# On the Array Embeddings and Layouts of Quadtrees and Pyramids*

GENE EU JAN, SHAO-WEI LEU+ AND CHENG-HUNG LI++
*Department of Computer Science*
*+Department of Electrical Engineering*
*National Taiwan Ocean University*
*Keelung, 202 Taiwan*
*E-mail: {b0199, +b0119}@mail.ntou.edu.tw*
*++Department of Electronic Engineering*
*National Taiwan University of Science and Technology*
*Taipei, 106 Taiwan*
*E-mail: D9102102@mail.ntust.edu.tw*

Quadtree and pyramid structures have attracted considerable attention in recent years. They are increasingly being applied to the fields of digital image and signal processing. As a result, the efficient embedding of these structures in VLSI arrays has become an important research topic. In this paper, we propose three schemes to embed either quadtrees or pyramids in a rectangular, hexagonal, or octagonal mesh, respectively, with three different node shapes for VLSI layout. Our analyses show that the best achievable node utilization is 67% when embedding either structure in an octagonal mesh. This result outperforms the best utilization recorded in literature by 25%. Our study also indicates that the octagonal node gives the most favorable compromise between high area utilization desired and routing space required among the processing nodes.

***Keywords:*** quadtree, pyramid, embedding, mesh, VLSI layout

## 1. INTRODUCTION

Quadtree and pyramid structures [1, 5, 6, 9, 13] have been used extensively to represent two-dimensional data in applications such as image processing and geographic information systems. Despite their structural simplicity and regularity, two-dimensional quadtrees and three-dimensional pyramids are difficult to implement on VLSI due to layout complexity. It is also known that, when building a multiprocessor on a VLSI chip, communication is more expensive than computation. Nevertheless, each processing element in tree-like structures has limited connectivity, which means that the impact of the communication channels on chip area will also be limited. To take advantage of this important property, suitable placement and mapping strategies need to be devised.

The embeddings of binary trees have been studied extensively [4, 5, 10, 16]. It has been found that when embedding a binary tree in a hexagonal or octagonal mesh topol-

Received January 31, 2003; accepted July 4, 2003.
Communicated by Shiuh-Pyng Shieh.
* A preliminary version of this paper was presented at the 2002 International Computer Symposium, 2002.

ogy the highest achievable area utilization ratio is 93%. Beyond the binary tree embedding, Bhattacharya developed in 1991 a method to map a quadtree on a rectangular mesh [1]. Unfortunately, the proposed method incurs too many crossings and is difficult to scale. As to the embedding of pyramids in two-dimensional structures, none have appeared in the literature. On the basis of Bhattacharya's dotted triangle, we have developed a set of methods to embed either quadtrees or pyramids in rectangular, hexagonal, or octagonal meshes with remarkable node and area utilization ratios.

The rest of this paper is organized as follows. In section 2, we first review and analyze previous works on embedding, then we introduce our methods and compares the node utilization ratios. Section 3 discusses in detail how the shapes of the nodes affect the area utilization ratio. Section 4 concludes the paper.

## 2. TWO-DIMENSIONAL EMBEDDINGS OF PYRAMIDS AND QUADTREES

Placement and routing greatly influence the efficiency and cost of a VLSI circuit. Hence, it is highly desirable that an embedding method is capable of not only minimizing wire crossings, but also of maximizing the utilization of available VLSI real estate for building circuits, while retaining enough space among the processing nodes for routing.

### 2.1 Pyramid Architecture

The pyramid is a well-know parallel network in fields such as image processing and pattern recognition. It is a highly efficient interconnection structure and has extensive applications [2, 6, 8, 10, 13]. Fig. 1 shows a three-level pyramid network. The following is a brief introduction to the pyramid architecture, which is adopted mainly from [12].

