

Another Approach of Mesh-Connected trees

Gene Eu Jan[†]

Shao-Wei Leu

Cheng-Hung Li

Wan-Rone Liou

National Taipei University[†]
National Taiwan Ocean University

Department of Computer Science[†]
Department of Electrical Engineering

[†]E-mail: b0199@mail.ntou.edu.tw E-mail: b0119@mail.ntou.edu.tw E-mail: Kristof_lee@ds.ee.ntou.edu.tw E-mail: b0222@mail.ntou.edu.tw

Abstract-Mesh network is commonly seen in parallel computation network, as it is straightforward and easy to expand. Two-dimensional mesh network also attract considerable attention in the VLSI layout due to its simplicity. In applying binary tree to the layout of VLSI array in the form of H, better flexibility and regularity might be achieved. Based on the advantages of mesh architecture and binary trees, we present a new approach of mesh-connected trees that are different to those previously. They unite mesh architecture and binary trees into a new architecture and apply it to the layout of VLSI array. This architecture is free from the problems of crossing, saving the area of overall layout. In addition, this improves the problem that traditional meshes of trees and mesh-connected tree difficult to expanded in terms of number of levels. Furthermore, now that we have another mesh-connected architecture, the new mesh-connected tree architecture provided by us will be more efficient than traditional binary tree. Concentrating on the communications between binary trees.

Key word: VLSI layout, VLSI array, binary tree, mesh of trees, mesh-connected tree.

1. Introduction

Recently, with advances in semiconductor technologies, various parallel computation structures are presented one after another. Among which mesh-connected computers [2,7] still attract considerable attention because of their simple and regular architecture. In a parallel system, an effective interconnection network [4,8,11,17] is used to exchange data between processors. Mesh-connection [11,13] is a very common multiple processing architecture, which is used to solve scientific problems. In parallel computation network it is easy to expand and simple. Moreover, mesh-connection is frequently used in VLSI layout due to its simplicity.

The purpose of our study is to embed mesh-connected trees in two-dimensional structures, while making proper arrangements so as to reduce the cost

of connection and improve the overall performance of the system. The two-dimensional structure obtained through conversion is very suitable for the design of chips. The reason being that after proper arrangements, all the nodes can use smaller chip areas and reduce the possibility of intersection of link between the nodes. In addition, it can also reduce the complexity while reducing costs in designing and testing.

A $n \times n$ two-dimensional mesh-connected structure has N Processor Element (PE): P_0, P_1, \dots, P_{N-1} , where $n = N^{1/2}$. There is a node $P(i, j)$ on two-dimensional mesh-connected architecture. i, j stands for row and column, respectively, where $0 \leq i \leq n-1$, $0 \leq j \leq n-1$. There are four neighbors, which are $P(i+1, j)$, $P(i-1, j)$, $P(i, j+1)$, $P(i, j-1)$, respectively. Figure 1 shows a two-dimensional mesh-connected architecture. Two-dimensional mesh has $N = n^2$ nodes, the number of links $E = 2n(n-1) = 2N - 2n$, network diameter $D = 2(n-1)$, link necessary to mesh of N nodes is $O(N)$. This is a less than fully connected graph, capable of being separated into different areas by $O(\sqrt{N})$ links, and suitable for being used in interconnected networks. In a mesh-connected diagram, there is no intersection between links, which are symmetric in nature, extremely suitable for designing VLSI circuits.

When implementing multiple processor arrays [1,10,12,16] with VLSI, tree structure is a very attractive choice, as the architecture connects is simple and regular. In consideration of the cost of VLSI, the cost of computation element is lower than that of connection [5,6]. If each processor element only maintains tree structure with its immediate neighbor, the cost of connection might be reduced. Apart from root nodes and leaf nodes, each node of

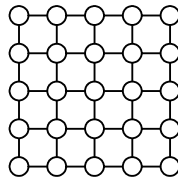


Fig. 1. A 5x5 mesh-connected architecture.

