Chapter 3

RiCoBiT Architecture: A Structured And Scalable Architecture For Network On Chip Based System

This section discusses a new structured and scalable RiCoBiT for network on chip based system and comparison with mesh and torus is dealt in this chapter.

3.1 RiCoBiT (Ring Connected Binary Tree) Topology

The proposed RiCoBiT topology resembles the spider’s cob web as shown in figure 3.1. The topology consists of growing interconnected concentric rings. The rings are numbered from 1 to K where K is the number of rings in the configuration. The number of nodes in ring L is $2^L$ numbered from 0 to $2^L - 1$. Therefore, the number of nodes $N_r$ in the configuration with K rings is

$$N_r = \sum_{L=1}^{K} 2^L$$  \hspace{1cm} (3.1)

The node n in ring L is connected to its neighbouring nodes 2n and 2n+1 in ring L + 1. The number of wire segments to connect the nodes in ring L to those in ring L + 1 are therefore $2^{L+1}$. Similarly the nodes within each ring are also interconnected by wire segments. The number of such wire segments in ring L is $2^L$. The proposed topology has several advantageous properties. The topology is symmetric and regular in nature. It is structured, modular and scalable limited in size by the fabrication technology. The scalability property is well complemented with high performance with marginal increase in chip area. The performance study is presented in section three.
3.1.1 RiCoBiT Addressing

The processing nodes are placed and addressed relative to the ring to which they belong and the position within the ring. The rings are numbered starting from one (innermost to outermost) and the nodes within the ring are numbered starting from zero.

3.1.2 RiCoBiT Organisation

A node in RiCoBiT communicates with its five adjacent nodes through its five half duplex link interfaces. The communication between two adjacent nodes through their corresponding interface is depicted in figure 3.2. A node sends a request signal from its link interface to the corresponding one of the adjacent node to send a packet of size \( p \) bits from the link interface send register and waits for its acknowledgment. The receiving link interface acknowledges the request when its receive register is free. Let the time to acknowledge be \( T_a \) time units. The minimum clock time \( T_a(\text{Min}) \) is 2 clock cycles. On receiving back the acknowledgment signal, the requesting link interface sends the packet to the node in \( p \) clock cycles. The adjacent node receives the packet and stores it in its link interface receive register. If the node happens to be the destination for the packet it is delivered there itself and exits the network.
Else, the RiCoBiT routing algorithm presented in section 3.1.4 determines in $T_p$ less than $p$ time units the next node to be traversed by it towards its destination. If the send buffer of size $N_b$ packets for the link interface to the next node is not full it is stored therein. It waits in the receive register otherwise. A new packet originating at the node is similarly placed in the send buffer of one of the five link interfaces of the node according to the routing algorithm. Therefore, one packet from each node interface may enter the node in parallel in a receive time slot. The ones with this node as their destination exit the network. A new packet from the node may enter if less than four transit packets arrived or five arrived and at least one left. One packet from each send register may leave the node in parallel in a send time slot. While a packet leaves the send register a packet from the send buffer may enter it and one from a receive register may move to the send buffer. The send and receive time slots occur alternately. Each slot is $p$ clock cycles long. Thus six packets at most enter the node in a receive time slot, five at most from receive registers and five at most from the send registers leave in a send time slot. If a packet moves from a send register to the next node and one from the receive register enters the send buffer, $T_a$ is $2p$. In the extreme case when the packets from each of the five registers are to pass through the same send buffer, they leave the node serially one-by-one and $T_a$ varies from $2p$ to $T_a(\text{max}) = 10p$. The input rate to the node is accordingly reduced. Considering this, the value of $N_b$ is selected as five. The transit time of a node at a node, therefore, ranges from $2p$ to $(T_a(\text{max}) + 2p) \times (N_b + 2)$. Due to the interaction between the send buffer and receive register across the adjacent nodes through request and acknowledgment there is no packet loss in the network. In view of the entry of a new packet into the network, node by-node transit of packets towards their respective
destination and the exit from the network on reaching the destination in the manner explained above, there is no congestion in the network.

