Chapter 2

Mesh And Torus : A Performance Comparison

This section discusses most used architecture in Network On Chip.

2.1 Mesh And Torus

2D mesh and torus are the most popular, simplest and widely used topologies in Network on Chip based systems. In these topologies, the processing modules are placed at the intersection of the row and column as shown in the figure (Fig 2.1). Torus topology is obtained from 2D mesh by adding an extra edge between the end points of the mesh topology.

(i) Mesh

(ii) Torus

Fig 2.1 2D Mesh and Torus showing the modules at the intersections with the addressing

In this discussion for a given order (size / number of nodes), we consider some important parameter like the maximum hop count, the average hop count, the number of wires as interconnects and the wire length for the interconnections. These are the important parameters which govern latency, throughput and chip area for a given design. These in turn would be decisive factors in determining power consumption / dissipation

2.2 Addressing

The modules are addressed based on the relative position in the X-Y co-ordinate system. The address starts from (0,0) to (M-1, N-1). (0,0) indicates top left corner location and (M-1,N-1) indicates bottom right corner location of MXN matrix.
2.3 Maximum Hop Count (Worst Case)

This is a very important parameter which has a direct impact on the performance of the system. As the number of hops which a packet takes in reaching from the source to destination increases, the latency of the systems is also increased and the performance dips.

In a 2D mesh of order M X N, there are MN number of modules. The maximum number of hops that a packet can travel is from the corner node to its diagonal opposite corner which is expressed as

\[ H_{\text{max}}(\text{Mesh}) = M + N - 2 \]  \hfill (2.1)

In the case of regular 2D mesh wherein the rows and columns are equal, Maximum Hop can be reduced to

\[ H_{\text{max}}(\text{Mesh}) = 2N - 2 \]  \hfill (2.2)

In case of a 2D torus, the maximum hop would be for a packet travelling from the corner to the middle of the system which can be expressed as

\[ H_{\text{max}}(\text{Torus}) = \lfloor M / 2 \rfloor + \lfloor N / 2 \rfloor \]  \hfill (2.3)

Reducing the equation for same dimension of rows and columns

\[ H_{\text{max}}(\text{Torus}) = 2 \lfloor N / 2 \rfloor \]  \hfill (2.4)

From the above equations, it is to be noted that the max hop count for torus is half of that in mesh for a larger configuration. The graph (Fig 2.2) illustrates the variation of Hmax with the total number of node in the configuration.
2.4 Maximum Hop (Average Case)

As a part of the comparison we also compute an average value of the maximum hop count. This parameter helps us to compare the performance of the packet transfer averaged over all the source destination pairs in the configuration. This parameter is calculated by taking the arithmetic mean of the maximum hop count for all possible module pairs. By this comparison, we are looking at the overall working of the system. The Figure 2.3 shows the comparison. It is evident that torus performs better. The average maximum hop count for torus as compared to that in mesh improves as the configuration size increases.

Fig 2.2 Graph Plotting the Maximum Hop of Torus & Mesh

Fig 2.3 Graph Plotting Average Hop for Torus & Mesh
2.5 Number of Wire Segments

This is another important parameter which governs 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 number of wire segments $Nw$ (Mesh) for an $M \times N$ order 2D mesh is given as

$$Nw \ (Mesh) = 2 \ MN - M - N = 2MN - (M + N) \tag{2.5}$$

$$\approx 2MN \text{ if } MN \gg M + N$$

Reducing the equations for equal rows and columns

$$Nw \ (Mesh) = 2N^2 - 2N = 2N \ (N - 1) \tag{2.6}$$

$$\approx 2N^2 \text{ if } N \gg 1$$

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

$$Nw \ (Torus) = 2 \ MN \tag{2.7}$$

$$= 2N^2 \text{ if } M = N \tag{2.8}$$

The number of wire segments in mesh is lower than that in torus. However this advantage of mesh diminishes as $MN \gg (M + N)$. The Figure 2.4 shows the variation of this parameter with size of the configuration.