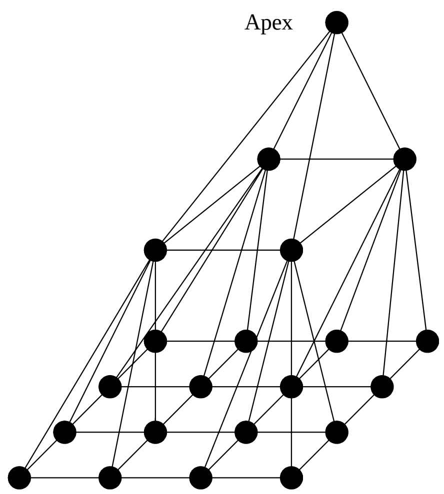



Fig. 1. A three-level pyramid.

A pyramid of size $n$ is a structure that can be regarded as a rooted 4-ary tree of height $\log_4 n$, with additional horizontal links that make each horizontal level a mesh. It is often convenient to view the pyramid as a structure that has a mesh of size $n$ at its base and contains a total of $(4n - 1)/3$ processors. A processor at level $k$ is connected to nine neighbors, if they exist, including 4 siblings at level $k$, 4 children at level $k + 1$, and a parent at level $k - 1$. The pyramid structure combines the advantages of a two-dimesional mesh and a quadtree. The advantage of the pyramid over the mesh is that the communication diameter of a pyramid network of size $n$ is only $\Theta(\log n)$. This is true since any two processors in the pyramid can exchange information through the apex. The advantage of the pyramid over the quadtree can be seen from the fact that the distance between neighboring nodes at the same level is definitely shorter due to direct links between them.

### 2.2 Embedding Method

Fig. 2 shows a quadtree and a pyramid, both two-level, being mapped as a two-dimensional mesh. We place the upper-level node in the center to avoid line crossings and the lower-level nodes around the center to reduce the total area. Fig. 3 shows a three-level quadtree and pyramid being embedded in a two-dimensional mesh. This method can be used to embed structures with more levels. Note that wire crossings only appear when embedding the pyramid. For further observation, Fig. 4 shows a mesh embedding of a four-level quadtree.

(a) (b)

Fig. 2. Two-dimensional view of a two-level (a) quadtree and (b) pyramid.

(a) (b)

Fig. 3. A three-level (a) quadtree and (b) pyramid in a 2-D mesh.

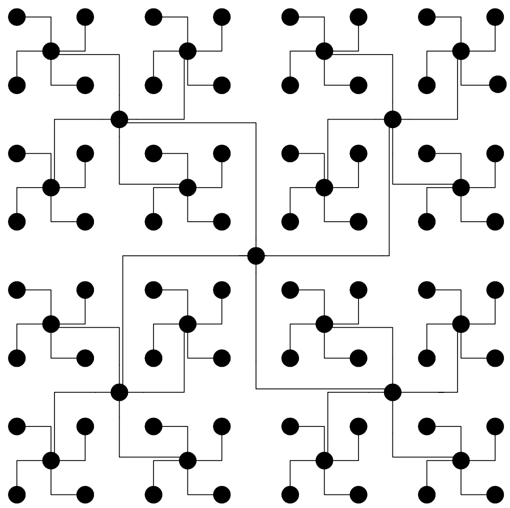


Fig. 4. A four-level quadtree.

Fig. 5 shows a pyramid structure being embedded in a two-dimensional mesh. It shows that up to three links have to be combined in order to save routing space. At first glance, this seems to be a serious drawback. However, our approach guarantees that no more than three links will be placed together in any particular direction regardless of the number of levels.

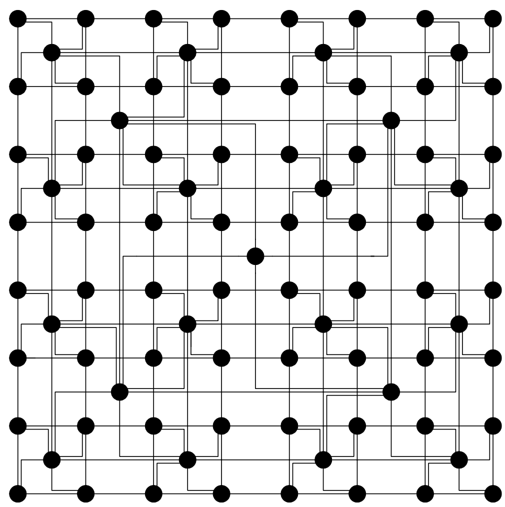
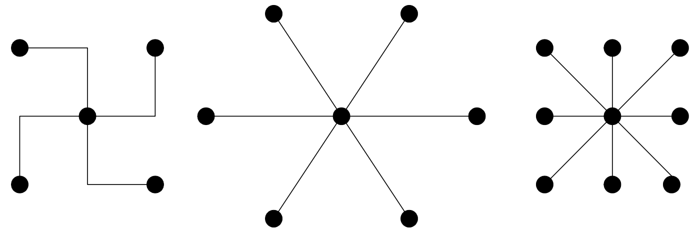


