Efficient VLSI Layout of Grid Pyramid Networks

Saeedeh Bakhshi[1,2], Marzieh Bakhshi[1] and Hamid Sarbazi-Azad[1,2]

[1]Sharif University of Technology, Tehran, Iran
[2]IPM School of Computer Science, Tehran, Iran
E-mail : bakhshi@ ipm.ir, m_bakhshi@ce.sharif.edu, azad@ ipm.ir

ABSTRACT

Reducing the VLSI layout area of on-chip networks can result in lower costs and better performance. Those layouts that are more compact can result in shorter wires and therefore the signal propagation through the wires will take place in less time. The grid-pyramid network is a generalized pyramid network based on a general 2D Grid structure (such as mesh, torus, hypermesh or WK-recursive mesh). Such pyramid networks form a wide class of interconnection networks that possess rich topological properties. In this paper, we study these topologies from the VLSI-layout efficiency point of view. Also, we investigated on the layout of RTCC-pyramid networks that we believe can be considered in the class of Grid-pyramid networks.

Keywords: *VLSI layout, Grid Pyramid networks*

1. INTRODUCTION

The number of transistors per chip has vastly increased in recent years and will likely increase by another order of magnitude in the next two decades [8]. These changes made it necessary for chip-multiprocessors and SoCs to use smaller layouts. More compact layouts can lead in shorter wire lengths and therefore reducing signal propagation delay as well as lower cost in the implementation process. The collinear layout that we discuss in this paper can result in better packagablity of the network, not only in one-layer chips, but also in today's technology for multi-level chip designs.

When the same physical chip area is used and the network is wire-bounded, a network layout with smaller layout area (when unit link width is assumed) will lead to an implementation with faster communication links. More precisely, a layout with smaller area by a factor of *f* (in terms of unit of area where a unit of length is taken as the physical width of a link) will lead to communication links whose bandwidth is higher than that of another layout with larger area by a factor of $\sqrt{f}$, since more wires can be used to implement a single communication link using the former layout. Thus, the impact of efficient VLSI layout on cost and performance become more crucial [10].

The grid-pyramid networks were first introduced in [2] as a generalized pyramid topology. Various topological properties of these networks were studied in [2]. Here, we study the packagibility and implementation properties of this family of networks. As we see in the following sections this class of networks are so appropriate in VLSI implementation according to their cost-effective layout area and the regular topologies they have. We selected these class of networks to show how efficient they can be placed on chips.

In section 2, we formally state some definitions, which are used throughout the paper; collinear layouts

and their special cases are introduced in this section. Section 3 briefly introduces the most known VLSI complexity model, called Thompson Model [7], which will be used in the subsequent sections. Number of required tracks and layout area of grid-pyramid networks are obtained in section 4. We present the layout of RTCC and RTCC-Pyramid network in section 5. Finally, the paper is concluded in section 6.

2. DEFINITIONS AND PRELIMINARIES

Definition 1. An $a \times b$ mesh network, $M_{a,b}$, is a set of nodes $V(M_{a,b}) = \{(x, y) \mid 1 \le x \le a, 1 \le y \le b\}$, where nodes (x_1, y_1) and (x_2, y_2) are connected by an edge iff $|x_1 - x_2| + |y_1 - y_2| = 1$[1]. See Fig. 1 (a).

Definition 2. The $a \times b$ torus network, denoted as $T_{a,b}$, consists of a set of nodes $V(T_{a,b}) = \{(x, y) \mid 1 \le x \le a, 1 \le y \le b\}$, where each node (x_1, y_1) is connected to its four neighboring nodes $(x_1 \pm 1 \bmod a, y_1)$ and $(x_1, y_1 \pm 1 \bmod b)$. See Fig. 1(b).

Definition 3. An $a \times b$ hypermesh, $HM_{a,b}$, is a set of nodes $V(HM_{a,b}) = \{(x, y) \mid 1 \le x \le a, 1 \le y \le b\}$ where nodes (x_1, y_1) and (x_2, y_2) are connected by an edge iff $x_1 = x_2$ or $y_1 = y_2$. See Fig. 1 (d).

