# Efficient Embeddings of Quadtrees and Pyramids in VLSI Arrays

詹景裕* 呂紹偉** 李政宏

國立台灣海洋大學 *資訊科學系 電機工程學系
*e-mail:b0199@mail.ntou.edu.tw
**e-mail:b0119@mail.ntou.edu.tw
e-mail:m89530064@mail.ntou.edu.tw

## 摘 要

本文針對四元樹及金字塔型結構的二維嵌入問題提出一套基於二維網格的嵌入方法，使得四元樹和金字塔被轉換為平面結構之後能夠獲得最大可能的網格節點利用率，我們除詳述嵌入之方法外，並深入推導和比較在不同網格結構上所能獲得的節點使用率。我們所提出的方法可以在八邊形網格上嵌入四元樹或金字塔結構，並且能在四元樹或金字塔層數無限增加之狀況下保持不變的節點使用率。此方法將可運用於超大型積體電路陣列之佈局設計。

關鍵字：二維嵌入（2D Embedding）、八邊形網格（Octagonal-connected Mesh）、金字塔（Pyramid）、四元樹（Quadtree）、超大型積體電路佈局（VLSI Layout）。

## 一、前言

近年來由於影像及圖形處理的大量需求，四元樹及金字塔架構被廣泛應用在相關領域的研究上。由於四元樹或金字塔原本屬於三維立體架構，並不適合運用在 VLSI 的佈局上，所以如何將三維四元樹或金字塔轉換成二維的平面設計是一項重要的技術。我們的研究目標在於將金字塔轉成二維平面結構後並作適當的排列，以求降低連線的成本，並且提高系統整體的效能。經轉換得到的二維的平面結構十分適合應用在晶片的設計上，因為所有節點經過適當的排列後，可以使用較小的晶片面積，也可以減少節點之間連線交叉的機會，降低線路的複雜性，同時也可以降低設計與測試的成本。

以 VLSI 實現多處理器陣列時，樹狀結構是一個相當吸引人的選擇，因其連結的架構簡單且具有規律性。以 VLSI 的成本考量，計算單元的成本相對低於連線成本，因此若每一個處理單元只與緊鄰的節點（Immediate Neighbor）維持樹狀結構，將可以降低連線的成本。

在影像處理的領域中，金字塔網路是一種效率極佳，應用甚廣的網路型態。圖 1 是一個三層的金字塔型網路，除了最上層與最下層之外，每一節點皆與其四個子節點直接連接，而每一層各自形成一二維網格（Mesh），由上而下，依序為第 1 層，第 2 層…，則第 $n$ 層具有 $4^{n-1}$ 個節點。以橫切面來看，每一層都由四元樹結構[11]組成；亦即，金字塔結構可視為四元樹與二維網格的組合。這種結在影像處

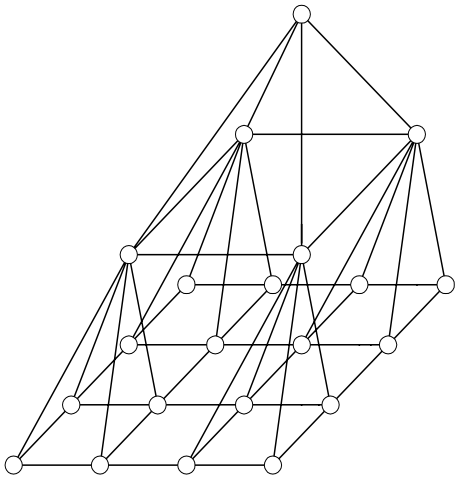

圖1 一個3-層的金字塔型網路

理中被大量使用。一個規模為 $n$ 層的金字塔總共有 $(4^n-1)/3$ 個節點，圖 1 為一個 $n$=3 的金字塔結構，也稱二維金字塔結構，它的退化形式為一維金字塔結構，又稱為 X-Tree[4][6][9]。

金字塔型網路結構兼具二維網格與四元樹結構的優點，原因如下：第一、它含有二維網格，使得同一層節點之間比四元樹有更快的傳輸速率和較高的容錯能力。第二、它也包含四元樹的架構，可以獲得較二元樹為優之平面傳輸，與更為廣泛的應用場合。在傳遞訊息時，無論是上下層的親子（Parent-Children）之間，或是同一層前後左右的鄰居（Neighbor）之間，金字塔型網路都比單純的二維網格或四元樹結構有更好的傳輸效率。

