal links of a three dimensional *DCM* with crossbar switches, we need crossbar switches of two

different sizes:

- Crossbar switches of size 8×8 : for connecting fully-connected sub-graphs with their nodes forming a cube. The sub-graphs in Figure 2(d), highlighted with a gray color are as such.
- Crossbar switches of size 4×4 : for connecting fully-connected sub-graphs with their nodes forming a square. Nodes a, b, c, and d in Figure 2(d) are as such.

5. Experimental Results

In this section, we verify the efficiency of *MCC* in comparison with its equivalent traditional topologies such as Mesh and Torus. For this purpose, we select topologies of size $4 \times 4 \times 4$ and set the packet length to 32 flits. The illustrated results are true for other sizes of the network and packet lengths but with minor variations. We will also evaluate the efficiency of our proposed topology under popular traffic patterns: uniform and hot spot with the hot rate of 15%. The simulations are done with Xmutator [16] for performance evaluation and with Orion [19] for power and energy evaluation in the 65nm technology.

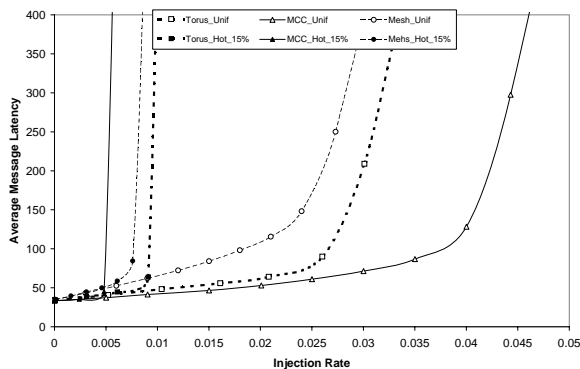


Figure 7. Performance of Mesh, Torus, and MCC under hot spot and uniform traffic patterns

Figure 7 compares latency in Mesh, Torus, and *MCC* topologies, with 5 virtual channels. It is apparent that under uniform traffic pattern, *MCC* has a considerable performance enhancement in comparison with Mesh and Torus, which shows the efficiency of our proposed topology. We mentioned that crossbar switches decline the performance due to long waiting time to get the communication channel. This fact is observable where the performance of *MCC* is worse than Torus and Mesh. Due to higher traffic load around the hot spot node, messages passing the crossbars around that node and its neighbors experience

more blocking time. Because of the nature of crossbar switch this waiting time will increase exponentially. Figure

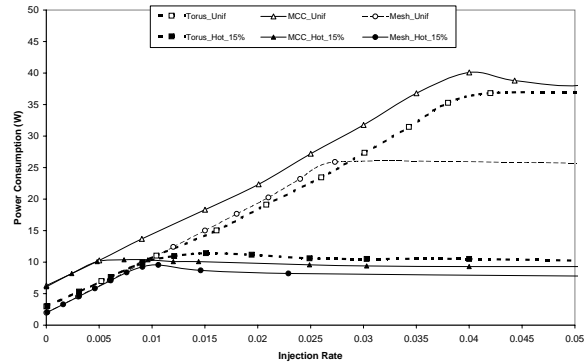


Figure 8. Power consumption of Mesh, Torus, and MCC under hot spot and uniform traffic patterns

8 shows the power consumption of these topologies. The high power consumption of *MCC* reflects the incremental effect of crossbar switches on power consumption. Indeed, better bandwidth and improved performance cost more power consumption. The power consumption of *MCC* under hotspot traffic pattern is considerably lower compared to uniform traffic pattern. Again, we can conclude that the blocking time in *MCC* with hot spot traffic model is so high that leads to power reduction. We have shown that the performance of *MCC* is worse than Torus under hot spot traffic model. But there is a trade off between power consumption and performance or throughput. As we can see from Figure 8, *MCC* consumes less power than Torus, while the performance of Torus outperforms its *MCC* counterpart. It

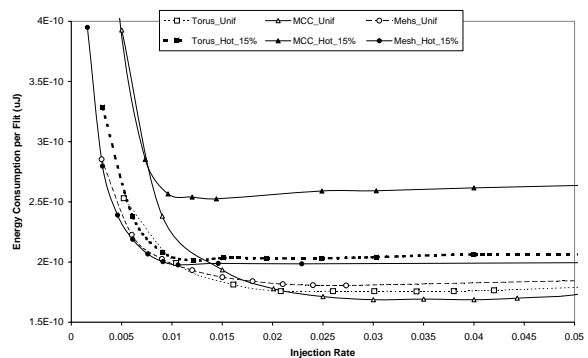


Figure 9. Energy consumption of Mesh, Torus, and MCC under hot spot and uniform traffic patterns

has been shown in [14] that it is unfair to compare different architectures in terms of their power efficiency without

considering their throughputs. It is thus more desirable to examine the energy consumption ratio (which is the same metric as the often-quoted power-delay product) of competing architectures. Figure 9 provide the energy consumption diagrams for each flit communicated in a network. It is clear that the energy consumption of *MCC* is lower than Mesh and is close to that in Torus. Energy consumption is a vital factor for many designers due to battery life limitations. This means, although *MCC* consumes more power compared to other topologies, its better performance leaves the designers with a controversial tradeoff to be made.