Definition 4. A t-level WK-recursive mesh network $WK(d, t)$ with amplitude d and expansion level t, consists node set $V(WK(d, t)) = \{a_t a_{t-1} \ldots a_1 \mid 0 \le a_1 < t\}$. The node with address $A = (a_t a_{t-1} \ldots a_1)$ is connected to:

(1) all the nodes with addresses $(a_t a_{t-1} \ldots a_1 k)$, $0 \le k < t$, $k \ne a_1$, as sister nodes, and (2) node $(a_t a_{t-1} \ldots a_{j+1} a_{j-1} (a_j)^j)$, if for one j, $1 \le j < t$; $a_{j-1} = a_{j-2} = \ldots = a_1$ and $a_j \ne a_{j-1}$, as a cousin node. See Fig. 1.

Definition 5. A grid-pyramid of n levels, denoted by $P_{G,n}$, consists of a set of nodes $V(P_{G,n}) = \{(k, x, y) \mid 0 \le k \le n, 1 \le x, y \le 2k\}$. A node $(k, x, y) \in V(P_{G,n})$ is said to be node at level k. All the nodes in level k form a $2^k \times 2^k$ grid network G which can be one of the grid-based networks: mesh, torus, hypermesh or WK-recursive mesh, i.e. ($G \in M, T, HM, WK$); the resulted pyramid can then be then denoted as $P_{m,n}$, $P_{T,n}$, $P_{HM,n}$, and $P_{WK,n}$, respectively.

Fig. 2 shows all grid pyramid networks.

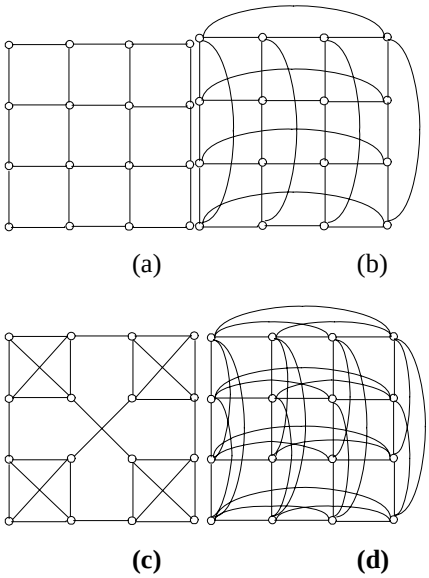

Fig. 1: The topologies of (a)$M_{4,4}$ (b)$T_{4,4}$ (c)$WK_{4,2}$ (d)$HM_{4,4}$

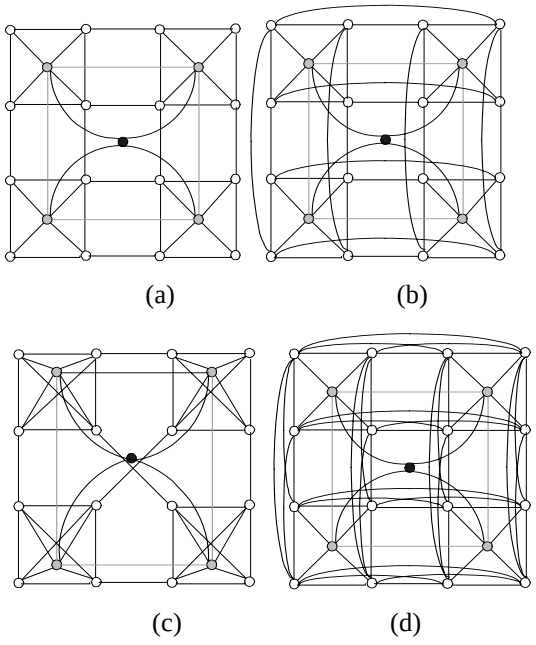

Fig. 2: The topologies of (a)$P_{M,2}$ (b)$P_{T,2}$ (c)$P_{WK,2}$ (d)$P_{HM,4}$

Definition 6. A VLSI layout is called collinear if all the nodes are placed along a straight line [11].

Definition 7. A C-node cycle consists of a set of nodes {0, 1, 2, 3, …, C-1} and set of edges $\{e_0, e_1, e_2, \ldots, e_{C-1}\}$ such that $e_i = (i, i+1 \bmod C)$.