Fig. 5. A four-level pyramid.

### 2.3 Analyses of Node Utilization after Embedding

In this section, we introduce our methods to embed pyramids and quadtrees in two-dimensional meshes of three distinct topologies, namely, rectangular, hexagonal, and octagonal meshes, as shown in Fig. 6. We will thoroughly analyze and compare the node utilization ratios from these configurations.



(a) (b) (c)

Fig. 6. Three mesh topologies: (a) rectangular (b) hexagonal (c) octagonal.

### 2.3.1 The rectangular mesh

Fig. 7 shows a quadtree and a pyramid, both three-level, being embedded in a rectangular mesh. Although the connections are different, both structures occupy the same nodes in the mesh. Let $N_k$ be the total number of nodes in a quadtree or pyramid, then

$$N_k = \frac{4^k - 1}{3} \tag{1}$$

(a) (b)

Fig. 7. A three-level (a) quadtree and (b) pyramid embedded in a rectangular mesh.

Since the number of nodes increases as the quadtree or pyramid grows, the area enclosing all the occupied nodes will gradually extend in the mesh. Let $M_k^{\text{REC}}$ be the total number of nodes in an area just large enough to contain an entire *k*-level quadtree or pyramid. We get

$$M_k^{\text{REC}} = (2^k - 1)^2 \tag{2}$$

From equations (1) and (2), we obtain the ratio of node utilization for a *k*-level quadtree or pyramid as

$$r_k^{\mathrm{REC}} = \frac{N_k}{M_k^{\mathrm{REC}}} = \frac{(4^k - 1)/3}{(2^k - 1)^2} \tag{3}$$

where REC is for "rectangular" and $k$ for "levels". Numerical substitutions give the plot in Fig. 8. For structures of seven levels or more, the ratio remains at about 33%, meaning that, out of all the nodes within the structure's footprint, 67% go unused. Obviously, a ratio of 33% is too low to be of any practical value.

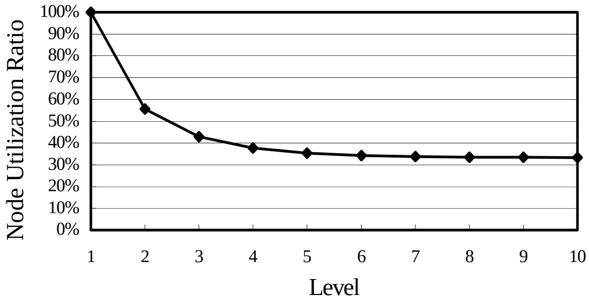


Fig. 8. Node utilization ratios from embedding in a rectangular mesh.

### 2.3.2 The hexagonal mesh

This section studies node utilization from embedding quadtrees in a hexagonal mesh. Embedding in a hexagonal mesh was first suggested by Gordon [4, 5], who embedded a binary tree in a hexagonal mesh and obtained a 93% ratio of node utilization. In the current study, we use a no-crossing approach to embed a quadtree in a hexagonal mesh, as shown in Fig. 9. It can be shown that it needs $(4 \times 3) + (5 \times 3)$ nodes to form the footprint of a three-level quadtree in the mesh and $(11 \times 7) + (11 \times 6)$ nodes for a four-level one. Nodes required for the fifth level and up can be obtained by

$$M_k^{\mathrm{HEX}} = (13 \times \sum_{i=5}^{k} 2^{i-5} + 11)(7 \times 2^{i-3} - 1) \tag{4}$$

where HEX stands for hexagonal mesh. Putting equations (1) and (4) together, the node utilization ratio is

$$r_k^{\mathrm{HEX}} = \frac{N_k}{M_k^{\mathrm{HEX}}} = \frac{(4^k - 1)/3}{(13 \times \sum_{i=5}^{k} 2^{i-5} + 11)(7 \times 2^{i-3} - 1)} \tag{5}$$