3.1.3 RiCoBiT Routing Algorithm

This section presents an optimal routing algorithm for RiCoBiT. The proposed algorithm to route the packet from the source to the destination is presented below.

Step 0: Check destination address. Initialize current src as node address and current dest as destination address.

Case 1: current src and current dest in same ring.

1) Compute the minimum hop count between current src and current dest. If it is greater than 3, set current src and current dest as their parent nodes respectively in the adjacent ring below.
2) Repeat Step 1 till shorter hop count is greater than 3.
3) Move to the next node from current src towards current dest along the minimum hop count path. Set the current src as the next node.
4) Repeat Step 3 until current src is equal to current dest.
5) Consider a complete binary tree with top right node of the current src as root. If destination node lies in the tree, mark the right node as set current src else mark the sibling of the left node as current src. Move to the current src.
6) Repeat Step 5 until current src is equal to destination address.

Case 2: When destination node is in a ring above

1) Move one step at a time from the current dest to the ring below until ring of the current src is reached. Set current dest as the node obtained in the ring of the source.
2) Repeat the steps as in Case 1.
3) Repeat Step 5 and 6 of Case 1.

Case 3: when destination node is in a ring below that of the source node
1) Move one step at a time from current src to the ring below until the ring of current dest is reached. Set current src as the node obtained.
2) Repeat the procedure as in case 1.

A packet when received by a node is checked by the routing logic of the interface. The logic is built around three cases based on the ring to which the current source node and the destination of the packet respectively belong. If the current source node and the destination of the packet are in the same ring, the packet is routed to the parent node on the adjacent ring until the node difference is greater than three. Then the node traverses the lateral path along the shortest route to current destination. After reaching the current destination, the packet travels up the ring towards the destination. Whenever the destination is above the current node, the packet is initially routed along the same ring to the parent node of the destination. Then the packet moves up towards the destination. Whenever the destination node is below the current node, the packet moves towards the ring of the destination along the parent node of current node. Then the routing logic routes the packet towards the destination.

From the above, it is evident that the packet uses binary tree traversal which is known for its optimal routing and a lateral traversal through the shortest path to reach the destination. This proves that the packet is routed through the shortest path. Therefore the proposed algorithm is optimal.

**Statement:** The proposed algorithm is optimal. i.e the routing algorithm would route the packet from the source to the destination for any configuration with K rings through the shortest path. The statement is proved by mathematical induction.

**Case 1: K = 1**

Let us consider the algorithm for the smallest case i.e for ring K = 1 with \( N = \sum_{L=1}^{K} 2^L = 2 \) nodes. The source and destination being adjacent to each other, there are two alternate paths each consisting of single hop. Note that this is the shortest possible.

**Case 2: K = 2**
1) If source and destination are both in ring 1 then it is same as case 1
2) If the source and destination are both in ring two, then the shortest path consists of one hop in case source and destination are adjacent to each other. Otherwise the shortest path consists of two hops.
3) If the source and destination are in two different rings then the shortest path consists of one hop in case they are adjacent. Otherwise the shortest path consists of two hops. It is noted that such a path has an intermediate node in ring one which is adjacent to source and destination both.

Therefore it is of length two hops and is shortest. When the intermediate node is in ring one, it is concluded that if a configuration with a single ring offers shortest path between adjacent nodes in ring one then a configuration with two rings also offers shortest path for any source destination pair in two different rings. From this we infer that, there is a shortest path between any source destination pair when the

**Case 3: \( K = 3 \)**

1) If source and destination are confined to ring one and ring two, then it is same as Case 2.
2) If the source and destination both are in ring three, then the shortest path consists of one hop in case these are adjacent to each other. Otherwise there is a shortest path along the ring three consisting of two hops or three hops or four hops if there are one or two or three intermediate nodes between source and destination pair respectively. In case of three intermediate nodes there are alternate shortest paths of four hops each which pass through ring two.
3) If one of the nodes is in ring three and the other one in ring two, then the shortest path is of one hop when the source and destination are adjacent to each other. Else there exists an intermediate node in ring two which is adjacent to the node in ring three. According to the Case 2, it is known that there exists a shortest path for any source destination pair for configuration of two rings. Hence there exists a shortest path between a node in ring three and any node either in ring one or two.

