

A Tight Layout of the Cube-Connected Cycles

Guihai Chen and Francis C.M. Lau
Department of Computer Science
The University of Hong Kong
Pokfulam Road, Hong Kong
{gchen,fcmlau}@cs.hku.hk

Abstract

Preparata and Vuillemin proposed the cube-connected cycles (CCC) in 1981 [16], and in the same paper, gave an asymptotically-optimal layout scheme for the CCC. We give a new layout scheme for the CCC which requires less than half of the area of the Preparata-Vuillemin layout. We also give a non-trivial lower bound on the layout area of the CCC. There is a constant factor of 2 between the new layout and the lower bound. We conjecture that the new layout is optimal (minimal).

Keywords: *Interconnection networks, VLSI, cube-connected cycles, embedding, routing, layout.*

1 Introduction

Interconnection network is one of the most crucial design issues for parallel computers. There are many criteria to be considered in choosing a specific interconnection for a given set of processors. With the rapid technological progresses in VLSI, it is now common to connect a huge number of processors together to cooperate for the execution of parallel algorithms. Obviously, in these situations, one of the criteria for packing these processors together would be the “compactness” of the layout in a VLSI grid. In general, the more compact the better.

The cube-connected cycles (CCC) is one of the most popular interconnection networks. Preparata and Vuillemin [16] put forward the CCC as a practical substitute for the hypercube in 1981, and at the same time gave an asymptotically-optimal layout scheme for it. Their layout scheme, however, cannot produce the *minimal* layout for the CCC. Our work addresses this issue. We have two goals: one is to derive a more compact layout for the CCC than the Preparata-Vuillemin layout; and the other is to reduce the long wires in the layout while keeping the asymptotically-optimal area. This paper reports the result coming from our effort in trying to achieve the first goal.

Research in graph embedding and VLSI layout has developed many powerful techniques [2, 5] which can produce embeddings and layouts that are quite efficient—often within constant factors of being optimal. However, even a modest constant factor may render an asymptotically-optimal layout or embedding unacceptably inefficient in practice. This motivates the current paper.

2 Preliminaries

2.1 The Thompson Model

Among the many mathematical models that have been proposed for VLSI computations, the most widely accepted one is due to Thompson, which is now known as the Thompson grid model [18, 19]. In this model, the chip is presumed to consist of a grid of vertical and horizontal tracks which are spaced at unit intervals. Two layers of interconnect are used to route the wires. Vertical wires are routed in the top layer of the interconnect and horizontal wires are routed in the bottom layer. Hence, wires may cross but they cannot overlap for any distance or cut cross a node to which they are not incident. To change direction, wires may turn into the other layer by contact cuts or vias which facilitate connections between the two layers. In our discussion, no knock-knees are allowed—that is, two wires cannot turn at the same grid point [14, 15].

Formally, an embedding or layout of a graph \mathcal{G} in a Thompson grid is an assignment of the nodes of \mathcal{G} to intersection points in the grid and the edges of \mathcal{G} to paths along the grid tracks. One of the important measures of a layout is the layout area which is defined as the product of the number of vertical tracks and the number of horizontal tracks that contain a node or a path segment of the graph.

2.2 Cube-Connected Cycles

The s -dimensional cube-connected cycles (CCC) is constructed from the s -dimensional hypercube by replacing each node of the hypercube with a cycle of s nodes [13, 16]. The i th-dimension edge incident to a node of the hypercube is then connected to the i th node of the corresponding cycle in the CCC. For example, see Fig. 1(a,b). The resulting graph has $s2^s$ nodes, each of degree 3. By adopting the labeling scheme of the corresponding hypercube and modifying it slightly to take into account the cycles that are introduced, we can represent each node of the CCC by the pair $\langle w, i \rangle$ where i ($0 \leq i \leq s-1$) is the position of the node within its cycle and w (any s -bit binary string with dimension 0 being at the rightmost bit position) is the label of the node in the hypercube that corresponds to the cycle. Then two nodes $\langle w, i \rangle$ and $\langle w', i' \rangle$ are linked by an edge in the CCC if and only if either