Definition 8. The definition of the *RTCC*(C, L) is based on a C-node cycle. We name all the nodes in this cycle, *'Extern nodes'* or *'Open nodes'*. An *RTCC*(C, 2), consists of a number of C discrete *RTCC*(C, 1) networks, or C-node cycles, numbered 0 to C-1. Each external node *i* of each C-node cycle *j* is connected to node *j* of C-node cycle *i*. It is obvious that a node whose number is equal to the number of the *RTCC*(C, 2) in which it resides, is not directly connected to any other cycle, and is of node degree one less than other nodes in the network. There is one such node in each *RTCC*(C, 1) used to construct an *RTCC*(C, 2), and thus a total of C such nodes. We name these nodes as the external nodes of the *RTCC*(C, 2), the number of each one being equal to the number of the *RTCC*(C, 1) to which it belongs. In a similar manner, the *RTCC*(C, 3) can be defined as C discrete *RTCC*(C, 2) networks that are connected in such a way that each external node *i* in *RTCC*(C, 2) number *j* is connected to external node *j* in *RTCC*(C, 2) number *i*. Once again, a node whose number is equal to the address of the *RTCC*(C, 2) in which it resides, is not directly connected to any other cycle, and is of node degree one less than other nodes in the network [18]. Fig. 3 shows an RTCC (4,2).

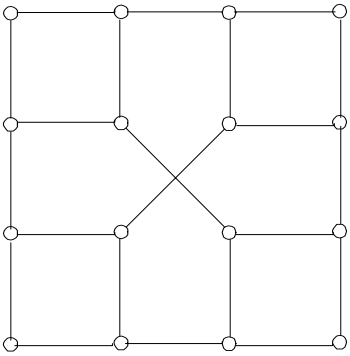

Fig. 3: The topology of RTCC(4,2)

Definition 9. An *RTCC-pyramid* network, denoted as $P\text{-}RTCC_{(C,L)}$, consists of a set of nodes $V(P\text{-}RTCC_{(C,L)}) = \{ (k, (a_k a_{k-1} \ldots a_1)) \mid 0 \le k \le L, 0 \le a_i \le C\text{-}1, 1 \le i \le k \text{ or } k = 0 \text{ and } a_1 = 1 \}$. A node with addressing scheme $(k, (a_k a_{k-1} \ldots a_1))$ is said to be a node at level *k*, e.g. the *apex* is at level 0. The part $(a_k a_{k-1} \ldots a_1)$ of the address determines the address of a node within the *RTCC* network at layer *k*.

All the nodes in level *k* form an $RTCC_{(C, L)}$ network. Hence, there exist a total of $N = \Sigma_{k=0}^{L} C^k = (C^{L+1}\text{-}1)/C\text{-}1$ nodes in a $P\text{-}RTCC_{(C, L)}$. A node with address $(k,(a_k a_{k-1} \ldots a_1))$ is connected, within the *RTCC* network at level $k > 0$, to node $(k,(a_k a_{k-1} \ldots a_2 (a_1 \pm 1)_{\text{mod}C}))$, as the neighbouring *brother* nodes, and connected to a node with address schema $(k,(a_k a_{k-1} \ldots a_{j+2}\ a_{j+1} a_j (a_{j-1})^j)$, $1 \le j \le L\text{-}1)$ if there exists one *j* such that $1 \le j \le L\text{-}1$, $a_{j-1} = a_{j-2} = \ldots = a_1$ and $a_j \ne a_{j-1}$; as a cousin node (nodes at the same level). This node is also connected to nodes $(k+1,(a_k a_{k-1} \ldots a_2 a_1 b)$, for $1 \le b \le C$, in level $k+1$, as a *child* node, and connected to node $(k\text{-}1,(a_k a_{k-1} \ldots a_2))$, in level $k\text{-}1$, as a *father* node. Fig. 4 illustrates an RTCC-pyramid network, $P\text{-}RTCC_{(4,2)}$ [19].

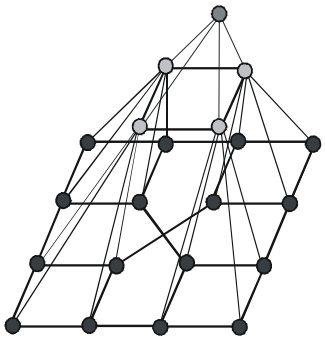

Fig.4. An $P\text{-}RTCC_{(4, 2)}$ network. We illustrate the apex node and the corner nodes of low-grade level in the network.

3. THOMPSON GRID MODEL

Thompson proposed a mathematical model for VLSI computations which is widely accepted and is known as the Thompson grid model [7]. In this model, he presumed the chip consists of some vertical and horizontal tracks which are spaced apart at unit intervals. Two layers of interconnect are used to route the wires. Vertical wires are routed in one layer while horizontal wires are routed in the other. The circuit is viewed as a graph *G* in which vertices correspond to processing elements and edges to wires.