Fig. 10 shows the node utilization ratios from embedding quadtrees up to ten levels in a hexagonal mesh. The ratio remains at about 47% for levels of eight or more. For most practical purposes, this ratio is still insufficient, even though this embedding scheme excels in producing no wire crossings at all. The hexagonal mesh is also unsuitable for embedding the pyramid since it generates too many crossings.

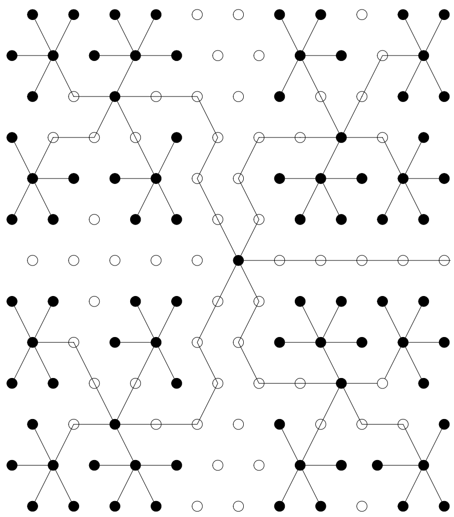


Fig. 9. A no-crossing approach to embedding a quadtree in a hexagonal mesh.

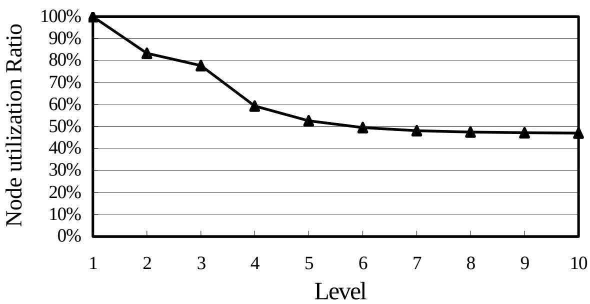



Fig. 10. Node utilization ratios for embedding quadtrees in a hexagonal mesh.

### 2.3.3 The octagonal mesh

Our third approach to embedding quadtrees and pyramids is based on an octagonal mesh topology [16]. Figs. 11 and 12 show a four-level quadtree and pyramid embedded in an octagonal mesh. The ratio of node utilization can be derived in a similar way. The total number of nodes residing in the area covered by an embedded quadtree or pyramid is given by

$$M_k^{\mathrm{OCT}} = (2^{k-1})^2 + (2^{k-1}-1)^2 \qquad (6)$$

Combining equations (1) and (6), the ratio of node utilization is

$$r_k^{\mathrm{OCT}} = \frac{N_k}{M_k^{\mathrm{OCT}}} = \frac{(4^k-1)/3}{(2^{k-1})^2 + (2^{k-1}-1)^2} \qquad (7)$$

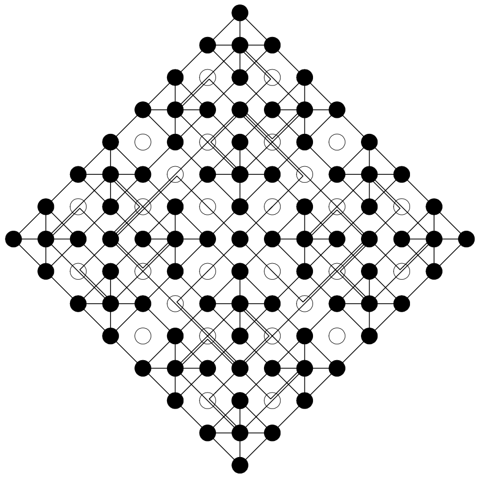


Fig. 11. A four-level quadtree embedded in an octagonal mesh.