From this we infer that, there exists a shortest path between any source destination pair when the configuration has three rings.
Case 4: K = 4

1) if source and destination are confined to ring one, ring two and ring three, then it is same as Case 3.
2) If the source and destination both are in ring four, then the shortest path consists of one hop in case these are adjacent to each other. Otherwise there is a shortest path along ring four, consisting of two hops or three hops or four hops if there are one or two or three intermediate nodes between source and destination pair respectively. In case of three intermediate nodes there are alternate shortest paths of four hops each which pass through separated by more than four hops in ring four, then the shortest path passes through ring three to the destination at ring four. The number of hops in such shortest path vary from four to six on case to case basis.

3) If one of the nodes is in ring four and the other node in ring three, then the shortest path is of one hop when the source and destination are adjacent to each other. Else there exists an intermediate node in ring three which is adjacent to the node in ring four. According to the case of ring three, there is a shortest path for any source destination pair for configuration of three rings.

Hence there is a shortest path between a node in ring four and any node in other rings. From this we infer that, there exists a shortest path between any source destination pair when the configuration has four rings.

Case K - 1:
Assume that there are shortest paths between any source destination pair in a configuration consisting of K - 1 rings.

Case K:

1) If the source and destination both are in ring K, then the shortest path consists of one hop in case these are adjacent to each other. Otherwise there is a shortest path along the same ring consisting of 2 hops or 3 hops or 4 hops if there are one or two or three intermediate nodes between source and destination pair respectively. When the source and destination are
separated by more than 4 hops in ring K, then the shortest path passes through ring K - 1 to the destination at ring K.

2) if one of the nodes is in ring K and the other node in ring K - 1, then the shortest path is of one hop when the source and destination are adjacent to each other. Else there exists an intermediate node in ring K -1 which is adjacent to the node in ring K.

In view of the assumption for the Case K - 1 above, there is a shortest path between any source destination pair for configuration of K rings passing through the adjacent intermediate node between ring K and K - 1.

From the above discussion it is evident that the routing algorithm routes the packets from any source to any destination through the shortest route. Thus the proof.

3.2 Performance parameters and comparison

In this section, the performance of RiCoBiT is studied and compared with two of the most popularly used architectures in Network on Chip based systems namely 2D mesh and torus.

In this comparison, some important parameters like the maximum hop for a given order (size / no of nodes), the number of wire segments and the wire length used for the interconnection are considered. These are the major factors governing the system performance in terms of latency, throughput and area of a given design which in turn are decisive factors for power consumption and dissipation.

3.2.1 Maximum Hop Count

Consider the set of shortest path(s) between all possible source destination nodes pairs for a given configuration. The largest element in the set is referred to as maximum hop count. As the number of hops traversed by a packet from the source to the destination increases, the latency of the system also increases and the performance dips. Hence this is one of the most important parameters which determines the performance of a system.

In case of RiCoBiT, the packet travels from the source to the destination with a maximum hop count

\[ H_c(\text{Max}) = 2 \log_2(N_r +2) - 4 ; N_r \text{ is the no of nodes} \]  

(3.2)
Proof:

In a complete binary tree of \( N_r \) nodes, the maximum distance from the root to the leaf node is expressed as \( \text{Max Distance} = \log_2(N_r+1) - 1 \). Therefore the distance from the extreme left to the extreme right would be twice of the above expression

\[
\text{Max Distance} = 2 \times \log_2(N_r+1) - 2 \quad (3.3)
\]

By Construction, complete the figure to form a balanced complete binary tree with \( N_r + 1 \) nodes

\[
\text{Max Distance} = \log_2(N_r+2) - 1 \quad (3.4)
\]

Therefore the distance from the extreme left or the extreme right would be twice of the above expression

\[
\text{Max Distance} = 2 \times (\log_2(N_r+2) - 1) = 2 \times \log_2(N_r+2) - 2 \quad (3.5)
\]

Now removing the two extra edges from the construction we get

\[
\text{Max Distance} = 2 \times \log_2(N_r+2) - 2 - 2 = 2 \times \log_2(N_r+2) - 4 \quad (3.6)
\]

Thus the proof.