- (1) $w = w'$ and $i - i' = \pm 1 \pmod{s}$, or
- (2) $i = i'$ and w differs from w' in precisely the i th

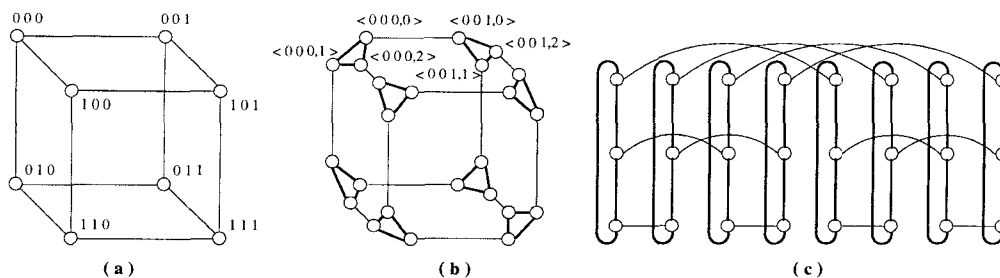


Figure 1: (a) 3-dim hypercube. (b) 3-dim CCC. (c) Another drawing of 3-dim CCC; cyclic edges in thick and cubical edges in thin.

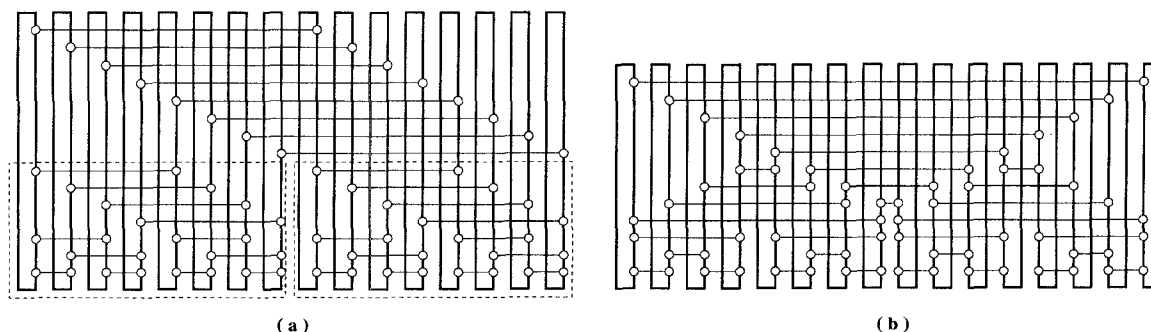


Figure 2: (a) Preparata-Vuillemin layout for 4-dim CCC. (b) Improved Preparata-Vuillemin layout for 4-dim CCC.

bit. Edges due to (1) are cyclic edges and edges due to (2) are cubical edges. As shown in Fig. 1(c), the CCC is often drawn in the “multi-stage” format which can directly give rise to the Preparata-Vuillemin layout.

2.3 The Preparata-Vuillemin Layout

Fig. 2(a) provides a base, inductive hypothesis for proving that a CCC of $N = s \cdot 2^s$ nodes can be placed on a $2 \cdot 2^s \times (2^s + 1)$ chip. Since $s \simeq \log(N/\log N)$, the chip size is about $O((N/\log N)^2)$. In general, we say that a network of N nodes has an (asymptotically) optimal layout if it can be laid out in $O(N^2/T^2)$ area, where T is the time to execute an ascend-descend algorithm [4, 19]. The CCC can execute the ascend-descend algorithm in time $O(\log N)$ [16]. Therefore, the Preparata-Vuillemin layout is (asymptotically) optimal.