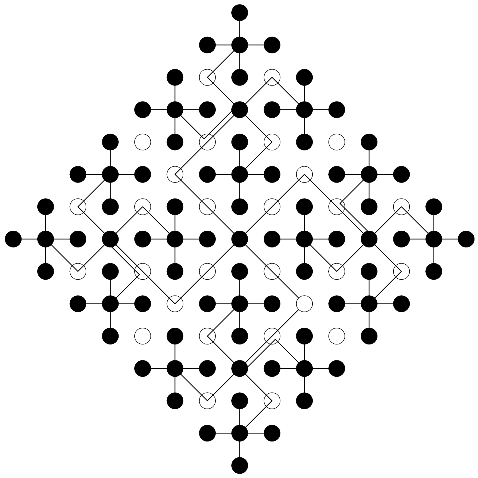


Fig. 12. A four-level pyramid embedded in an octagonal mesh.

Fig. 13 shows a plot for quadtrees/pyramids of up to ten levels. It is easy to derive that, regardless of how tall a quadtree/pyramid grows, the node utilization ratio stays above 67%, which is the best result among the embedding schemes we have proposed.

### 2.3.4 Overall comparison

Table 1 compares the node utilization ratios of all the embedding schemes we have proposed, along with the results derived from Battacharya's method. Obviously, the embedding scheme based on the octagonal mesh outperforms the others. One less obvious, but certainly not less important, feature of our methods is that they completely avoid the crossing problem which Battacharya's approach suffers from. This feature is not to be overlooked since wire crossings increase the cost and complexity of VLSI layout.

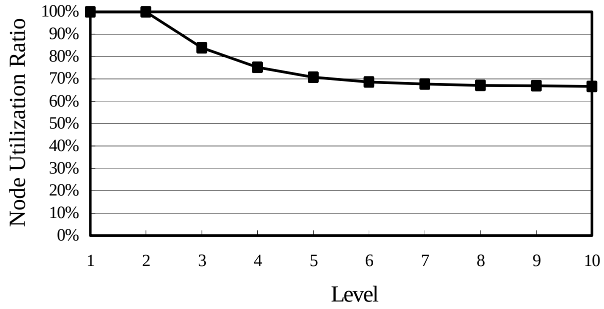



Fig. 13. Node utilization ratios from embedding quadtrees or pyramids in an octagonal mesh.

**Table 1. Node utilization ratios from four embedding schemes.**

| Level \ Shape | Rectangular | Hexagonal | Bhattacharya | Octagonal |
|---|---|---|---|---|
| 1 | 100% | 100% | 100% | 100% |
| 2 | 55.5% | 83.3% | 83.3% | 100% |
| 3 | 42.8% | 77.7% | 84% | 84% |
| 4 | 37.7% | 59.4% | 77.2% | 75.2% |
| 5 | 35.4% | 52.6% | 67.3% | 70.8% |
| 6 | 34.3% | 49.6% | 63.1% | 68.7% |
| 7 | 33.8% | 48.2% | 61.1% | 67.7% |
| 8 | 33.5% | 47.5% | 60.1% | 67.1% |
| 9 | 33.4% | 47.2% | 59.7% | 66.9% |
| 10 | 33.3% | 47.0% | 59.4% | 66.7% |

## 3. AREA UTILIZATION

Our earlier effort on finding the best method to fit a tree-like structure into a two-dimensional mesh was somewhat idealized for not considering the effect of node shapes. In VLSI layout, the shapes of circuit blocks do affect the total area needed. To form a proper basis for analysis, we now assume three popular polygon shapes for the nodes in the following analyses on the area utilization ratio.

