Tree-Based 3-D Topology for Network-On-Chip

Mohammad Yahiya Khan, Sapna Tyagi and Mohammad Ayoub Khan

1College of Science, King Saud University, Riyadh, Saudi Arabia
2Institute of Management Studies, Ghaziabad UP, India
3Department of Computer Science, Sharda University, Gr. Noida, UP, India

Abstract: The increasing demand of numerous applications in consumer electronics has increased the number of computing resources in single chip. In such scenario, application needs many computing resources to build a System-on-Chip (SoC). Therefore, interconnection among Intellectual Property (IP) cores becomes another challenging issue. The performance of the network is measured in terms of throughput. The throughput and efficiency of interconnect depends on network parameters of the topology. Therefore, topology of any communication networks has an important role to play for efficient design of network. This works considers the design of an efficient tree-based topology to apply for Network-on-Chip. The degree of proposed topology is 25% less than the torus along with drastic reduction in the diameter of proposed topology. We have obtained reduced degree for proposed topology that varies from 3 to 6 for the network with k layers, while the diameter of topology is obtained as \( D = (2n-1) + k - 1 \). We have also presented and simulated an efficient routing algorithm that computes shortest path in the network.

Key words: NoC · SoC · IP · Diameter · Routing

INTRODUCTION

The concept of a system-on-chip (SoC) device evolved over the past decade, influenced by significant advances in semiconductor design and manufacturing technologies. Most notably, the consistent scaling of Moore’s Law continues to extend the capabilities and the levels of function integration (i.e. for integrated circuits or ICs). The functionality that a decade ago may have been contained on a printed circuit board (PCB) and executed by multiple discrete chips (e.g. memory, processors, interface, graphics, etc.), can now be integrated on a single IC, commonly called a SoC. To meet the growing computation-intensive applications and the needs of low-power, high-performance systems, the number of computing resources in single-chip has enormously increased, because current VLSI technology can support such an extensive integration of transistors.

SoC provides a way to reduce design effort and costs by combining multiple functions that previously existed in discrete semiconductors onto a single chip or subsystem. SoCs refers to integrating all components of a computer or any other electronic system into a single IC (chip) [1-4]. A typical SoC consists of a microcontroller, microprocessor, memory blocks including a selection of ROM, RAM, EEPROM and flash memory, timing sources including oscillators and phase-locked loops, peripherals including counter-timers and power-on reset generators, external interfaces including industry standards such as USB, Firewire, Ethernet, USART, SPI, analog interfaces including ADCs and DACs, voltage regulators and power management circuits. Likewise, each of these components may include different specifications such as variable bandwidth, buses and different communicative protocols. One example of its application is in the area of embedded systems. In most System-on-chip applications, a shared bus interconnection is adopted to communicate with each integrated processing unit. Although, busses have successfully been implemented in virtually all complex System-on-chip designs, however, such shared bus interconnection has some limitation in an environment where the number of bus requesters is large and their required bandwidth for interconnection is more than the current bus. By increasing the number of processing elements, the bus itself is transmuted into a bottleneck.

Corresponding Author: Mohammad Yahiya Khan, King Saud University, PO Box 2455, Riyadh-11451, Saudi Arabia.
Since the intercommunication requirements of SoCs made of hundreds of cores was not feasible using a single shared bus due to their poor scalability with system size, to overcome these problems of scalability and complexity, on-chip packet-switched micro-network of interconnects, generally known as Network-on-Chip (NoC) architecture was proposed [1,9]. NoC is an approach to designing the communication subsystem between IP cores in a SoC.

The work presents investigation on topology and routing with primary focus on 3-D Network-on-Chip. The investigation consists of a detailed exploration by means of the theoretical proofs, simulation and analysis. The routing strategy in 3-D considers routing at every layer apart from via interconnects. In this work, we present the results of the exploration and we discuss network topology to reduce the costs. We have also proposed tree based 2-D and 3-D topologies.