在 80 年代初期，二元樹曾被廣泛深入的探討，相關的研究也很多[2] [3] [4] [5] [8] [9] [11] ，除了 H-Tree 之外，將二元樹嵌入六邊形或是八邊形平面結構，都可以獲得 93% 的有效面積利用率[2][3]。90 年代初期，Bhattacharya[1]將四元樹嵌入四邊形網格，以這種嵌入方法會有跨線（Crossing）的問題存在，並且擴展性也很差，嵌入後的使用率也未臻理想。而金字塔結構至今尚無任何文獻提出其二維嵌入的排列方法，所以本論文將以 Bhattacharya 所提出的 Dotted Triangle 為基礎，提出一種新的二維嵌入方式，將四元樹或

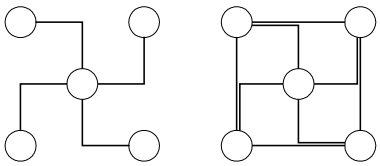

圖2 將2-層的四元樹和金字塔轉換成二維平面圖

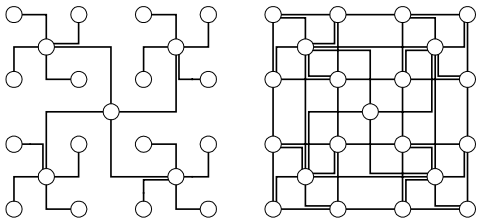

圖3 將3-層的四元樹和金字塔轉換成二維平面圖

金字塔在最有效率的排列方式下嵌入四邊形、六邊形或八邊形的平面結構中。

## 二、嵌入方式

晶片的佈線方式對 VLSI 整體的效能與成本有重要的影響，若能減少線路的交叉，並藉由排列的方式減少晶片面積的浪費，將明顯有助於提昇效能與降低成本。此外，在尋求對晶片面積作最有效利用的同時，也必須考慮在節點之間保留足夠的空間，以作為佈線之用。

圖 2 為兩層的四元樹結構和金字塔型結構轉成二維結構之平面圖，為了減少線路之間的交叉點，將上層的節點置於圖形中央而將下層的節點置於圖形的四周，亦即正方形的四個頂點，用以縮小整體面積。圖 3 是一個三層的四元樹和金字塔的二維結構之平面圖。以此原則來轉換更多層的四元樹和金字塔型結構，雖然在金字塔上無法避免連線交叉的產生，但是就整體來說，仍在可接受的成本範圍內。圖 4 是一個四層的四元樹結構轉換成二維平面佈局後的圖形。由圖上可知，四元樹的每個節點之間並沒有互相連接，也沒有交叉點的產生。所以在最多合併兩條線的情況下，可大幅節省佈線的成本。圖 5 是金字塔的結構轉換成二維