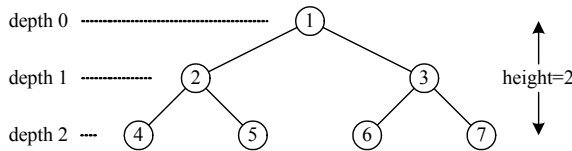


Fig. 2. A 3-level binary tree.

tree-connection is connected to its father node and two child nodes. Each node in the figure is connected downward to contrasting nodes through two different links. The nodes of the trees are divided into leaf

nodes (also knows as terminal nodes) and non-terminal nodes. The height of the tree is also called the depth, meaning the maximum level of the figure. This is shown in figure 2, in which node 1 is depth 0, node 2 and 3 are depth 1 and node 4, 5, 6 and 7 are depth 2. Node 1, 2, 3 are non-terminal nodes, while node 4, 5, 6 and 7 are terminal nodes. When the degree of non-terminal nodes (excluding root nodes) is three and the terminal nodes are of the same level, we call this binary tree “a complete binary tree”, and figure 2 is a complete binary tree. h level complete binary tree has a graph node number, which is $N = 2^{h+1} - 1$ and has $E = N - 1$ links. The graph diameter is $D = 2 \log_2(N + 1) - 2$, therefore, complete binary tree $E = O(N)$, $D = O(\log_2 N)$. Due to the clear and orderly structure of binary trees, the subordinate relationships of the data from the top down can be seen clearly. These are appropriate for use in database systems. In terms of transferring and arrangement, the root nodes of binary tree structure are usually the bottleneck of transferring data.

According to the references in Leighton [3] and Efe [11], its arranging method may have the problem of crossing the wire and cause increases in overall layout costs. In addition, Leighton [3] the numbers of levels of binary trees are limited in second level and unable to increase. With this in mind, we present this topology to the problem of traditional mesh-connected tree crossing the wire on VLSI.

In the early 90s, Bhattacharya [1] used to embed quadtree in rectangular mesh architecture, but this embedding method may have had problems. For example, crossing the wire, poor extensibility and availability after being embedded. Hence, this paper will, on the basis of Dotted Triangle given by

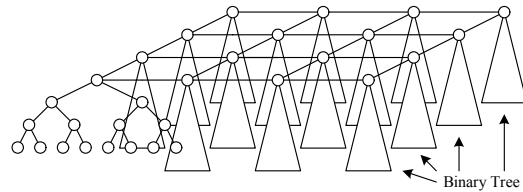


Fig. 3. A mesh-connected tree made up of 4x4 mesh-connected and 3 depth binary trees.

Bhattacharya, present a new type of two-dimensional embedding method for mesh-connected tree.

2. Structural Properties

In this paper, we assume a network is seen as an undirected graph, whose nodes represent processor elements (PEs) and whose edges represent bidirectional links between the processor elements. Our topology, as shown in figure 3, is made up of one mesh-connected and binary tree. The number of nodes for our topology is $N = n^2 \times (2^{h+1} - 1)$, where h is for the height of the binary tree while time complexity is $O(n^2 \cdot 2^h)$. The number of links for the graph is $L = (2n^2 - 2n) + n^2(2^{h+1} - 2)$ with the time complexity of $O(n^2 \cdot 2^h)$, and average degree is 4.

When message transfer is required for the nodes of any two binary trees, transfer may start from the leaf to the root, whereby $2 \log_2(N + 1)$ step is required to reach the root. When transferring by means of mesh-connected architecture, $(2\sqrt{N} - 2)$ step is required, so the step consumed by transference of any two nodes is at most $2 \log_2(N + 1) + (2\sqrt{N} - 2)$. When transferring message, mesh-connection may be considered as global communication, while binary tree as local communication.

3. Node utilization for VLSI array

Layout method may exert great influences on overall performance and costs of VLSI [1,5,6,9]. If the intersection of wires can be reduced and the chip area can be saved by means of arranging, it will be helpful in improving performance and reducing costs significantly. Moreover, in seeking the most effective use of chip area, consideration must be given to leave enough space between the nodes for the purpose of wiring.

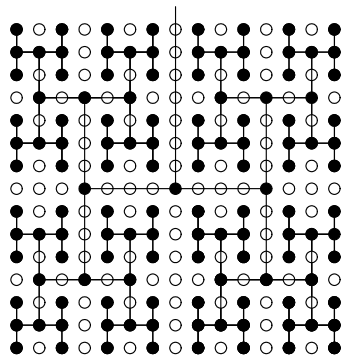


Fig. 4. A traditional H-tree.