For an s -dimensional CCC with $n = 2^s$ cycles, which we denote by CCC_n , let $W(s)$ and $H(s)$ be the numbers of vertical and horizontal tracks respectively, *i.e.*, the width and the height of a layout. For the Preparata-Vuillemin layout, we have

$$\begin{aligned} W(1) &= 4, \\ H(1) &= 3, \\ W(s) &= 2W(s-1), \\ H(s) &= H(s-1) + 2^{s-1}. \end{aligned}$$

We get $W(s) = 2^{s+1} = 2n$ and $H(s) = 2^s + 1 = n + 1$. Hence the area occupied by the Preparata-Vuillemin

layout is $W(s) \times H(s)$, *i.e.*,

$$2n(n+1) = 2n^2 + 2n. \quad (1)$$

For the improved Preparata-Vuillemin layout which is shown in Fig. 2(b),

$$\begin{aligned} W(1) &= 4, \\ H(1) &= 3, \\ W(s) &= 2W(s-1), \\ H(s) &= \begin{cases} H(s-1) + 2^{s-1} & \text{if } s \text{ is odd} \\ H(s-1) + 2^{s-2} + 1 & \text{if } s \text{ is even.} \end{cases} \end{aligned}$$

We get $W(s) = 2^{s+1} = 2n$ and $H(s) = 3 + (2 + 4 + 5 + \dots + 2^{s-2} + (2^{s-2} + 1)) = \frac{2}{3}2^s + \frac{1}{2}s + \frac{4}{3}$ for even s , and $H(s) = \frac{5}{6}2^s + \frac{1}{2}s + \frac{5}{6}$ for odd s . For simplicity we only consider even s . Hence the area is

$$\frac{4}{3}n^2 + n \log n + \frac{8}{3}n. \quad (2)$$

3 New Layouts

Although the Preparata-Vuillemin layout for the CCC is asymptotically optimal (up to a constant), it is not the minimal layout. For real implementations, we are interested in the minimal layout. Here, we give a new layout for the CCC. It is more compact than the Preparata-Vuillemin layout; we conjecture that this layout is optimal (minimal).

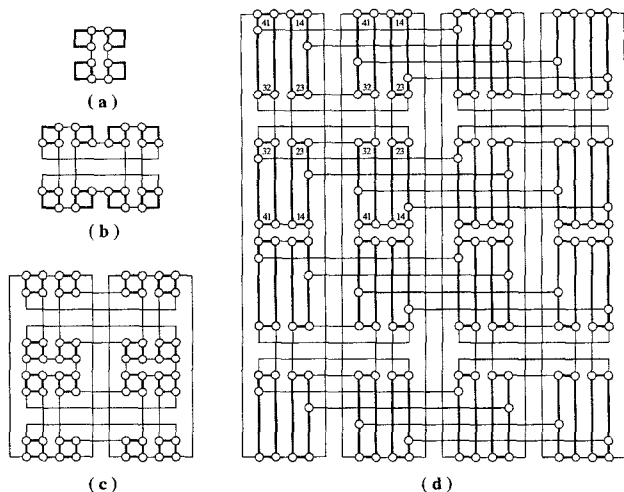


Figure 3: New layouts of small *CCC*'s. (a) 2-dim *CCC* needs area 4×4 . (b) 3-dim *CCC* needs area 8×6 . (c) 4-dim *CCC* needs area 12×12 . (d) 5-dim *CCC* needs area 24×28 ; note that all 5th-dimension nodes are inserted into the same cycles that are in the 4-dim *CCC*.

3.1 Small *CCC*'s

With the new layout scheme, the layouts for the several initial small *CCC*'s are as shown in Fig. 3(a,b,c). It can be easily verified that these few simple cases do occupy minimal area. These small *CCC* layouts are the foundation on which to recursively lay out bigger *CCC* networks.