Now let us consider a 2D mesh. It is one of the most popular, simplest and widely used topology in Network on Chip based systems. In this topology, the processing modules are placed at the intersection of the row and column as shown in the Figure 3.3.

![Fig. 3.3 A 2D mesh of M rows and N columns showing the modules at the intersections with the addressing](image)
In the comparison of the parameter maximum hop, we consider the proposed architecture of L rings with $N_r = \sum_{i=1}^{L} 2^i$ nodes. This is compared with a 2D mesh with $M$ nodes along its row and $N$ nodes along its column having total $MN$ nodes. The equation below shows the relation between the number of nodes of 2D mesh and RiCoBiT

$$N_r = MN - 2 \quad (3.7)$$

Where $N_r$ is the number of nodes in RiCoBiT and $MN$ is the number of nodes in 2D mesh. Whenever $MN \geq 2$, $N_r$ tends to $MN$ or $N_r \approx MN$. For example, in 2D mesh of 32 rows and 32 columns we have $MN = 1024$ nodes, which is compared to $N_r = MN - 2 = 1024 - 2 = 1022$ nodes of RiCoBiT.

Extending the above equation for a square mesh, the equation reduces to

$$N_r = N^2 - 2 \approx N^2 \approx \sqrt{N_r} \quad (3.8)$$

In case of a 2D mesh, the maximum hop would be when a packet can travels from the corner node to its diagonally opposite corner. This is expressed as

$$H_{c}(\text{Max}) = M + N - 2 \quad (3.9)$$

Where $M$ and $N$ are the order of the mesh and $MN$ is the number of nodes.

In case of a regular 2D mesh of equal rows and columns the expression reduces to

$$H_{c}(\text{Max}) = 2N - 2 \quad (3.10)$$

$N$ is the order of the mesh and $N^2$ is the number of nodes.

Taking number of nodes $N_r$ into consideration, we can rewrite the above the above equation as

$$H_{c}\text{Max} = 2\sqrt{N_r} - 2 \approx 2\sqrt{N_r} \quad (3.11)$$
Fig. 3.4 A 2D Torus of M rows and N columns showing the modules at the intersections with the addressing

Torus is another popular, simple and widely used topology in Network on Chip based systems. The topology is evolved from 2D mesh by adding an extra edge between the end points of the mesh topology. Like 2D mesh, the processing modules are placed at the intersection of the row and column as shown in the Figure 3.4 above.

The discussion on the number of nodes of torus and its comparison with RiCoBiT remains the same as in the previous case. The maximum hop parameter in the case of torus is when the packet travels from the corner node to the middle of the system and is given by

\[ H_c(\text{Max}) = |M/2| + |N/2| \]  \hspace{1cm} (3.12)

M & N are the order of the mesh and MN is the number of nodes.

Reducing the equation for same dimension of rows and columns

\[ H_c(\text{Max}) = 2 \times |N/2| \]  \hspace{1cm} (3.13)

N are the order of the mesh and \( N^2 \) is the number of nodes.

Taking number of nodes \( N_r \) into consideration, we can rewrite the above the above equation as

\[ H_c(\text{Max}) = 2\sqrt{N_r} - 2 \]  \hspace{1cm} (3.14)

Where \( N_r \) is the number of nodes.

The graph (Fig. 3.5) compares the maximum hop of the mesh and torus with RiCoBiT. From the graph and
equations, it is quite evident that the maximum hop is highest in mesh. It is half for torus as compared to mesh and is lowest in the proposed architecture. As the number of modules scales up, the parameter rises substantially in both mesh & torus but the growth is very slow in case of our proposed architecture. This implies that the performance is not compromised and outperforms when design is scaled up.

3.2.2 Average Hop Count

As a part of the comparison we also compute an average value of the hop count $H_c(Avg)$. This parameter helps us to compare the overall performance of the packet transfer in the system. This parameter is calculated by taking the arithmetic mean of the hop counts for all possible node pairs. From the graph (Fig 3. 6), it is evident that the average hop is substantially less when compared to the other architectures. This also proves that the proposed architecture performs well when compared to other architectures.