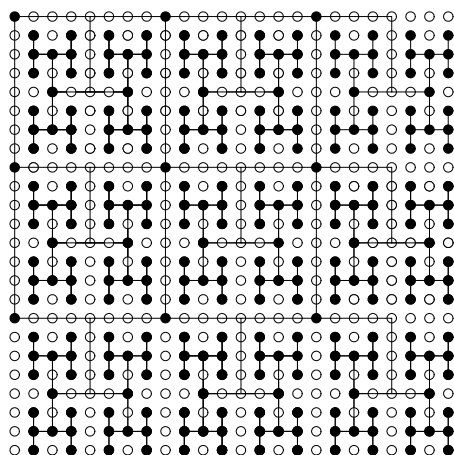


Fig. 5. mesh-connected tree made up of one 3x3 mesh-connected and 5 level binary trees in VLSI array.

The graph arranging method shown in figure 4, is obtained when embedding binary trees in the VLSI array [4,10,17] in the manner of H, which will achieve regularity and extensibility.

The traditional binary tree, because of its extensibility, is frequently used in VLSI layout.

Let N_h^{REC} be the number of nodes in an area just large enough to contain an entire h -level binary tree.

We get
$$N_h^{REC} = \frac{2^h}{\left(2^{\lceil h/2 \rceil} - 1\right) \times \left(2^{\lceil (h+1)/2 \rceil} - 1\right)}$$
. Where

REC is for “rectangular” and h for “level”. Following this equation, we obtain that after the 13th level, the node utilization of H-tree will remain at 50 % and will not change with the increase in the scale of binary tree. In our topology, what we should consider is the coordination between mesh-connection of different size and binary trees. This is because we found that the arranging methods for the odd-number level and even-number level are different, so they deserve separate consideration.

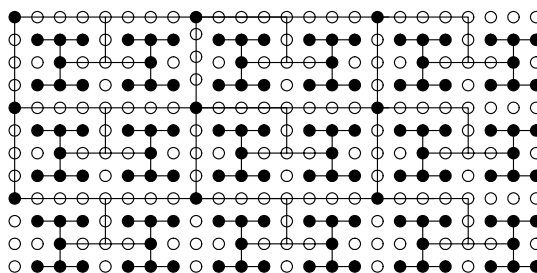


Fig. 6. mesh-connected tree made up of one 3x3 mesh-connected and 4 level binary trees in VLSI array.

As with odd-number level, the number of level must be higher than 3 to be meaningful. Assume h is the number of levels for binary tree and n is the size of the mesh-connection, we can obtain the equation in case of odd-number levels.

$$N_{odd}^{REC}(h, n) = \frac{(2^h - 1) \times n^2}{2^{h-3} \times (4n)^2} \dots\dots\dots(1)$$

When we take h and n to limit ∞ , $(h, n) \rightarrow \infty$, we

get $\lim_{\substack{h \rightarrow \infty \\ n \rightarrow \infty}} N_{odd}^{REC}(h, n) = 0.5$, and so we discover that in

the case of odd-number levels, the scale of our topology will not change with increases in binary trees or mesh-connection. Shown in figure 5 is the binary tree embedding method for odd-number levels. In case of even-number levels, the numbers must be higher than 3 to be meaningful, therefore the result we obtained is

$$N_{even}^{REC}(h, n) = \frac{(2^h - 1) \times n^2}{(2^{h-4}) \times (4n \times 8n)} \dots\dots\dots(2)$$

Then in taking h and n to limit ∞ , $(h, n) \rightarrow \infty$

respectively, we get $\lim_{\substack{h \rightarrow \infty \\ n \rightarrow \infty}} N_{even}^{REC}(h, n) = 0.5$. This

value will not change with increases in the number of levels for binary tree, or increases in the scale of mesh-connection. Shown in figure 6 is the binary tree embedding method for an even-number level.

From equations (1)(2), we can derive a common equation $N_{COM}^{REC}(h, n)$ suitable for odd number and even number levels as follows:

$$N_{COM}^{REC}(h, n) = \frac{(2^h - 1) \times n^2}{2^{h+1} \times n^2} \dots\dots\dots(3)$$

We can obtain a result from the above equation that regardless of odd-number or even-number levels, or the scale of mesh-connection, the overall VLSI layout may not be affected and its available effective nodes are, $N_{COM}^{REC}(h, n) = 0.5$, the same as traditional binary trees.

4. Comparison

In this section, we show table 1 lists these six parameters for some of the commonly used interconnection topologies. From this table indicates that the node utilization is better than mesh of trees, mesh-connected tree and pyramid. Besides, MOT and MCT have serious crossing and poor extensibility for VLSI layout.