3.2 Recursive Construction

Starting from the 5th dimension, the construction is inductive. We take two copies of the layout for the $(s-1)$ -dimensional *CCC* and place them side by side. Stretch every cycles vertically by an extra height of 2^{s-3} to allow for the insertion of the s th-dimension nodes and edges. Since there are four rows of cycles from top to bottom, totally 2^{s-1} extra horizontal tracks are added. Note that at each recursive expansion, all new nodes (*i.e.*, the s th-dimension nodes) are inserted into the same cycles, which ensures the correctness of the new layout scheme. We show the 5-dimensional *CCC* and the 6-dimensional *CCC* that are laid out using the new scheme in Fig. 3(d) and Fig. 4(a) respectively. For the s -dimensional *CCC* with $n = 2^s$ cycles, it is easy to see that

$$\begin{aligned} W(4) &= 12, \\ H(4) &= 12, \\ W(s) &= 2W(s-1), \\ H(s) &= H(s-1) + 2^{s-1}. \end{aligned}$$

We get $W(s) = 12 \times 2^{s-4} = \frac{3}{4}n$ and $H(s) = 2^s - 4 = n - 4$. Hence the area $W(s) \times H(s)$ is

$$\frac{3}{4}n^2 - 3n. \quad (3)$$

Like the improved Preparata-Vuillemin layout, the new layout can also be improved. The improved new

layout of the 6-dimensional *CCC* is shown in Fig. 4(b). For the improved new layout scheme,

$$\begin{aligned} W(4) &= 12, \\ H(4) &= 12, \\ W(s) &= 2W(s-1), \\ H(s) &= \begin{cases} H(s-1) + 2^{s-1} & \text{if } s \text{ is odd} \\ H(s-1) + 2^{s-2} + 4 & \text{if } s \text{ is even.} \end{cases} \end{aligned}$$

We get $W(s) = 12 \times 2^{s-4} = \frac{3}{4}n$ and $H(s) = 12 + (16 + 20 + 64 + 68 \cdots + 2^{s-2} + (2^{s-2} + 4)) = \frac{2}{3}2^s + 2s - \frac{20}{3}$ for even s . Hence the area is

$$\frac{1}{2}n^2 + \frac{3}{2}n \log n - 5n. \quad (4)$$

3.3 Comparison

By ignoring the low-order terms in Formulae 1, 2, 3 and 4, the four layout schemes of *CCC* _{n} discussed above take areas approximately equal to $2n^2$, $\frac{4}{3}n^2$, $\frac{3}{4}n^2$ and $\frac{1}{2}n^2$ respectively. We compare the new layout with the Preparata-Vuillemin layout, and the improved new layout with the improved Preparata-Vuillemin layout: The new layout scheme takes less than half of the area of the Preparata-Vuillemin layout in either case; the crux of the new layout is that the corner points of the cycles (now as rectangles in grid) are occupied by nodes (processors) so that their corresponding cubical edges need no or little extra space, which is unlike the Preparata-Vuillemin layout.

The new compact layout presented here also shows the superiority of the *CCC* in this aspect over other hypercube substitutes such as the shuffle-exchange network and the butterfly network [13]. Much work had been devoted to the layout of the shuffle-exchange network until Leighton found its optimal layout [12]. However, all these layouts of the shuffle-exchange network are complicated; they are not regular or recursive. The best known layout of the butterfly network with n inputs or outputs was due to Wise [20] with area $\simeq 2n^2$. Recently, more compact layouts for the butterfly were found with area $\simeq \frac{11}{6}n^2$ [8], or $n^2 + o(n^2)$ [2]. However, the butterfly networks discussed in [2, 8, 20] are unfolded. To be fair, the folded butterfly network [13] should be used in comparison with the *CCC*. Generally, the corresponding areas of the folded butterfly using these layout schemes [2, 8, 20] are at least doubled.

4 Lower Bound on Layout Area

To prove the optimality of the new layout, it is desirable to have a tight lower bound, say, $\frac{1}{2}n^2 - o(n^2)$, so that we can conclude that the deviation of the new layout from optimality is at worst of some lower order than a constant factor. While such a tight lower bound is difficult to derive, we give below a lower bound of $(\frac{1}{2}n - 1)^2$ for *CCC* _{n} . Given this bound, we can see that the deviation of the new layout is at worst of a small additive factor of $\frac{1}{2}$ or a multiplicative factor of 2 from optimality.