Wires can cross each other but cannot overlap with each other. To change direction, wires may turn into the other layer by vias. The graph is then

embedded in a two-dimensional grid. An embedding of a graph G in a Thompson grid is an assignment of nodes of G to intersection points in the grid and the edges of G to paths along the grid tracks. The area of a layout is the area of the smallest upright rectangle that contains all the nodes and wires. When there are two layers of wires, it is guaranteed that we can lay out the network within the area. The maximum wire length is the length of the longest wire in the layout.

4. LAYOUT OF GRID-PYRAMID

In this section, we present a method for laying out $P_{G,.n}$ network, on chip. We use a recursive method based on layout of the base network for constructing the layout.

First, we assume that we have an efficient layout scheme for the base network.

To find the collinear layout complexity of $P_{G,n}$ we use a recursive process. Let the number of tracks which are needed by $P_{G,n}$ be $T(P_{G,n})$. Consider the case where there is just one node in $P_{G,n}$ and no links. This is the case of $P_{G,0}$ for which we have $T(P_{G,0}) = 1$, because it is an isolated node that can be put in one square.

For $P_{G,n}$, we need to lay $P_{G,n-1}$ out and add a $G_{2^n,\ 2^n}$ with $4N$ $(G_{2^{n-1},\ 2^{n-1}})$ edges connected the base network to the last layer of $P_{G,n-1}$. Hence, the layout of $P_{G,n}$ requires $T(P_{G,n}) = 4N\ (G_{2^{n-1},\ 2^{n-1}}) + T\ (G_{2^n,\ 2^n})$ tracks.

4.1 Mesh-Pyramid layout

Now, let us investigate the layout of the mesh-pyramid. For this purpose it is needed to find the number of tracks required by mesh networks in their layout. We can put each row of the mesh in a track, then all links connecting those nodes in the same track. Then, we should put n copies of such a row in a grid area to obtain an efficient layout of n^2-area for the $M_{n,n}$.

Fig.5 shows the collinear layout of $M_{4,4}$. For building Mesh-Pyramid network we can use the same layout mechanism.

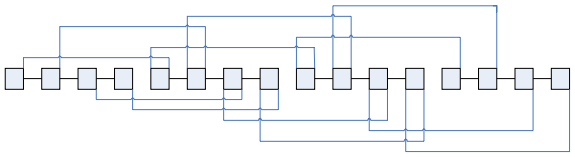

Fig. 5: Collinear layout of $M_{4,4}$

Now, let us examine the area needed by $P_{M,1}$, which is a $M_{2,2}$ that all of its vertices are connected to the apex node. It is clear that $T(P_{M,1}) = 4N\ (P_{M,0}) + T(M_{2^1,2^1})$. Since $T(M_{2,2}) = 4$, we can simply conclude that $P_{M,1}$ needs 8 tracks to lay out on a chip. Similarly, for $P_{M,n}$, we need to lay $P_{M,n-1}$ out and add a $M_{2^n\times 2^n}$ with $4N$ $(M_{2^{n-1}\times 2^{n-1}})$ edges connecting the base mesh to the last layer of $P_{M,n-1}$. Hence, the collinear layout of $P_{M,n}$ needs $T(P_{M,n}) = 4N\ (M_{2^{n-1}\times 2^{n-1}}) + T\ (M_{2^n\times 2^n}) = 2^{2n} + O\ (2^n)$ tracks. Since $N = N\ (P_{M,n}) = (4^{n+1} - 1)/3$, we have

$$T(P_{M,n}) = \frac{3N}{2} + O\left(\sqrt{N}\right)$$

Since a mesh-pyramid has a maximum vertex degree of 9, if each node in the layout consumes a 9 square area, then $P_{M,n}$ can be laid out in $13.5N^2 + O\ (N\sqrt{N})$ area. In Fig.6. the collinear layout of $P_{M,2}$ is shown.

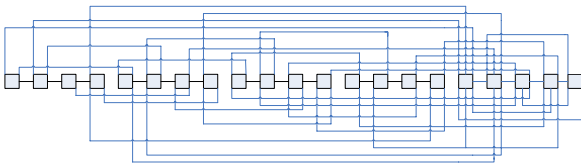

Fig. 6: Collinear layout of $P_{M,2}$

4.2 Torus-Pyramid layout

Now that we have obtained the number of tracks and the area needed by mesh-pyramid to lay out on a chip, we use the same approach to describe layout of the torus-pyramid.