Network Topologies: Network topology is defined as the how various elements (processing elements, IP’s, DSP’s and Processors etc.) are connected physically and logically. Physical topology means the physical design of a network. Logical topology refers to how actually data is being transferred in a network. The NoC is a communication centric interconnection approach which provides a scalable infrastructure to interconnect different IPs and sub-systems in a SoC [5, 6, 7]. NoC’s can make SoC’s more structured, reusable and can also improve their performance. Since the communication between the various processing cores will be deciding factor for the performance of such systems, therefore we need to focus on making this communication faster as well as more reliable.

Characteristics of Network Topology: Topology has a great impact on the system performance and reliability. It generally influences network diameter (the length of the maximum shortest path between any two nodes), layout and wiring. Before we delve deeper into the widely used topologies, the main characteristics of Network topology which are described in below Table 1 should be understood first:

Table 1: Characteristics of Network Topology

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bisection of Network</td>
<td>A Bisection of a network is a cut that partitions the entire network nearly in half.</td>
</tr>
<tr>
<td>Throughput</td>
<td>The throughput of a network is the data rate in bits per second that the network accepts per input port. The ideal throughput is defined as the throughput assuming a perfect routing and flow control i.e. Load is balanced over alternate paths and no idle cycles on bottleneck channels [8,14,15,16]</td>
</tr>
<tr>
<td>Latency</td>
<td>The latency of the network is the time required for a message to traverse a network, i.e. a time taken for a packet, flit or message to reach from source to destination. It also includes the time taken for computing arbitration logic as well as routing computation and other delays.</td>
</tr>
<tr>
<td>Diameter</td>
<td>The diameter (D) of a network is the maximum internodes distance. The smaller the diameter of a network, the less time it takes to send a message from one node to the farthest node</td>
</tr>
<tr>
<td>Node Degree</td>
<td>The node degree (ND) is defined as the number of physical channels emanating from a node. This attribute shows the node’s I/O complexity.</td>
</tr>
</tbody>
</table>

We also present some of the widely used topologies in Network-on-Chip as follows.

2-D Mesh Topology: The switches are arranged into a 2-D lattice. The communication is allowed only between neighboring switches. Variants allow wrap around connections between switches on edge of mesh (known as TORUS). The evaluation parameters of 2-D Meshes are as follows:

- Diameter: \( \Theta(n^{1/2}) \)
- Bisection width: \( \Theta(n^{1/2}) \)
- Number of edges per switch: 4

Hypercube: The number of nodes \( n \) in hypercube is a power of \( 2(2^n) \) for k-dimensional structure [13]. For example in a 3-D Hypercube \( n = 2^3 = 8 \). The node addresses range from 0, 1, …, \( 2^n - 1 \) for k-dimensional structure. In a 3-D Hypercube node addresses are 0, 1, 2, …, 7. The node is connected to nodes whose addresses differ from it in exactly one bit position. The evaluation parameter of hypercube network is as follows:

- Diameter: \( \log n \)
- Bisection width: \( n/2 \)
- Edges per node: \( \log n \)
Binary Tree Network: In conventional network, the processors are connected indirectly through multiple routers. In such case the total number of nodes, \( n = 2^d \) processor nodes where “d” is height of the tree. For example in a Binary tree of height 3, \( n = 2^3 = 8 \) as shown in Figure 1. The maximum number of intermediate switches in order to communication from one node to another node will be \( n-1 \).

The Evaluating Parameters of Binary Tree Network Are as Follows:

- Diameter: \( 2 \log_2 n \)
- Bisect width: 1
- Edges / node: 3

Hyper-tree Network: The hyper-tree network shares low diameter of binary tree. They greatly improve bisection width. From “front” looks like \( k \)-ary tree of height \( d \). From “side” looks like upside down binary tree of height \( d \). We can evaluate \( 4 \)-ary Hyper-tree as follows:

- Diameter: \( \log n \)
- Bisect width: \( n/2 \)
- Edges / node: 6

Butterfly Network: The butterfly network is another indirect topology. The total number of nodes \( n = 2^d \) processor nodes connected by \( n \) (log \( n+1 \)) switching nodes as shown in Figure 2.