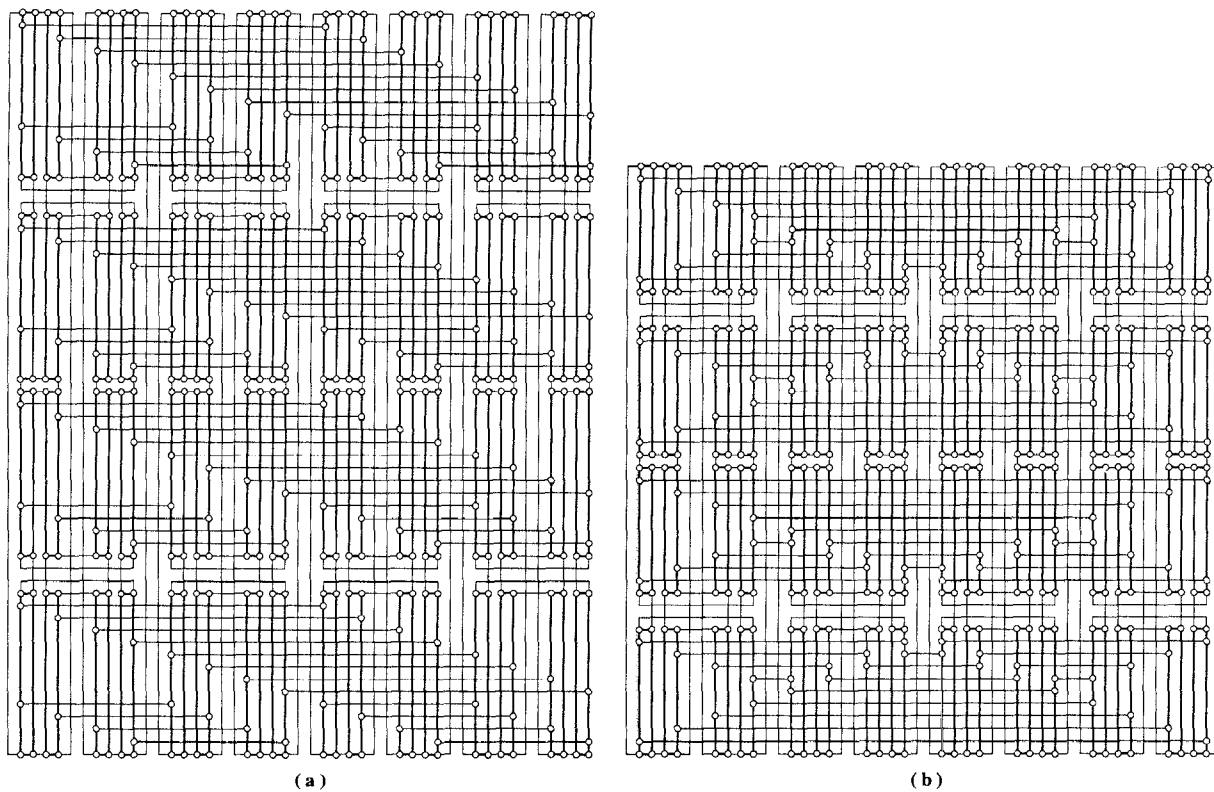


Figure 4: (a) New recursively-structured layout of 6-dim CCC with area 48×60 . (b) Improved new layout of 6-dim CCC with area 48×48 .