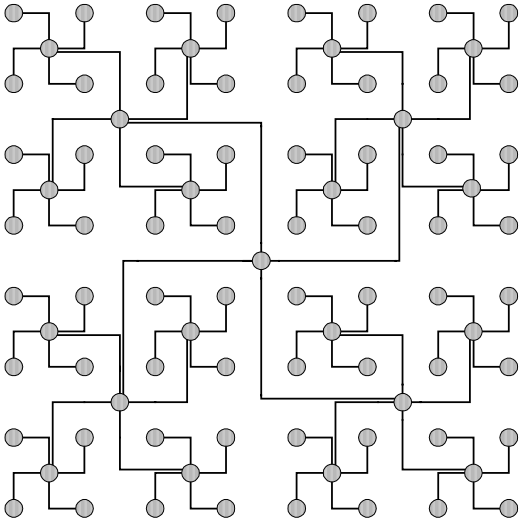

圖4 將4層的四元樹網路轉換2D平面

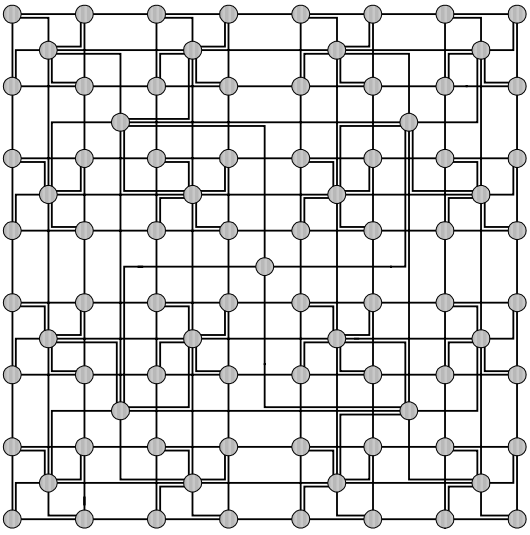

圖5 將4-層的金字塔網路轉成二維平面

網格後的圖形。由圖形上可以看出每個節點在同一方向上最多可合併 3 條線，值得注意的是，此數字並不隨金字塔的擴展而增加。

在下一節我們將以三種不同的網格形式為基礎，將四元樹及金字塔結構嵌入二維平面，並且深入分析比較各種嵌入方式的節點使用率。此三種網格如圖 6 所示，分別為（a）四邊形網格(Rectangular-connected Mesh)、(b)六邊形網格(Hexagonal-connected Mesh)和(c)八邊形網格（Octagonal-connected Mesh)。

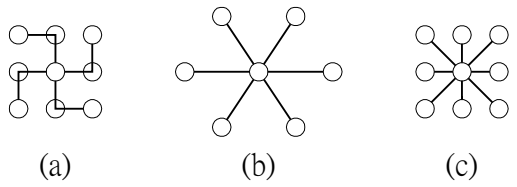


圖6 三種不同的排列相位

## 三、二維嵌入及結果之分析與比較

本節針對三種不同網格進行四元樹及金字塔結構之二維嵌入，並分析與比較節點使用率。

1. 四邊形網格

在四邊形網格裡面，將金字塔和四元樹分別的嵌入到所排列的網格上，排列情形分別如圖 7 所示。我們將四元樹及金字塔分別嵌入四邊形網格中。由圖中可知這兩個圖形只是嵌入的方式不一樣，但所佔用的節點是一樣的。設 $n$ 為四元樹或金字塔的階層數（$n\geqq 2$），且令 $n$ 層的四元樹或金字塔的總節點數為 $N_n$，其通式如下：

$$N_n = \frac{4^n - 1}{3} \quad \cdots\cdots(1)$$

由於嵌入二維網格的節點數將隨著四元樹或金字塔的規模成長而增加，因而在網格上所佔用的區域也將逐漸擴大。我們以 $M_n^{\mathrm{R}}$ 表示能容納 $n$-層四元樹或金字塔的最小四邊形網格區塊內的節點總數，則：

$$M_n^{\mathrm{R}} = \left(2^n - 1\right)^2 \quad \cdots\cdots(2)$$

由式（1）及式（2）我們可以得到 $n$-層四元樹或金字塔的節點佔用比率為

$$r_n^{\mathrm{R}} = \frac{N_n}{M_n^{\mathrm{R}}} = \frac{\left(4^n - 1\right)/3}{\left(2^n - 1\right)^2} \quad \cdots\cdots(3)$$

其中 R 代表四邊形網格，$n$ 為層數，以數值代

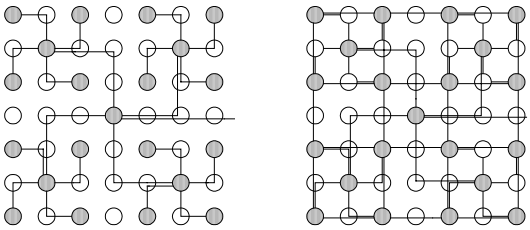

圖7 一個3-層的四元樹和金字塔嵌入在二維網格上

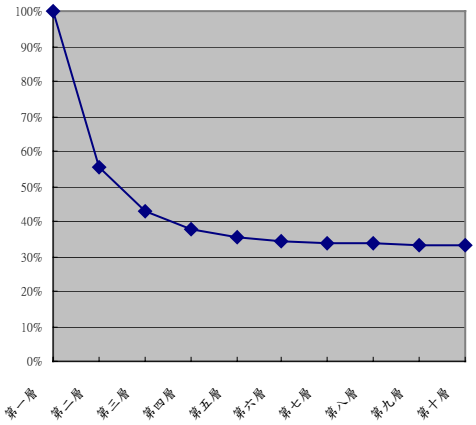


圖8 四元樹的節點佔用率（至十層）

入公式可得到圖 8 的曲線。觀察曲線可知在四元樹成長至第七層後，節點利用率大約維持在 33%，意即有約 67%的面積被浪費。由此可知，利用四邊形網格為基礎嵌入四元樹或金字塔所得到的節點佔用率相當低。我們以同樣的過程可推導出以 Bhattacharya[1]的方法作的嵌入大約有 59%的利用率，但仍偏低。