3.2.3 Latency and Throughput

Consider the route of a packet from its source to destination determined by the routing algorithm. Note that this is the shortest path between the source destination pair. Let this be max hop path with length $H_c(Max)$. Referring to section 3.3.3, the latency, $L_p$, of a packet following this path may be bounded as
The throughput $Th$ therefore for this route may be bounded as

$$\frac{1}{(Ta_{\text{Max}} + 2X) \times (N_b + 2 \times Hc_{\text{Max}}) + P}$$

(3.16)

The bounds for a route of average length corresponding may be obtained by replacing $Hc_{\text{Max}}$ with $Hc_{\text{Av}}$. In view of the maxhop and average hop comparative study presented in section 3.2.1 and 3.2.2 it is to be observed that the latency and throughput performance of RiCoBiT is substantially superior as the topology scales up.

### 3.2.4 Number of Wire Segments

This is another important parameter which would govern the area and routing complexity of the design. If the design has more wire segments, it would increase the area, power consumption and also pose difficulty for CAD tool to place and route the wire segments. The parameter expressed in terms of number of nodes $N_r$ for the proposed architecture is

$$\text{No. of Segments } W = 2 \times N_r - 2$$

$$\approx 2 \times N_r$$

(3.17)
**Proof:** Let us prove the above equation by method of induction.

Consider the equation for the base case with 2 nodes at ring 1

No Of segments = \( W = 2 \times 2 - 2 = 2 \)

The statement is true for \( N = 2 \)

Now let us express the number of nodes \( N \) in terms of ring \( L \)

\[ N = \sum_{i=1}^{L} 2^i \]

Lets the equation for some ring \( K \) be true

\[ W = 2 \times \sum_{i=1}^{K} 2^i - 2 \]

Now let us prove for ring \( K+1 \)

\[ W = 2 \times \sum_{i=1}^{K} 2^i - 2 + 2^{K+1} + 2 \times 2^K \]

Where the term \( 2^{K+1} \) represents the number of links to interconnect nodes at \( K+1 \) ring and the term \( 2 \times 2^K \) is for the number of links to interconnect the nodes at ring \( K \) and \( K+1 \).

Reducing the above equation

\[ W = 2 \times \sum_{i=1}^{K} 2^i - 2 + 2^{K+1} + 2^{K+1} \]
\[ = 2 \times \sum_{i=1}^{K} 2^i + 2 \times 2^{K+1} - 2 \]
\[ = 2 \times \left( \sum_{i=1}^{K} 2^i \right) - 2 \]
\[ = 2 \times N - 2 \]

Where the number of nodes at ring \( K+1 = \sum_{i=1}^{K+1} 2^i \)

Thus the proof.

The number of wire segments for a \( M \times N \) order 2D mesh is given as

\[ W = 2M \times N - M - N \] \hspace{1cm} (3.18)

\( M \) & \( N \) are the order of the mesh and \( MN \) is the number of nodes.

Reducing the equations for equal rows and columns
\[ W = 2N^2 - 2N \]
\[ = 2N (N - 1) \] 

(3.19)

N is the order of the mesh and \( N^2 \) is the number of nodes.

Considering the number of nodes \( N_r \), we can express the relation as

\[ \text{No. of Segments} \ W \approx 2N_r \] 

(3.20)

In the case of torus, the number of wire segments would increase due to the addition of extra wire segments to join the end points. This would give rise to an expression as given below.

\[ W = 2 M \times N \] 

(3.21)

Reducing the equations for equal rows and columns

\[ W = 2 N^2 \]

Expressing the above expression in terms of number of nodes \( N_r \),

\[ \text{No. of Segments} \ W \approx 2N_r \] 

(3.22)

![Comparison Number of Wire Segments](image)

**Fig. 3.7 Graph Plotting The Number Of Segments For All The Topologies**

From the graph (Fig. 3.7), we could observe that the number of segments is almost the same as other architectures. This implies that the area requirement of RiCoBiT for a given number of nodes would remain the same with advantage of performance over the others.
3.2.5 Wire Length

This is also an important parameter affecting the performance and area of the design. The same effects of number of wire segments hold good even in this parameter. For considering the wire length of the proposed architecture, we consider unit length for interconnects between the modules with an additional wire of length equal to the number of nodes to connect the ends. The parameter for the proposed architecture is calculated as

\[
WL = 3N_r - 4 \\
\approx 3N_r
\]  

(3.23)

Proof:

Let us prove the above equation by method of induction.