### 3.1 Square Nodes

Fig. 14 shows the layouts of embedded quadtrees/pyramids for up to three levels using square nodes. Assuming unit area for each node and by applying equation (1), the area utilization ratio of an embedded quadtree in a rectangular mesh is given by

$$\alpha_k^{\text{REC}} = \frac{(4^k - 1)/3}{(2^k - 1)^2} \tag{8}$$

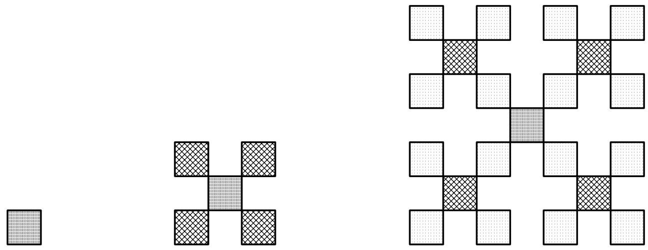


Fig. 14. Layouts using square nodes.

A plot of the above equation is given in Fig. 15. We can see that the area utilization is no more than 33% for quadtrees of seven levels or more. Obviously, this ratio is not acceptable.

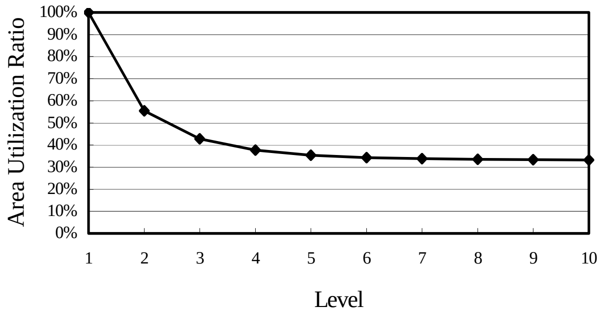



Fig. 15. Area utilization ratios using square nodes.

### 3.2 Hexagonal Nodes

Fig. 16 shows example layouts using hexagonal nodes in a hexagonal mesh. The area utilization ratio is given by

$$\alpha_k^{\mathrm{HEX}} = 2^{k-2}(5+6\sum_{k=3}^{5} 2^{k-3}) \tag{9}$$

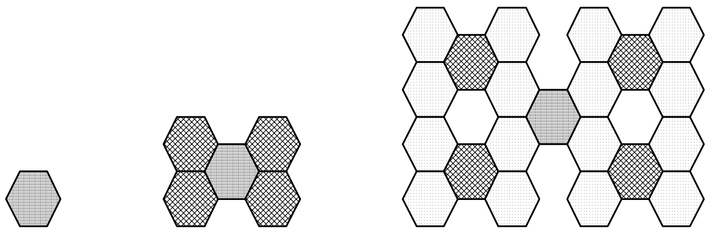


Fig. 16. Layouts using hexagonal nodes.

A plot is shown in Fig. 17, which shows that the area utilization stays above 66%, substantially higher than the previous value. However, this configuration is highly prone to lack of routing space between nodes, and thus may not always be economical for VLSI layout.

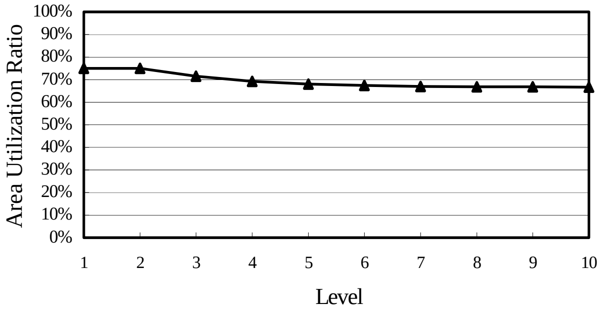



Fig. 17. Area utilization ratios using hexagonal nodes.

### 3.3 Octagonal Nodes

Example layouts using octagonal nodes appear in Fig. 18. The area utilization for an embedded *k*-level quadtree is given by

$$\alpha_k^{\mathrm{OCT}} = \frac{7(4^k - 1)/3}{(\sum_{i=0}^{k} 2^i)^2} \tag{10}$$

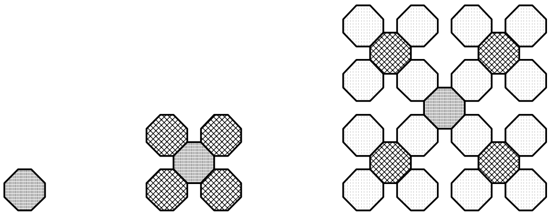


Fig. 18. Layouts using octagonal nodes.