2. 六邊形網格

本節進一步分析以六邊形網格為基礎，將四元樹或金字塔嵌入後所能獲得的節點使用率。

六邊形網格的使用首由 Gordan[2][3]在 1982 年將二元樹以幾近 93%的比率嵌入六邊形網格。我們以類似的觀念將四元樹以無跨線之方式嵌入六邊形網格，如圖 9 所示。由圖可觀察得知，3 層四元樹時所涵蓋的六邊形網格節點數為（4x3）＋（5x3），4 層所涵蓋的節點數則為（11x7）＋（11x6）。四元樹的層數

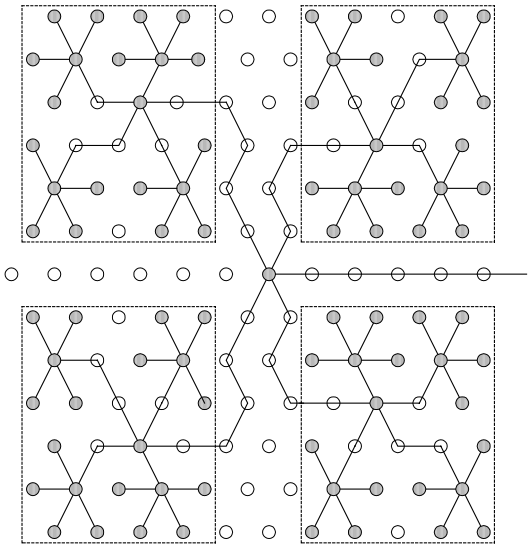

圖9 一個四層無跨線的四元樹

為 5 以上時，所涵蓋的六邊形網格區塊內的節點總數可由以下通式表示：

$$M_n^{\mathrm{H}} = \left(13\times\sum_{k=5}^{n}2^{k-5}+11\right)\left(7\times 2^{k-3}-1\right) \quad \text{.....(4)}$$

其中 H 表示六邊形網格，$n$ 為四元樹層數。由上是即是（1）可求得節點佔用比例為：

$$r_n^{\mathrm{H}} = \frac{N_n}{M_n^{\mathrm{H}}} = \frac{\left(4^n-1\right)/3}{\left(13\times\sum_{k=5}^{n}2^{k-5}+11\right)\left(7\times 2^{k-3}-1\right)} \quad \text{.....(5)}$$

圖 10 為 $n \le 10$ 的節點佔用曲線。由圖可知四元樹層數為 8 之後，節點佔用率可維持在 47%，可見這種嵌入的方式雖然沒有跨線的問題，但是佔用率仍然太低。此外，我們也發現金字塔結構不適合在六邊形網格上嵌入，原因是會產生大量連線重疊以及同層節點無法連線的問題。