Yeh and Parhami in [4] described a method for laying out k-ary n-cubes. They used a bottom-up approach, starting with a k-node ring. Using their method, we need $2n + 2$ tracks and $n^2 + O\ (n^2)$-area for the layout of $T_{n,n}$. In Fig. 7 we can see a collinear layout of $T_{4,4}$.

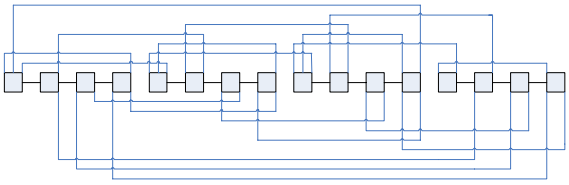

Fig. 7: Collinear layout of $T_{4,4}$

Similar to that of Mesh-Pyramid, for the $P_{T,n}$, we need to lay $P_{T,n-1}$ out and add a $T_{2^n,2^n}$ with $4N$ ($T_{2^{n-1}\times 2^{n-1}}$) edges connecting the mesh to the last layer of $P_{T,n-1}$. Hence, the layout of $P_{T,n}$ needs $T(P_{T,n}) = 4N(T_{2^{n-1},2^{n-1}}) + T(T_{2^n,2^n}) = 2^{2n} + O(2^{n+1})$ tracks. Since $N = N(P_{T,n}) = (4^{n+1} - 1)/3$, we have $T(P_{T,n}) = \frac{3N}{2} + O(\sqrt{N})$. As the torus-pyramid has a maximum vertex degree of 9, each node in the layout consumes a 9 square area, so the area consumed by $P_{T,n}$ on a chip would be $13.5N^2 + O(N\sqrt{N})$ squares (like that of a mesh-pyramid). Fig. 8 shows the layout of $P_{T,2}$.

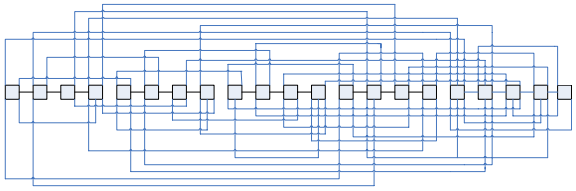

Fig. 8: Collinear layout of $P_{T,2}$

4.3 Hypermesh-Pyramid layout

Layout of the hypermesh and its pyramid network is slightly different to that of the mesh-pyramid and torus-pyramid.

In [5], it is shown that an N-node complete graph can be optimally laid out using $\lfloor N^2/4 \rfloor$ tracks for a collinear layout, and can be laid out in $N^4/16 + o(N^4)$ area for a 2D layout. Fig. 9 shows the collinear layout of $HM_{4,4}$.

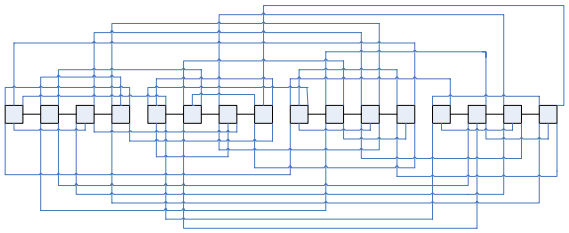

Fig. 9: Collinear layout of $HM_{4,4}$

For the layout of a hypermesh-pyramid, we have: $T(P_{HM,1}) = 4N(P_{HM,0}) + T(HM_{2^1,2^1})$. Since $T(HM_{2,2}) = 4$, $P_{HM,1}$ needs 8 tracks for its layout. For a $P_{HM,n}$, we have :

$T(P_{HM,n}) = 4N(HM_{2^{n-1}\times 2^{n-1}}) + T(HM_{2^n \times 2^n}) = 2^{2n} + o(2^{2n})$ tracks.

Since $N = N(P_{HM,n}) = (4^{n+1} - 1)/3$, we have $T(P_{HM,n}) = 1.5N + o(N)$.

In a hypermesh-pyramid, each node is connected to all of nodes in the same row and the same column. Also, each node is adjacent to its father (except for the apex) and four children (except for the last layer nodes), hence the maximum degree of a node in a hypermesh-pyramid equals $2^{n+1} - 1$ for $n \geq 2$. So, the hypermesh-pyramid can be laid out in the area of $2^{3n+1} + o(2^{3n+1})$ which is $6N\sqrt{6N} + o(N\sqrt{N})$-area. You can see collinear layout of $P_{HM,2}$ in Fig. 10.