Simple manipulation of the equation reveals that, as $k$ grows larger, the ratio asymptotically approaches $0.58\bar{3}$, which is also confirmed by the plot in Fig. 19.

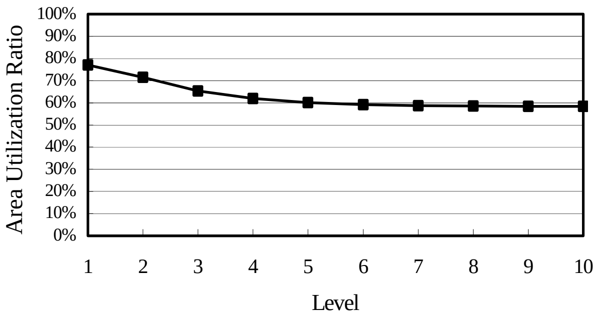



Fig. 19. Area utilization ratios using octagonal nodes.

### 3.4 Overall Comparison

Table 2 lists the area utilization ratios resulting from embeddings discussed in the previous sections using three different node shapes. A few observations are as follows:

1. Square nodes leave too much space unused, and may not be suitable for any practical purpose.
2. Hexagonal nodes make very efficient use of the VLSI space, but may not provide enough space for routing between some nodes. Its applicability depends heavily on the communication requirements of the topology.
3. Octagonal nodes exhibit the best aspects of the two other varieties and may therefore be the best choice for most applications.

**Table 2. Area utilization ratios with three node shapes**.

| Shape \ Level | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Rectangular | 100% | 55% | 42% | 37% | 35% | 34% | 33% | 33% | 33% | 33% |
| Hexagonal | 75% | 75% | 71% | 69% | 68% | 67% | 67% | 66% | 66% | 66% |
| Octagonal | 77% | 71% | 65% | 61% | 60% | 59% | 58% | 58% | 58% | 58% |

Figs. 20 and 21 are examples of a 4-level (8 × 8) quadtree and pyramid network, respectively. Note that all the octagonal nodes in the figures have been rotated by 45 degrees for better alignment of the drawings. No change is made to the embedding method.

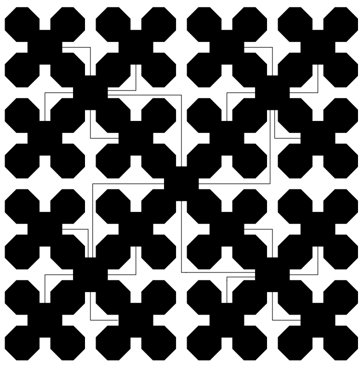


Fig. 20. The quadtree network for octagonal nodes.

## 4. CONCLUSIONS

In this paper, we have proposed three schemes to embed quadtrees or pyramids in a two-dimensional mesh. Each embedding method uses a distinct mesh topology. For each proposed method, we analytically determined its node utilization ratio. Our analyses

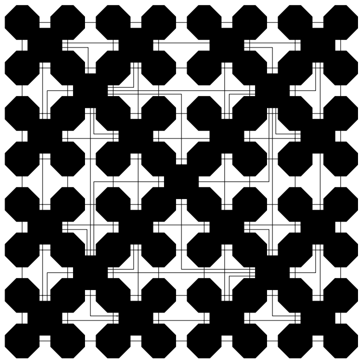


Fig. 21. The pyramid network for octagonal nodes.

show that the octagonal mesh topology, besides being free from any crossing problem, outperforms all other known approaches in terms of node utilization. Our study should provide useful indicators for designers who intend to implement quadtrees and/or pyramids as VLSI arrays. Since a large number of processing nodes are likely to be deployed in such arrays, the shapes of the nodes will have a major impact on layout economics. We use three different polygons to demonstrate how the shape of the nodes affects the ratio of area utilization. Our study shows that nodes of octagonal shape provide a good compromise between area utilization and the space needed for routing.