Table 1. topologies parameters of selected interconnection networks.

	Degree	Diameter	Bisection	Fault Tolerance	Extensibility	Node Utilization
Meshes	4	$N^{1/2}$	N	OO	OO	100%
Trees	3	$\log N$	1	O	OO	50%
Pyramid	9	$\log_4 N$	$2N$	×	× ×	33%
Mesh of Trees	3	$\log N$	N	OO	O	NA
Mesh-connected Trees	3	$\log N$	N	OO	O	NA
Our design	4	$N^{1/2}$	$N^{1/2}$	OO	OO	50%

*OO: very good, O: good, = : fair, × : poor, × × : very poor.

Conclusion

This paper discusses the embedding of another mesh-connected tree architecture, given by us into VLSI array in a detailed manner. As proved by equation, we can obtain that under mesh-connected architecture and binary trees of different scales, their overall performance will remain unchanged. In addition, better communication capabilities might be obtained from our topology than traditional H-tree and pyramid. This would be without the problem of crossing the wire with traditional mesh of tree and mesh-connected tree. Therefore, it is extremely suitable for VLSI.

References

1. S. Battacharya, S. Kriani, and W. T. Tsai, "Quadtree Interconnection Network Layout," *Proceedings of the Second Great Lakes Symposium on VLSI*, pp. 81-87, 1991.
2. D. A. Carlson, "Modified-Mesh Connected Parallel Computers," *IEEE Trans. Computer*, vol. 37, Issue: 10, pp. 1315-1321, Oct. 1988.

3. K. Efe and A. Fernández, "Mesh-Connected Trees: Bridge Between Grids and Meshes of Trees," *IEEE Trans. Parallel and Distributed Systems*, vol. 7, no. 12, pp. 1281-1291, Dec. 1996.
4. T. Y. Feng, "A Survey of Interconnection Network," *IEEE Computer*, pp. 12-27, Dec. 1981.
5. D. Gordon, I. Koren, and G. M. Silberman, "Embedding Tree Structure in VLSI Hexagonal Arrays," *IEEE Trans. Computers*, vol. C-33, no. 1, pp. 104-107, Jan. 1984.
6. D. Gordon, "Efficient Embeddings of Binary Tree in VLSI Arrays," *IEEE Trans. Computers*, vol. C-36, no. 9, pp. 1009- 1018, Sept. 1987.
7. C. T. Ho and L. Stockmeyer, "A New Approach to Fault-Tolerant Wormhole Routing for Mesh-Connected Parallel Computers," *IEEE Trans. Computers*, vol. 53, Issue: 4, pp. 423-438, Apr. 2004.
8. K. Hwang and F. A. Briggs, *Computer Architecture and Parallel Processing*. McGraw-Hill, 1984.
9. G. E. Jan, S. W. Leu, and C. H. Li, "On the Array Embeddings and Layouts of Quadrees and Pyramids," *Journal of Information Science and Engineering*, vol. 20, no. 1, pp. 127-141, Jan. 2004.
10. S. Y. Kung, *VLSI Array Processors*. Prentice-Hall, 1988.
11. F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann, 1992.
12. C. Mead and L. Conway, *Introduction to VLSI System*. Addison-Wesley, 1980.
13. R. Miller and Q. F. Stout, *Parallel Algorithms for Regular Architectures: Meshes and Pyramids*. MIT Press, 1996.
14. S. K. Nandy and I. V. Ramakrishnan, "Dual Quadtree Representation for VLSI Design," *Proceedings of the 23th Design Automation Conference*, pp. 663-666, 1986.
15. H. Samet, "The Quadtree and Related Hierarchical Data Structures," *ACM Computing Survey*, vol. 16, pp. 187-260, Jun. 1984.
16. R. Suaya and G. M. Birtwistle, *VLSI and Parallel Computation*. Morgan Kaufmann, 1990.
17. L. Synder, "Introduction to the Configurable Highly Parallel Computer," *IEEE Computer*, pp. 47-64, Jan. 1982.
18. J. Wu, "Fault-Tolerant Adaptive and Minimal Routing in Mesh-Connected Multicomputers using extended Safety Levels," *IEEE Trans. Parallel and Distributed Systems*, vol. 11, no. 2, pp. 149159, Feb. 2000.