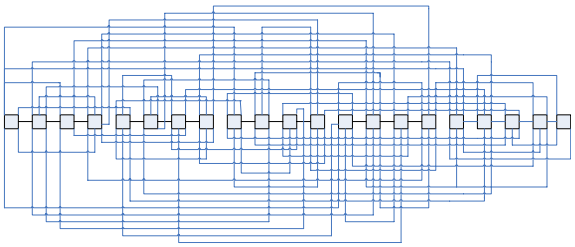

Fig. 10: Collinear layout of $P_{HM,2}$

4.4 WK-Pyramid layout

Finally, we work on the VLSI layout of WK-pyramid networks. In [6], an efficient layout for WK-recursive networks was proposed. The $WK(d, t)$ network requires $\frac{d^{t+1} - d^t + 2d^{t-1} - 2}{d-1}$ tracks to lay out collinearly. It means it needs an area of $N^2/16 + o(N^2)$. The collinear layout of $WK_{4,2}$ is shown in Fig. 11.

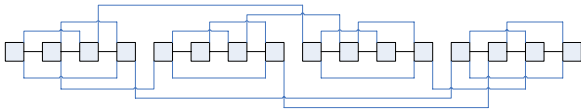

Fig. 11: Collinear layout of $WK_{4,2}$

Here, we use the same scheme to obtain the layout of a WK-pyramid network. To this end, it is enough to lay out $WK(4, n)$. So, the layout of $P_{WK,n}$ needs $T(P_{WK,n}) = 4N(WK(4, n-1)) + T(WK(4,$

n)) tracks. Since $N\ (W\,K\ (4, n-1)) = 4^{n-1}$, we have $T\ (P_{WK,\,n}) = \frac{13}{6} \times 4^n - \frac{2}{3}$. That means $P_{WK,\,n}$ can be laid out in $39/2N^2 + O(N)$ area. Collinear layout of $P_{WK,2}$ is shown in Fig. 12.

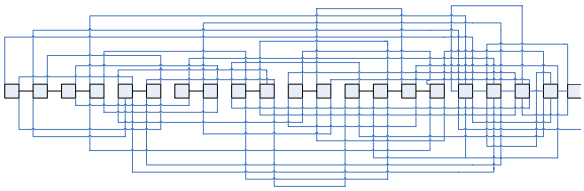

Fig. 12: Collinear layout of $P_{WK,2}$

5. RTCC AND RTCC-PYRAMID LAYOUTS

In this section we investigate on another grid network, which is not one of the networks introduced in [2] for using as a base network while building pyramids, but we think that it has the important characteristics of a grid network. We follow with the two dimensional and multilayer layout of RTCC, and then use them for constructing the RTCC-Pyramid layout.

5.1 Two dimensional layout of RTCC

An *RTCC*(C, L) contains C subgraphs isomorphic to *RTCC*(C, L-1), each pair of which are connected by 1 link. If we let *RTCC*(C, L-1) as a supernode, then the *RTCC*(C, L) becomes a C-supernode complete graph.

To lay out an *RTCC*(C, L), we first place nodes belonging to each *RTCC*(C, L-1) into a block which we call (L-1)-block, and lay out the *RTCC*(C, L) using that of complete graph, stated in [5]. Then we should continue this process recursively until the remainder graph is the C-ring and then we can simply lay out the ring.

To lay out *RTCC*(*C*, *L*), base on the layout of the complete graph [5], the area of $N^2 / 16 + o\ (N^2)$ will be required. For level h-block we first connect the wires outside the block to appropriate the (h-1)-blocks within it, and then follow the technique of laying out the K_{C^h} graph. Hence, the layout area will be obtained from:

$$\frac{C^{2L}}{16} + o\left(C^{2L}\right) + \frac{C^{2(L-1)}}{16} + o\left(C^{2(L-1)}\right) + \ldots + 1 = \frac{C^{2L}}{16} + o\left(C^{2L}\right)$$

and so an *RTC*(C, L) can be laid out in $N^2/16 + o(N^2)$, where $N = C^L$. The collinear layout of RTCC (4, 2) is shown in Fig. 13.

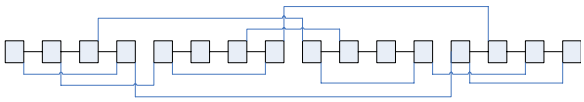

Fig. 13: Collinear layout of RTCC(4,2)

5.2 Multilayer layout of RTCC

In order to obtain the multilayer layout of *RTCC*(C, L) we use the same bottom-up approach, used in [4] for k-ary n-cube and several other networks.