Consider the equation for the base case with 2 nodes at ring 1

\[
\text{Wl} = WL = 3 \times 2 - 4 = 2
\]

The statement is true for \( N = 2 \) Now let us express the number of nodes \( N \) in terms of ring \( L \)

\[
N = \sum_{i=1}^{L} 2^i
\]

Let the equation for some ring \( K \) be true

\[
WL = 3 \times \sum_{i=1}^{K} 2^i - 4
\]

Now let us prove for ring \( K+1 \)

\[
WL = 3 \times \sum_{i=1}^{K} 2^i - 4 + \{2^{K+1} - 1\} + 2 \times 2^K + \{(2^{K+1} - 1) + 2\}
\]

Where the term \( 2^{K+1} - 1 \) represents the number of links to interconnect nodes at \( K + 1 \) ring and the term \( 2 \times 2^K \) is for the number of links to interconnect the nodes at ring \( K \) and \( K + 1 \).

Considering a unit length for each link, the number of links would be equal to the wire length. Also we would require \( \{(2^{K+1} - 1) + 2\} \) units for interconnecting the two end points. Reducing the above equation

\[
WL = 3 \times \sum_{i=1}^{K} 2^i - 4 + 2^{K+1} + 2^{K+1} + 2^{K+1} \\
= 3 \times \sum_{i=1}^{K} 2^i + 3 \times 2^{K+1} - 4 \\
= 3 \times \{ \sum_{i=1}^{K} 2^i + 2^{K+1}\} - 4
\]
\[= 3 \times \left\{ \sum_{i=1}^{K+1} 2^i \right\} - 4\]
\[= 3 \times N - 4\]

Where the number of nodes at ring \(K+1 = \sum_{i=1}^{K+1} 2^i\)

Thus the proof.

Similarly considering a unit and equal length for all the interconnects, the wire length for 2D mesh of order \(M \times N\) can be computed as

\[WL = 2 \times M \times N - M - N\]  \hspace{1cm} (3.24)

Reducing the equations for equal rows and columns

\[WL = 2N^2 - 2N\]
\[= 2N (N - 1)\]  \hspace{1cm} (3.25)

Expressing the above in terms of number of nodes \(N_r\),

\[WL \approx 2N_r - 2\sqrt{N_r}\]
\[\approx 2N_r\]  \hspace{1cm} (3.26)

In case of torus, considering a unit and equal length for all the interconnects between internal nodes and the length of the end points wires to be approximately equal to the order of the torus, the wire length can be computed as

\[WL = 4 \times M \times N - 2M - 2N\]  \hspace{1cm} (3.27)

Reducing the equations for equal rows and columns

\[WL = 4N^2 - 4N\]
\[= 4N(N - 1)\]  \hspace{1cm} (3.28)
Expressing the above in terms of number of nodes $N_r$,

$$\text{WL} \approx 4N_r - 4\sqrt{N_r}$$

$$\approx 4N_r$$ \hspace{1cm} (3.29)

It may be noted from the graph (Fig 8) that the wire length of the proposed architecture is more than mesh but is lesser than torus. We could take this marginal hike in the wire length for the performance that it would deliver. From the above discussion, it is proved that the proposed architecture is better in terms of performance with those of mesh and torus.

### 3.3 Concluding Remarks

The chapter introduces a new topology called RiCoBiT for network on chip based system along with the mathematical comparison with proof for mesh and torus. It is found that maximum hop and average hop is very less for the proposed topology while the wirelength and the number of wire segments remains almost the same. This proves that the proposed topology is better when compared to mesh and torus.