We Can Evaluate Butterfly Network as Follows:

- Diameter: \( \log n \)
- Bisect width: \( n/2 \)
- Edges per node: 4

The Mesh-based topologies are the most regular and simple architecture that is used in NoC designs. Its implementation is very simple and easy to understand and verify. These are the soul properties which are necessary for any topology design. In mesh architecture every node is connected to four of its neighbors. They need to communicate among each other for transfer of information. Various routing algorithm are used for routing packets in the network. Dimension ordered routing algorithm is the most basic routing strategy used for routing. Also, various modified and improved versions are also being worked upon and analyzed for better results. The parameters like throughput, latency and link utilization are analyzed to conclude which topology is better and efficient.

In an N-dimensional mesh network every node is connected to \( 2N \) of its neighboring nodes as shown in Figure 3. Thus the degree of a node in an N-dimensional mesh is \( 2N \). Although the number of connections per node remains fixed in a mesh network even if the size of the network increases, the performance degrades due to phenomenal increase in its diameter. Few basic evaluation parameters of a 2D-mesh, consisting of n-nodes, are given below.

- Diameter: \( (n^{1/2}) \)
- Bisect width: \( (n^{1/2}) \)
- Number of edges per switch: 4

Mesh based 3-D Topologies: Mesh architecture is the most regular and simple architecture that is used in NoC designs. Its implementation is very simple and easy to understand and verify. These are the soul properties which are necessary for any topology design. In mesh architecture every node is connected to four of its neighbors. They need to communicate among each other for transfer of information.
Torus based Topologies: A Torus network is same as a mesh network except that its terminal nodes are connected by wrap-around edges as shown in Figure 4. These wrap around edges significantly reduce the overall diameter of the network and thus improving its throughput and latency. Torus structure is the architecture which adds the wrap around edge at the border of the mesh structure and is the regular graph with fixed degree. Torus architecture adds a wrap around edge at the row and the column of the mesh respectively to achieve the ring structure.

The diameter and the network cost of the torus are just half of the mesh, the number of degree is 4, the number of the node N=n x n and the network cost is 4n in an n X n torus [12]. Although the torus architecture reduces the network diameter, the long wrap-around connections may result in excessive delay. Also scaling of the network becomes a complex task in torus topologies.

Now, we need to derive the network parameters for the proposed tree. We will find out derivation for followings:

- Total number of nodes (N)
- Average degree (d)
- Diameter of the network (D)

Number of Terminal Nodes at Level (l): The number of terminal nodes at level can be derived by induction.

\[
\begin{align*}
\text{for level } l=1, & \quad T=2 \\
\text{for level } l=2, & \quad T=3 \times 2 \\
\text{for level } l=3, & \quad T= 3 \times 2 \times 2 \\
\text{for level } l=4, & \quad T= 3 \times 2 \times 2 \times 2 \\
\text{for level } l=n, & \quad T= 3 \times 2^{n-1}
\end{align*}
\]

For otherwise,

\[
T = \begin{cases} 
3(2)^{n-1} & \text{if } l > 1 \\
2n & \text{otherwise}
\end{cases}
\]
Total Number of Nodes at Level (l): Total number of nodes in the network at given level can be derived by induction.

\[ N = \sum_{i=0}^{l} \frac{3 \times 2^{i-1}}{i} \]

We Can Also Derive the Total Number of Nodes by Another Hypothesis as Follows:

For level \( l=1 \), we have total nodes \( N = 0 + 3 \)
For level \( l=2 \), we have 3 internal nodes and 6 terminal nodes, \( N = 3 + 6 \)
For level \( l=3 \), we have 9 internal nodes and 12 terminal nodes, \( N = 9 + 12 \)
For level \( l=4 \), we have 21 internal nodes and 24 terminal nodes, \( N = 21 + 24 \)
For level \( l=5 \), we have 45 internal nodes and 48 terminal nodes, \( N = 45 + 48 \)
For level \( l=6 \), we have 93 internal nodes and 96 terminal nodes, \( N = 93 + 96 \)
For level \( l=n \), we have internal nodes and terminal nodes as, \( N = 3(2^n)-3+3(2^n) \)
\[ N = 3(2^n)-3 \]

Diameter D of Level (l) Network: The diameter of a network is the maximum inter-node distance i.e. the maximum distance between any two nodes in the topology. We have derived the \( D \) using induction as follows:

For level \( l=1 \), \( D = 1 \)
For level \( l=2 \), \( D = 1+2 = 3 \)
For level \( l=3 \), \( D = 1+2+2 = 5 \)
For level \( l=4 \), \( D = 1+3+3 = 7 \)
For level \( l=5 \), \( D = 1+4+4 = 9 \)

Average Degree of Network: This degree of a node represents number of communication ports (edges) required at each node (processsing elements). The node degree has a direct effect on the cost of each node, with the effect being more significant for parallel ports containing several wires. In the proposed topology, every internal node has a degree of 4, while every terminal node has degree of 2. Therefore, degree varies between 2 to 4.

The total number of terminal nodes at \( l \)
\[ T = 3(2^n) \]
The total number of internal nodes at \( l \) is
\[ l = 3(2^n)-3 \]

Therefore, the degree of internal and terminal nodes can be calculated as follows:
The degree of internal nodes in network is as follows:
\[ d(I) = 2(3(2^n)-3) \]

The degree of terminal nodes in network is as follows:
\[ d(T) = 2(3(2^n)) \]

Total degree of the network would be as follows:
\[ d = d(T) + d(I), d = \left( \frac{18(2^{n-1})-12}{N} \right) \]
\[ d = \frac{18(2^{n-1})-12}{3(2^n-1)}, d = \frac{18(2^{n-1})-4}{2^n-1} \]

The topology could be easily extended for 3-D as shown in Figure 6.

Proposed Algorithm and Analysis: We proposed routing algorithm for the tree-based topology as follows:

Algorithm: Compute_path (Source, Destination)/
Defining variables
STACK S, D, Path
INT source, parent, destination
**Step 1:** /* Finding out a path where there is common parent between source and destination node - Traversing from Source Node */

   a) while (abs(source) !=1)
   {
   if(source < 0)
   {
   parent=source/2;
   if(parent==0)
   {
   push(S,abs(parent));
   }
   else
   {
   push(S,parent);
   }
   }
   source=parent;
   }
   else
   {
   parent=abs(source/2);
   push(S,parent);
   source=parent;
   }
   
   b)/*Finding out a path where there is common parent between source and destination node - Traversing from destination Node */
   while(abs(destination) !=1)
   {
   if (destination < 0)
   {
   parent=destination/2;
   if(parent==0)
   {
   push(D,abs(parent));
   }
   else
   {
   push(D,parent);
   }
   destination=parent;
   }
   else
   {
   parent=abs(destination/2);
   push(D,parent);
   destination=parent;
   }
   }

**Step 2:** Push (Path, Source)

**Step 3:** Eliminate Common Path and push on stack with following exceptions:

- Source or destination node is root (i.e. 1)
- The source and destination are across the root node(i.e. 1)

**Step 4:** Push (Path, Destination)

**Step 5:** Reverse (Path)

**Step 6:** Pop (Path)

The proposed tree topology can have large number of nodes if sufficient level is choose. If we choose \( l=1 \), then we have total 3 nodes while as topology can support 3145725 nodes for \( l=20 \). The main contribution of this topology is shortest diameter as compared to torus. The Figure 7 shows a drastic reduction in the diameter for the proposed topology. For a SoC of node 3145725, the diameter of proposed tree is 39. The diameter of torus topology for same number of nodes is approximately 1800 that is too large for a NoC applications. We found that proposed topology has 25% less degree than torus. The Degree of tours for such topology will be fixed to 4. But, in proposed topology the degree varies between 2 to 3.
CONCLUSION

In this paper we have described various topologies of a network, particularly a new 3-D topology. The particular 3-D arrangement determines the possible and efficient protocol strategies. The degree of a topology can be defined as number of links connecting that node to its neighbour vertices. A topology is considered to be regular when all the routers in the network have the same degree. In this work, we have constructed a modified structure of tree that reduces the network diameter and degree drastically. The proposed tree topology can have large number of nodes if sufficient level is chosen. If we choose \( l = 1 \), then we have total 3 nodes while as topology can support 3145725 nodes for \( l = 20 \). We have also reduced the degree by 25%. The correctness of the algorithm is shown by the simulation results.

REFERENCES