3. 八邊形網格

1982 年 Synder[12]也曾經提出八邊形的構想，我們以相同的觀念來設計我們所提出的嵌入法則。圖 11 和圖 12，分別為四層的四元

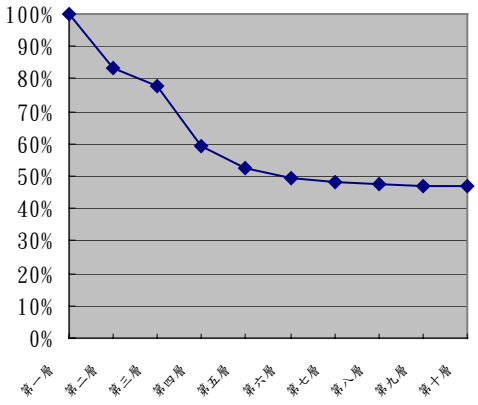


圖10 四元樹的節點佔用率（至十層）

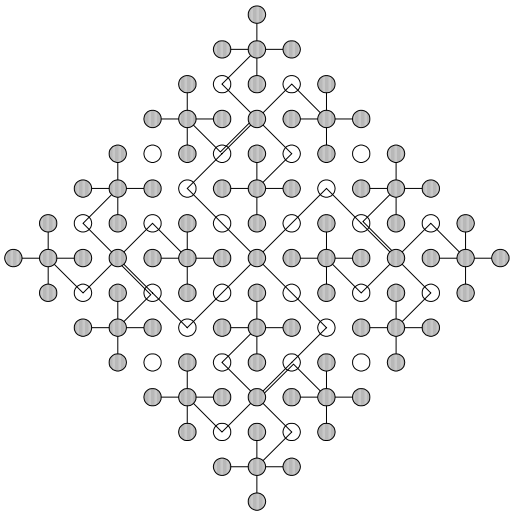

圖11 一個4-層四元樹嵌入到八邊形網格

樹和金字塔嵌入到八邊形網格後的圖形。當 *n*-層四元樹或金字塔被嵌入一個八邊形網格區塊後，該區塊內含節點總數可由下式表示：

$$M_n^{\mathrm{O}} = \left(2^{n-1}\right)^2 + \left(2^{n-1}-1\right)^2 \quad \text{.....(6)}$$

利用式（6）與式（1）可求得節點佔用比為：

$$r_n^{\mathrm{O}} = \frac{N_n}{M_n^{\mathrm{O}}} = \frac{\left(4^n-1\right)/3}{\left(2^{n-1}\right)^2 + \left(2^{n-1}-1\right)^2} \quad \text{.....(7)}$$

其中 O 與代表八邊形網格。圖 13 為實際數值帶入式（7）後所得到的曲線。由曲線可看出層數達 7 之後，節點佔用率即維持在 67%的水準，在我們所探討的三種網格結構中具有最高的佔用率。而當 $n\to\infty$，式（7）將收斂為 $0.\overline{6} \cong 0.67$，可知不論層數為何佔用率絕不會低於 67%。

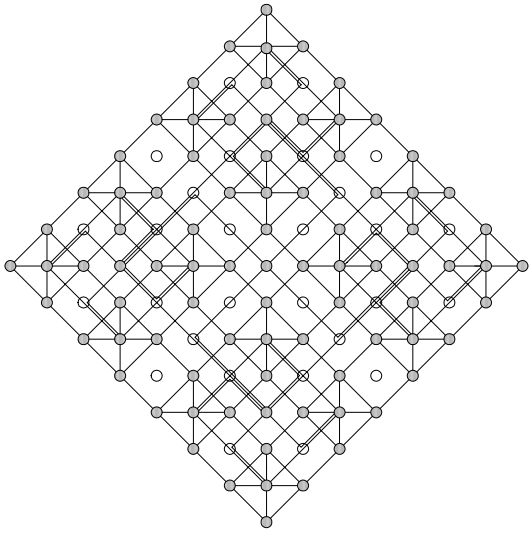

圖12 一個4-層完整的金字塔嵌入在八邊形網格上

## 4. 綜合比較

我們將以上所推導的結果詳列於表 1 及圖 14，以方便作完整的比較。由表及圖可看出，我們所提出的八邊形網格嵌入具有最高的節點利用率。尤其在與 Bhattacharya 所得出的結果比較時，我們的方法不僅獲得較高的利用率，並且無跨線之問題。