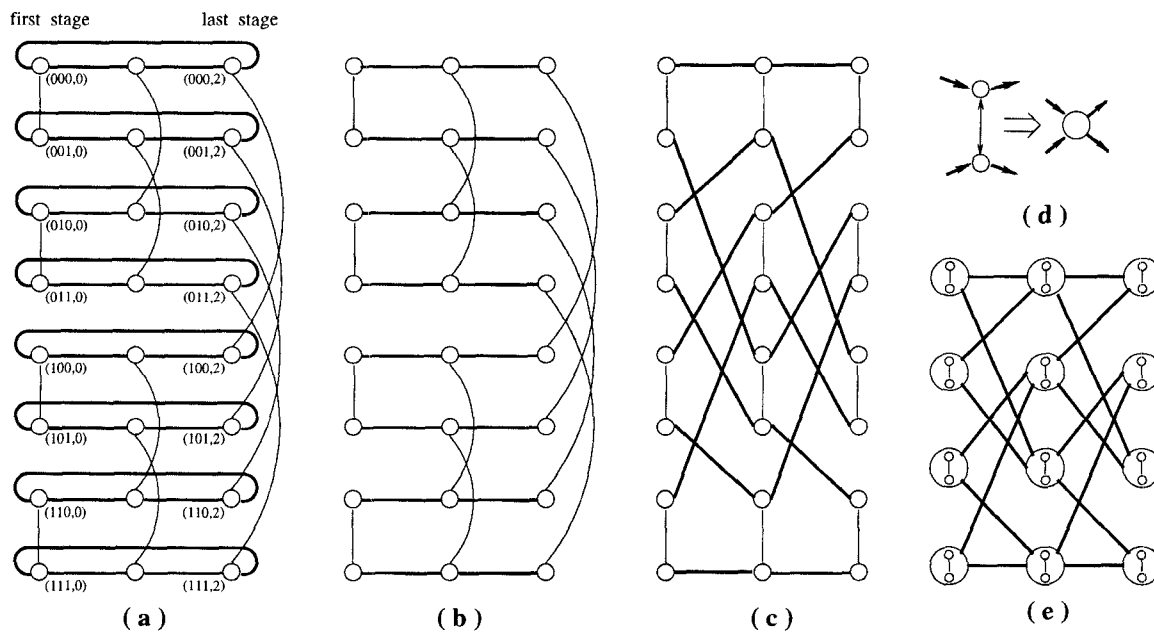


Figure 5: An embedding of $\mathcal{K}_{8,8}$ into CCC_n .

The lower bound $(\frac{1}{2}n - 1)^2$ is easily seen from the bounding strategy invented in [19] which is in terms of the bisection width of a graph. We present it below as Lemma 1.

Lemma 1 For any graph \mathcal{G} with bisection width $BW(\mathcal{G})$, $AREA(\mathcal{G}) \geq (BW(\mathcal{G}) - 1)^2$.

Proof: See [19]. □

The proof of the bisection width $\frac{1}{2}n$ of CCC_n , however, is complicated. Alternatively, we can use the modified bounding strategy, Lemma 2, from [2] where a lower bound of the butterfly network layout is proved by the same technique in terms of the minimum special bisection width.

Let \mathcal{G} be a graph having a designated set of $2c > 0$ special nodes. The minimum special bisection width of \mathcal{G} , denoted $MSBW(\mathcal{G})$, is the smallest number of edges whose removal partitions \mathcal{G} into two disjoint subgraphs, each containing half of \mathcal{G} 's special nodes.

The following three lemmas are due to Avior *et al.* [2].

Lemma 2 For any graph \mathcal{G} , $AREA(\mathcal{G}) \geq (MSBW(\mathcal{G}) - 1)^2$.

In order to bound the $MSBW$ of CCC_n , we employ the congestion argument originated in [12, 13] and refined in [2] which is used for bounding unknown $MSBW$'s from known ones.

Lemma 3 Let \mathcal{G} and \mathcal{H} be graphs having equal numbers of special nodes. If there is an embedding of \mathcal{G} into \mathcal{H} which maps special nodes to special nodes and which has congestion $\leq C$, then

$$MSBW(\mathcal{H}) \geq (1/C)MSBW(\mathcal{G}).$$

The complete bipartite graph $\mathcal{K}_{n,n}$ plays the role of the guest graph \mathcal{G} with known $MSBW = \frac{1}{2}n^2$.