6. Conclusions and Future Works

Improving communication performance of the MP-SoCs has been a captivating concept for multi-processor and NoC designers in recent years. Physical design constraints play a vital role to achieve this goal and there is a tradeoff between performance and physical constraints. In this paper we have proposed a new topology to cope with the physical constraints and enhance communication performance. Our solution, called *MCC*, is based on the idea of substituting each fully-connected sub-graphs of the network with a single crossbar switch. This substitution reduces the number of outputs emanating from a single node and frees most of the wiring space. Therefore wiring complexity will be declined and we can use the additional free space to increase the bandwidth of the remaining communication links and get a better performance. Experimental results proved that our proposed topology can enhance the performance and energy consumption of the network due to bandwidth expansion, and the cost paid for such improvements is power consumption.

Our future objective is to extend the mentioned approach of bandwidth enhancement to other well-known topologies which consist of fully-connected sub-graphs.

References

- [1] Cray xt3 datasheet. *Cray Incorporation*, 2004.
- [2] Cray xt4 datasheet. *Cray Incorporation*, 2006.
- [3] I. M. Ababneh. A performance comparison of contiguous allocation placement schemes for 2d mesh-connected multicomputers. *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications*, pages 926–933, 2007.
- [4] Adriahtantenaina, H. Charlery, A. Greiner, L. Mortiez, and C. A. Zeferino. Spin: A scalable, packet switched, on chip micro-network. *Proceedings of the IEEE Design, Automation and Test in Europe Conference and Exhibition*, pages 70–73, 2003.
- [5] L. Benini and G. D. Micheli. Networks on chips: A new soc paradigm. *Computer*, 35(1):70–78, 2002.
- [6] M. Blumrich, D. Chen, P. Coteus, A. Gara, M. Giampapa, P. Heidelberger, S. Singh, B. Steinmacher-Burow, T. Takken, and P. Vranas. Design and analysis of the bluegene/l torus interconnection network. *IBM Research Report RC23025, Thomas J. Watson Research Center*, pages 926–933, 2003.
- [7] W. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. *Proceedings of DAC*, pages 684–689, 2001.
- [8] J. Duato, S. Yalamanchili, and L. Ni. *Interconnection Networks: An Engineering Approach*. Morgan Kaufmann, 2002.
- [9] A. N. F. Karim and S. Dey. An interconnect architecture for networking systems on chip. *IEEE Transactions on Computers*, 22(5):36–45, 2002.
- [10] C. Glass and L. Ni. The turn model for adaptive routing. *Journal of the ACM*, 41(5):874–902, 1997.
- [11] Guerrier and A. Greiner. A generic architecture for on chip packet-switched interconnections. *Proceedings of the IEEE Design, Automation and Test in Europe Conference and Exhibition*, pages 250–256, 2000.
- [12] A. Jantsch and H. Tenhunen. *Networks on Chip*. Kluwer Academic Publishers, 2003.
- [13] D. N. Jayasimha, B. Zafar, and Y. Hoskote. On-chip interconnection networks: Why they are different and how to compare them. *Intel Corporation*, 2006.
- [14] M. Mirza-Aghatabar, S. Koochi, S. Hessabi, and M. Pedram. An empirical investigation of mesh and torus noc topologies under different routing algorithms and traffic models. *Proceedings of the 10th EuroMicro Conference on Digital System Design Architectures, Methods and Tools*, pages 19–26, 2007.
- [15] S. Murali, L. Benini, and G. D. Micheli. An application-specific design methodology for on-chip crossbar generation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(7):1283–1296, 2007.
- [16] A. Nayebi, S. Meraji, A. Shamaei, and H. Sarbazi-Azad. Xmulator: A listener-based integrated simulation platform for interconnection networks. *Proceedings of IEEE Asian International Conference on Modeling and Simulation*, pages 128–132, 2007.
- [17] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey, and A. Sangiovanni-Vencentelli. Addressing the system-on-a-chip interconnect woes through communication-based design. *Proceedings of DAC*, pages 667–672, 2001.
- [18] A. Tavakkol. *Performance of Crossbar-based Interconnection Networks for Multiprocessors*. M.S. Thesis, Dept. of CE, Sharif University of Tehcnology, 2008.
- [19] H. Wang, X. Zhu, L. Peh, and S. Malik. Orion: A power-performance simulator for interconnection networks. *Proceeding of the 35th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 294–305, 2002.