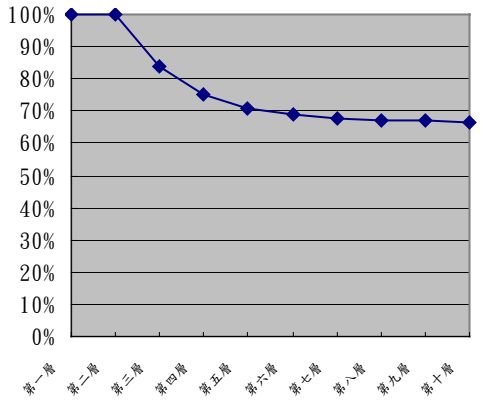


圖13 四元樹的節點佔用率（至十層）

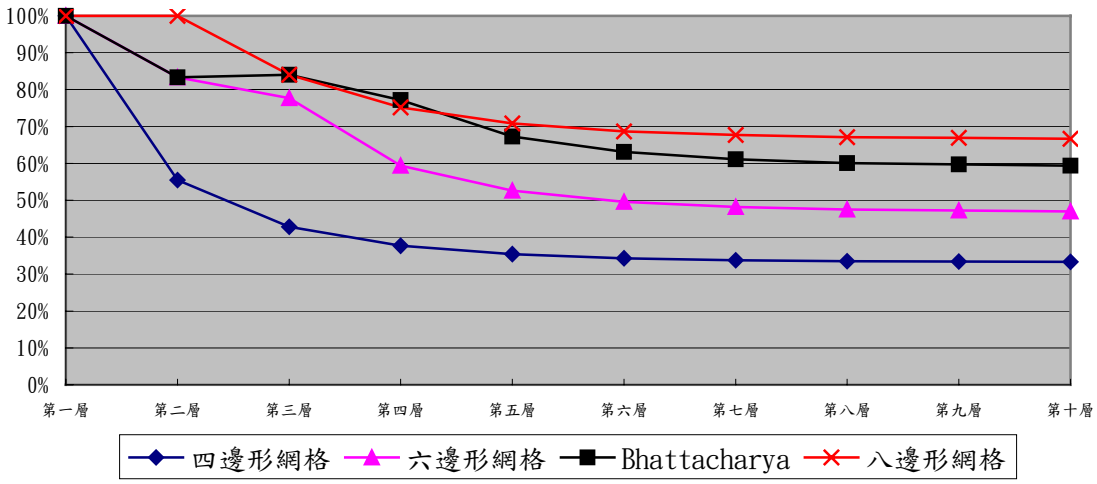


圖14 四種不同網格的嵌入的使用率比較

表 1 四種嵌入後的比較

| 型態 / n 層 | 四邊形網格 | 六邊形網格 | Bhattacharya | 八邊形網格 |
|---|---|---|---|---|
| 第一層 | 100% | 100% | 100% | 100% |
| 第二層 | 55.5% | 83.3% | 83.3% | 100% |
| 第三層 | 42.8% | 77.7% | 84% | 84% |
| 第四層 | 37.7% | 59.4% | 77.2% | 75.2% |
| 第五層 | 35.4% | 52.6% | 67.3% | 70.8% |
| 第六層 | 34.3% | 49.6% | 63.1% | 68.7% |
| 第七層 | 33.8% | 48.2% | 61.1% | 67.7% |
| 第八層 | 33.5% | 47.5% | 60.1% | 67.1% |
| 第九層 | 33.4% | 47.2% | 59.7% | 66.9% |
| 第十層 | 33.3% | 47.0% | 59.4% | 66.7% |

## 四、結論

本論文詳細探討了四元樹和金字塔結構的二維嵌入方法。我們以三種不同的網格排列說明我們如何將上述立體結構嵌入二維平面，並且導出每一種嵌入方式的節點使用率，其中基於八邊形網格的二維嵌入具有最高之利用率。此種嵌入方式有助於降低 VLSI 陣列佈線的複雜度，並且不受陣列規模的影響，能維持晶片面積的有效利用率。

## 參考文獻

[1] Battacharya, S., S.Kirani, and W. T. Tsai, "Quadtree Interconnection Network Layout," *Proceedings of the Second Great Lakes Symposium on VLSI*, pp.87-81, 1991.

[2] Gordan, D., I. Koren, and G.M. Silberman, " Embedding Tree Structures in VLSI Hexagonal Arrays," *IEEE Trans. Computer,* Vol. C-33, no. 1, pp.104-107, 1984.

[3] Gordon, D., "Efficient Embeddings of Binary Tree in VLSI Arrays," *IEEE Trans. Computers,* Vol. C-36, pp.1009-1018. Sept. 1987.

[4] Hwang, K. and F. A. Briggs, *Computer Architecture and Parallel Processi*ng. McGraw-Hill 1984.

[5] Kung, S. Y., *VLSI Array Processors.* Prentice -Hall, 1988.

[6] Leighton, F. T., *Introduction toParallel Algorithms and Architectures.* Morgan Kaufmann, 1992.

[7] Mead, C. and L. Conway, *Introduction to VLSI Systems.* Addision-Wesley, 1980.

[8] Miller, R. and Q. F. Stout, *Parallel Algorithms for Regular Architectures*. MIT

Press, 1999.

[9] Parhami, B., *Introduction to Parallel Processing.* Plenum Press, 1999.

[10] Roosta, S. H., *Parallel Processing and Parallell Algorithms*. Springer, 2000.

[11] Samet, H.,"The Quadtree and Related Hierarchical Data Structures," *ACM Computing Survey,* Vol. 16, pp.187-260, 1984.

[12] Snyder, L., "Introduction to the Configurable Highly Parallel Computer," *IEEE Computer,* pp.47-64, Jan. 1982.