Lemma 4 $MSBW(\mathcal{K}_{n,n}) = \frac{1}{2}n^2$ when all nodes of $\mathcal{K}_{n,n}$ are special.

Now we give an embedding of the guest graph $\mathcal{K}_{n,n}$ into the host graph CCC_n .

Lemma 5 One can embed $\mathcal{K}_{n,n}$ into CCC_n with congestion $2^s = n$ in such a way that the inputs and outputs of $\mathcal{K}_{n,n}$ map, respectively, to the first stage and the last stage of CCC_n .

Proof: Consider the embedding of $\mathcal{K}_{n,n}$ into the CCC_n which assigns the inputs of $\mathcal{K}_{n,n}$ to the first stage of CCC_n and the outputs of $\mathcal{K}_{n,n}$ to the last stage of CCC_n , and which routes the edges of $\mathcal{K}_{n,n}$ in increasing order of dimensions, *i.e.*, from right to left.

With no loss of generality, see Fig. 5 for an embedding of $\mathcal{K}_{8,8}$ into CCC_8 . Since the long wrap-around cyclic edges of CCC are not used for routing in the embedding, Fig. 5(a) is simplified to Fig. 5(b). Fig. 5(b) can be isomorphically arranged as Fig. 5(c) in which all stages of nodes except the first stage are reordered

so that pairs of nodes connected by cubical edges are put together like the first stage while the cyclic edges at each stage appear as the unshuffle-connection pattern [1, 7, 9]. Fig. 5(c) can be contracted to Fig. 5(e) by squeezing every pair of nodes into one big node shown in Fig. 5(d). Fig. 5(e) is apparently a reverse Omega network (or a flip network [3]). Hence the original CCC_n is turned into a reverse Omega network with $\frac{1}{2}n$ inputs and $\frac{1}{2}n$ outputs.

Note that the reverse Omega network (with $\frac{1}{2}n$ inputs and $\frac{1}{2}n$ outputs) has the banyan property [10]: each input node u is connected to each output node v by exactly one path of length $s - 1$. Let e be a stage- k edge of the reverse Omega network where $0 \leq k \leq s - 2$. One end-point of e reaches precisely 2^{s-k-2} distinct output nodes while the other end-point of e reaches precisely 2^k distinct input nodes. Hence edge e lies on precisely 2^{s-2} input-output paths. Since each input or output contains two nodes of $\mathcal{K}_{n,n}$, edge e actually lies on precisely 2^s input-output paths, *i.e.*, its congestion is $2^s = n$.

Further investigation shows that the congestion of the cubical edges of CCC_m which are shown as thin edges in Fig. 5(d), is also $2^s = n$, since from each input, exactly half of the paths will go through the cubical edges. □

Lemma 6 $MSBW(CCC_n) \geq \frac{1}{2}n$.

Proof: Directly from Lemmas 3, 4, and 5. □

Finally, Lemma 6 can be combined with Lemma 2 to yield the desired lower bound, Theorem 1, on the area of the layout of CCC_n .

Theorem 1 Any layout of CCC_n has area at least $(\frac{1}{2}n - 1)^2$.

5 Conclusion

The motivation underlying the work presented here is the question of how much we can reduce the layout area of the CCC . We have given a simple, regular and more compact layout scheme for the CCC ; the resulting area is $\frac{1}{2}n^2 + o(n^2)$. Some earlier attempts have been made: [17] gave a construction of the CCC with area n^2 ; [6] gave one with area $\frac{3}{4}n^2$.

We also give a lower bound on the layout area of the CCC by which we can judge how far our new layout may be from optimality. There is still a gap of a constant additive factor of $\frac{1}{2}$ between the new layout and the lower bound. To narrow or fill the gap, we need to find either a more compact layout or a tighter lower bound for the CCC . We conjecture that the new layout scheme is minimal. Hence, our future effort will be devoted mainly to finding a tighter lower bound.