A collinear layout of a ring can be obtained by placing the C nodes along a row and connecting neighboring nodes in one track and then connecting node 0 to C-1, in another track [4]. Let the number of tracks which are needed by *RTCC*(C, L-1) be $f_C\ (L-1)$. To obtain the collinear layout of *RTCC*(C, L) we need C copies of *RTCC*(C, L-1) which are placed in a space, horizontally C times of the space of *RTCC*(C, L-1) and two extra tracks to connect the level L inter-block links. Therefore, the number of tracks needed to layout the *RTCC*(C, L) is $f_C\ (L) = Cf_C\ (L-1) + 2$, and because $f_c\ (1) = 2$, we have

$$f_C\ (L) = Cf_C\ (L-1) + 2\ (C^{L-1} + C^{L-2} + \ldots + C + 1)$$

$$f_C\ (L) = \frac{2(C^L - 1)}{C-1} = \frac{2(N-1)}{C-1}$$

Now, we use the approach of orthogonal multilayer layout [4] scheme to obtain an S-layer layout.

The number of tracks per layer above a row is $\left\lceil \frac{2\left(C^{\lfloor L/2 \rfloor} - 1\right)}{\lfloor S/2 \rfloor (C-1)} \right\rceil$ and the number of tracks per layer to the right of the column is $\left\lceil \frac{2\left(C^{\lfloor L/2 \rfloor} - 1\right)}{\lfloor S/2 \rfloor (C-1)} \right\rceil$. So the area of the S-layer *RTCC*(C, L) is:

$$\frac{8N^2}{\lfloor S^2/2 \rfloor C^2} + o\left(\frac{N^2}{S^2C^2}\right)$$

And its volume is: $\frac{8N^2}{\lfloor S/2 \rfloor C^2} + o\left(\frac{N^2}{SC^2}\right)$

5.3 Layout of RTCC-Pyramid

Now that we obtained an efficient layout for RTCC network, we can use it in constructing a layout for its pyramid network.

For the *P-RTCC(C,L)* layout, it is enough to layout *RTCC(C,L)*, and then connect each node belongs to the last layer of *P-RTCC(C,L-1)* to the corresponding node in *RTCC(C,L)*. So the collinear layout of *P-RTCC(C,L)* needs $T(P - RTCC(C, L)) = C \times N\ (RTCC(C, L-1)) + T(RTCC(C, L))$ tracks.

Since $N(RTCC(C, L-1)) = \frac{3C^L - C}{2}$ and by theorem 7, $T(RTCC(C,L)) = \frac{2(C^L - 1)}{C-1}$, we have $T(P - RTCC(C, L)) = \frac{3C^{L+1} - C^2}{2} + \frac{2(C^L - 1)}{C-1} = \frac{3}{2} C^{L+1} + O(C^{L-1}) = \frac{3}{2} CN + O\ (N/C)$ and so a *P-RTCC(C,L)* needs $\frac{3}{2} CN + O\ (N/C)$ tracks to layout collinearly, where N is the number of nodes of *P-RTCC(C,L)*.

Since *P-RTCC(C,L)* has a maximum vertex degree of $C + 4$, it can be laid out in $\frac{3}{2} C^2N^2 + O\ (CN^2)$ area. You can see the collinear layout of *P-RTCC(4,2)* in Fig. 14.

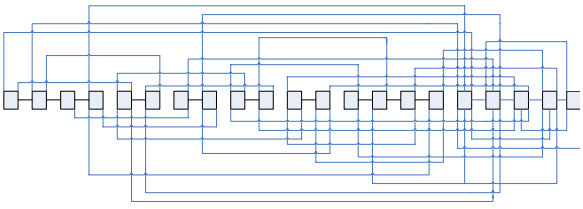

Fig. 14: Collinear layout of P-RTCC(4,2)