2.6 Wire Length

This is an important parameter affecting the performance and area of the design. The same effects as that of number of wire segments hold good for this parameter too. Considering a unit and equal length for all the interconnects, the wire length for mesh can be computed as

$$Lw \ (Mesh) = 2 \ MN - M - N = 2MN - (M + N) \tag{2.9}$$

$$\approx 2MN \text{ if } MN \gg M + N$$

Reducing the equations for equal rows and columns

$$Lw \ (Mesh) = 2N^2 - 2N = 2N \ (N - 1) \tag{2.10}$$

$$\approx 2N^2 \text{ if } N \gg 1$$

Considering a unit and equal length for all the interconnects and the wire length connecting end points wires to be approximately equal to the order of the torus, the wire length of torus can be computed as

$$Lw \ (Torus) = 4 \ MN - 2M - 2N = 4MN - 2(M + N) \tag{2.11}$$

$$\approx 4M \text{ if } MN \gg M + N$$

Reducing the equations for equal rows and columns
\[
Lw (\text{Torus}) = 4N^2 - 4N = 4N (N - 1) 
\approx 4N^2 
\] (2.12)

A comparison shows that the wire length in mesh are approximately half in that of torus. The approximation is much better if \(MN >> M + N\). The figure 2.5 shows the variation for wirelength Vs the number of nodes in the two configuration.

Fig 2.4 Graph Plotting Number of Wire Segments for Torus & Mesh

Fig 2.5 Graph Plotting Wire Length for Torus & Mesh
2.7 Verification

The above mentioned results are verified and found to be correct by simulating the two topologies. Basic concepts of graph theory like adjacency matrix and all pair shortest path algorithm are used for it. The topologies are represented using a graph wherein the vertices represent the processing nodes and the edges are wire segments. The algorithm is provided with the size of the topologies using which it generates graph $G( V, E)$. The graph $G( V, E)$ is given as an input to adjacency matrix generator producing the adjacency matrix $ADJ[ ][ ]$. This in turn serves as an input to an all pair shortest path algorithm to get the cost matrix $C[ ][ ]$. Using these matrices we obtain values for each of the parameters. For instance, the maximum value in cost matrix $C[ ][ ]$ denotes the maximum hop for a given size.

\[ H_{\text{max}} = \max ( c[i][j]) , 0 \leq i \leq M-1 & 0 \leq j \leq N-1 \]  \hspace{1cm} (2.13)

Similarly other parameters are calculated either from adjacency matrix or cost matrix.

```pseudo
pseudo_code_parameter(int size)
{
    int c[][], adjmat[][ ];
    g = createGraph(size);
    adjmat[][ ] = generate_adjacency();
    c[][ ] = all_pairs_shortest_path(adjmat);
    maxhopcount = max(c[][ ]); 
    avghopcount = avg(c[][ ]); 
    wirelength = calculate_wl(adjmat[ ]); 
    wiresegcount = calculate_seg(adjmat[ ]); 
}
```

2.8. Concluding Remarks

From the above discussion, it is clear that the maximum hop count and average hop count of mesh increases as the configuration size increases. This means the performance degrades as the size increases. However, In the case of torus, the performance does not degrade substantially but the increase in wire length is substantially high which would add to the area and power requirement of the system.
It is evident from the above discussion that the maximum hop count and average hop count increase in both the cases. This implies the time taken for a packet to be transferred within the system increases. This brings to light as the size of the design scales up, the performance gets degraded. Comparing the two topologies, it should be noted that the torus performs better than mesh.

Taking into consideration the area factor, the comparisons brings out that the wire length which contributes to the area and power consumption / dissipation is a negative factor in torus when compared to mesh. In view of the above, a new structured and scalable architecture for network on chip based systems is required. It should be symmetric / regular in structure and power efficient.