Another important measure of a layout is the maximum wire length [4, 11]. Another future research item will be to find layout schemes that will not produce long wires, and to consider the tradeoff between area and maximum wire length for CCC layouts.

Acknowledgement

We are grateful to the anonymous referees for their constructive comments.

References

- [1] F. Annexstein, M. Baumslag, and A. L. Rosenberg. Group action graphs and parallel architecture. *SIAM J. Computing*, 19(3):544–569, June 1990.
- [2] A. Avior, T. Calamoneri, S. Even, A. Litman, and A. L. Rosenberg. A tight layout of the butterfly network. In *Proceedings of 8th ACM Symposium on Parallel Algorithms and Architectures*, pages 170–175, 1996.
- [3] K. E. Batchler. The flip network in STARAN. In *Proceedings of International Conference on Parallel Processing*, pages 65–71, Detroit, MI, 1976.
- [4] R. Beigel and C. P. Kruskal. Processor networks and interconnection networks without long wires. In *ACM Symposium on Parallel Algorithm and Architecture*, pages 42–51, 1989.
- [5] S. N. Bhatt and F. T. Leighton. A framework for solving VLSI graph layout problem. *Journal of Computer and System Sciences*, 28(2):300–343, 1984.
- [6] S. Bhattacharya, C. T. Liang, and W. T. Tsai. Cubical bus connected columns: An alternative to hypercube. In *Proceedings of the Supercomputing Symposium*, pages 409–420, 1991.
- [7] G. Chen and F. C. M. Lau. Comments on “a new family of Cayley graph interconnection networks of constant degree four”. *IEEE Transactions on Parallel and Distributed Systems*, to appear.
- [8] Y. Dinitz. A compact layout of butterfly on the square grid. Technical Report 873, Technion-Israel Institute of Technology, Haifa 32000, Israel, November 1995.
- [9] B. N. Jain. Equivalence between cube-connected cycles networks and circular shuffle networks. In *Proceedings of International Conference on Parallel Processing*, pages 8–11, 1986.
- [10] C. P. Kruskal and M. Snir. A unified theory of interconnection network structure. *Theoretical Computer Science*, 48(1):75–94, 1986.
- [11] F. C. M. Lau and G. Chen. Optimal layouts of midmew networks. *IEEE Transactions on Parallel and Distributed Systems*, 7(9):954–961, September 1996.
- [12] F. T. Leighton. *Complexity Issues in VLSI: Optimal Layout for the Shuffle-Exchange Graph and Other Networks*. The MIT Press, 1983.
- [13] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercube*. Morgan Kaufmann Publishers, 1992.
- [14] C. Mead and L. Conway. *Introduction to VLSI Systems*. Addison Wesley, 1980.
- [15] K. Mehlhorn, F. P. Preparata, and M. Sarrafzadeh. Channel routing in knock-knee mode: Simplified algorithms and proofs. *Algorithmica*, 1:213–221, 1986.
- [16] F. P. Preparata and J. Vuillemin. The cube-connected cycles: A versatile network for parallel computation. *CACM*, 24(5):300–309, May 1981.
- [17] J. J. Shen and I. Koren. Yield enhancement designs for wsi cube connected cycles. In *Proceedings of IEEE International Conference on Wafer Scale Integration*, pages 289–298, 1989.
- [18] C. D. Thompson. Area-time complexity for VLSI. In *Proc. 11th Ann. Symp. on Theory of Computing*, pages 81–88, Atlanta, GA, May 1979.
- [19] C. D. Thompson. *A Complexity Theory for VLSI*. PhD thesis, CMU Computer Science Department, 1980.
- [20] D. S. Wise. Compact layouts of Banyan/FFT networks. In H. T. Kung, R. Sproull, and G. Steele, editors, *VLSI Systems and Computations*, pages 186–195. Springer-Verlag, 1981.