6. CONCLUSION

In this paper, we have investigated on the area of layouts for grid-pyramid networks, including the mesh-pyramid, torus-pyramid, hypermesh-pyramid and WK-pyramid. Also we added RTCC as base network for grid pyramid networks and studied RTCC-Pyramid's layout on chip. We have obtained the number of tracks required for collinear layout of these networks as well as the area required for them on a chip. We showed that a mesh-pyramid (or torus-pyramid) requires $13.5N^2 + O\ (N\sqrt{N})$ area for its layout, while a hypermesh-pyramid consumes an area of $6N\sqrt{6N} + o\ (N\sqrt{N})$ and a WK-pyramid network can be laid out on an area of $39/2N^2 + O(N)$. We also showed that an RTCC network can be laid out in $N^2/16 + o\ (N^2)$ area and $\frac{8N^2}{\lfloor S/2 \rfloor C^2} + o\left(\frac{N^2}{SC^2}\right)$ volume (when the chip is multilayer). By using the RTCC layout results, we obtained $\frac{3}{2} C^2N^2 + O\ (CN^2)$ as the area boundary needed for RTCC-Pyramid network to sit on the chip.

REFERENCES

1. R. C. Baker and B. Charlie, "Nonlinear Unstable Systems," International Journal of Control, Vol. 23, No. 4, pp. 123-145, 1989.
2. M. HoseinyFarahabady, H. Sarbazi-Azad, "The grid-pyramid: A generalized pyramid network", Journal of Supercomputing, Vol. 37, pp. 23-45, 2006.
3. F. T. Leighton, *Complexity Issues in VLSI*, MIT Press, Cambridge, Mass, 1983.
4. C-H. Yeh, E.A. Varvarigos, and B. Parhami, "Multilayer VLSI layout for interconnection networks," Proc. Int'l Conf. Parallel Processing, pp. 33-40, 2000.
5. C-H. Yeh, and B. Parhami, "VLSI layouts of complete graphs and star graphs," Information Processing Letters, Vol. 68, pp. 39-45, 1998.
6. S.Bakhshi, and H. Sarbazi-Azad, "Efficient VLSI layout of WK-recursive based networks", Technical report, IPM, Tehran, Iran, 2007.

7. C. D Thompson, “A Complexity Theory for VLSI,” PHD thesis, Carnegie-Mellon Univ., 1980.
8. W.J. Dally, and S. Lacy, “VLSI architecture: past, present, and future,” Proc. Advanced Research in VLSI Conf., 1999.
9. S. N. Bhatt and F. T. Leighton, “A framework for solving VLSI Graph Layout Problems”, Journal of Computer and System Sciences, vol. 28, pp.300-343, 1984.
10. C.H. Yeh, “optimal layout for fast Fourier transform in multilayer VLSI”**,** Proc. of 18th International Symposium on Parallel and Distributed Processing, pp. 91- , 2004.
11. A. Fernandez and K. Efe, “Efficient VLSI layouts for homogeneous product networks”, *IEEE Trans. Computer*, vol. 46, no. 10, pp. 1070-1082, 1997.
12. S.Bakhshi, and H. Sarbazi-Azad, “Efficient VLSI layout of Edge Product networks”, To appear in proc. of 4th IEEE International Symposium on Electronic, Design, Test and Applications, 2008.
13. Calamoneri T. and R. Petreschi, **“**Optimal layout of trivalent Cayley interconnection networks,” Int’l J. Foundations of Computer Science, Vol. 10, no. 3, pp. 277-287, 1999.
14. Chen G. and F.C.M. Lau, “A compact layout of cube-connected cycles,” Proc. Int’l Conf. High Performance Computing, pp. 422-427, 1997.
15. Yeh, C.-H. and B. Parhami, “On the VLSI Area and Bisection Width of Star Graphs and Hierarchical Cubic Networks,” Proc. of the 15th International Parallel and Distributed Processing Symposium, Vol. 1, pp.10072, 2001.
16. S. N. Bhatt and F. T. Leighton, “A framework for solving VLSI Graph Layout Problems,” J. Computer and System Sciences, vol. 28, pp.300-343, 1984.
17. Yeh, C.-H., E.A. Varvarigos, and B. Parhami, “Efficient VLSI layouts of hypercubic networks,” Proc. Symp. Frontiers of Massively Parallel Computation, pp. 98-105, 1999.
18. M. Hoseiny Farahabady, H. Sarbazi-Azad, “The Recursive Transpose-Connected Cycles (RTCC) Interconnection Network for Multiprocessors”, ACM SAC’05, Santa Fe, New Mexico, USA, 2005.
19. M. Hoseiny Farahabady, H. Sarbazi-Azad, “The RTCC-pyramid: a versatile pyramid network”, High-Performance Computing in Asia-Pacific Region Proc. Issue. 30, 2005