## REFERENCES

1. S. Battacharya, S. Kriani, and W. T. Tsai, "Quadtree interconnection network layout," in *Proceedings of the 2nd Great Lakes Symposium on VLSI*, 1991, pp. 81-87.
2. B. Codenotti and M. Leoncini, *Introduction to Parallel Processing*, Addison-Wesley, 1993.
3. T. Y. Feng, "A survey of interconnection network," *IEEE Computer*, Vol. 14, 1981, pp. 12-27.
4. D. Gordon, I. Koren, and G. M. Silberman, "Embedding tree structure in VLSI hexagonal arrays," *IEEE Transactions on Computers*, Vol. C-33, 1984, pp. 104-107.
5. D. Gordon, "Efficient embeddings of binary tree in VLSI arrays," *IEEE Transactions on Computers*, Vol. C-36, 1987, pp. 1009-1018.
6. C. T. Ho and S. L. Johnsson, "Dilation *d* embedding of a hyper-pyramid into a hypercube," in *Proceedings of the Supercomputing Conference*, 1989, pp. 294-303.
7. K. Hwang and F. A. Briggs, *Computer Architecture and Parallel Processing*, McGraw- Hill, 1984.
8. V. K. P. Kumar and D. Reisis, "Pyramids versus enhanced arrays for parallel image processing," Technical Report CRI-86-16, Department of Electrical Engineering, University of Southern California, 1986.
9. S. Y. Kung, *VLSI Array Processors*, Prentice-Hall, 1988.
10. F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, 1992.

11. C. Mead and L. Conway, *Introduction to VLSI System*, Addison-Wesley, 1980.
12. R. Miller and Q. F. Stout, *Parallel Algorithms for Regular Architectures: Meshes and Pyramids*, MIT Press, 1996.
13. S. K. Nandy and I. V. Ramakrishnan, "Dual quadtree representation for VLSI design," in *Proceedings of the 23rd Design Automation Conference*, 1986, pp. 663-666.
14. R. Suaya and G. M. Birtwustle, *VLSI and Parallel Computation*, Morgan Kaufmann, 1990.
15. H. Samet, "The quadtree and related hierarchical data structures," *ACM Computing Survey*, Vol. 16, 1984, pp. 187-260.
16. L. Synder, "Introduction to the configurable highly parallel computer," *IEEE Computer*, Vol. 14, 1982, pp. 47-64.

**Gene Eu Jan (詹景裕)** received a B.S. degree in Electrical Engineering from the National Taiwan University in 1982, and an M.S. and a Ph.D. in Electrical Engineering from the University of Maryland, College Park, in 1988 and 1992, respectively. He has been an associate professor with the Departments of Computer Science, and Navigation, National Taiwan Ocean University, Keelung, Taiwan, since 1993. Prior to joining the National Taiwan Ocean University, he was a Visiting Assistant Professor in the Department of Electrical and Computer Engineering at the California State University, Fresno, California. His research interests include parallel computer systems, interconnection networks, and VLSI and design automation.

**Shao-Wei Leu (呂紹偉)** received the B.S. degree in Biomedical Engineering from Chung-Yuan Christian University, Chungli, Taiwan, in 1979, the M.S. degree in Computer Engineering from Clemson University, Clemson, South Carolina, in 1985, and the Ph.D. degree from Florida Atlantic University, Boca Raton, Florida, in 1990. He has been an associate professor in the Department of Electrical Engineering at National Taiwan Ocean University, Keelung, Taiwan, since 1990. His research interests include VLSI system design and verification, parallel processing, and network computing. He is a member of IEEE Computer Society.

**Cheng-Hung Li (李政宏)** was born on August 14, 1974 in Taipei, Taiwan. He received the B.S. degree in Communication Engineering and Computer Technology from Da-Yeh University in 2000 and the M.S. degree in Electrical Engineering from National Taiwan Ocean University in 2002. Currently, he is a Ph.D. student in the Department of Electronics Engineering at National Taiwan University of Science and Technology. His research interests are in the areas of parallel processing, interconnection networks, and computer architecture. He is a student member of the ACM and IEEE Computer Society.