

# Tighter Layouts of the Cube-Connected Cycles

Guihai Chen and Francis C.M. Lau, *Member, IEEE*

**Abstract**—Preparata and Vuillemin proposed the cube-connected cycles (*CCC*) and its compact layout in 1981 [17]. We give a new layout of the *CCC* which uses less than half the area of the Preparata-Vuillemin layout. We also give a lower bound on the layout area of the *CCC*. The area of the new layout deviates from this bound by a small constant factor. If we “unfold” the cycles in the *CCC*, the resulting structure can be laid out in optimal area.

**Index Terms**—Interconnection networks, cube-connected cycles, VLSI, embedding, routing, layout.

## 1 INTRODUCTION

INTERCONNECTION network is a key component of a parallel computer. Many issues need to be considered when deciding on a specific topology for connecting a set of processors. Given the rapid technological advances in VLSI, it is reasonable to conceive of a huge number of processors being integrated tightly together to solve problems in a cooperative, parallel fashion. Therefore, one of the criteria to judge the suitability of an interconnection network for the implementation of parallel computers is whether the network can be laid out compactly in a VLSI grid.

The cube-connected cycles (*CCC*), one of the most extensively studied and frequently cited interconnection networks, was proposed by Preparata and Vuillemin [17] as a substitute for the hypercube in 1981. In the same paper, they gave an asymptotically optimal layout scheme for the *CCC*. Their layout scheme, however, cannot produce the minimal layout for the *CCC*. Our work aims at finding better layout schemes for the *CCC*. Research in the fields of graph embedding and VLSI layout has developed powerful techniques [2], [5] that can produce embeddings and layouts which are quite efficient—often within a constant factor from the optimal. However, even a modest constant factor may render an asymptotically optimal layout or embedding unacceptable for real implementation. It is necessary to try to achieve the minimal. This is the motivation behind our work.

Our project has two goals: 1) to give a more compact layout of the *CCC* than the Preparata-Vuillemin layout, and 2) to reduce the long wires of the layout while keeping the asymptotically optimal area. We have achieved the first goal—a new layout scheme which uses less than half the area of the Preparata-Vuillemin layout. Section 2 reviews the Preparata-Vuillemin layout. Section 3 presents the new

layout and compares it with the Preparata-Vuillemin layout. Section 4 gives a lower-bound on the layout area. The Appendix presents a layout of the “unfolded” version of the *CCC*—called the *cube-connected lines (CCL)*.

## 2 PRELIMINARIES

We assume the VLSI model by Thompson [18], [19]. In our constructions, no knock-knees are allowed—that is, two wires cannot turn at the same grid point [15], [16].

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 the layout uses to contain all the nodes and all the path segments.

### 2.1 Cube-Connected Cycles

The  $s$ -dimensional ( $s$ -d for short) cube-connected cycles (*CCC*) is constructed from the  $s$ -dimensional hypercube by replacing each node of the hypercube with a cycle of  $s$  nodes [14], [17]. The  $i$ th-d edge of a node of the hypercube is then connected to the  $i$ th node of the corresponding cycle of the *CCC*. For example, see Figs. 1a and 1b. The resulting graph has  $s \cdot 2^s$  nodes, each of degree 3. By extending the labeling scheme of the hypercube, we can represent each node of the *CCC* by  $\langle w, i \rangle$  where  $i$  ( $1 \leq i \leq s$ ) is the position of the node within its cycle and  $w$  (an  $s$ -bit binary string with the first-d at the rightmost) is the label of the node in the hypercube that corresponds to the cycle. 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 bit.

Edges of kind (1) are cycle-edges and edges of kind (2) are cube-edges. As shown in Fig. 1c, the *CCC* is often drawn in the multistage format which will directly give rise to the Preparata-Vuillemin layout. The first and the last stage,

- G. Chen is with the State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210024, China. E-mail: gchen@nju.edu.cn.
- F.C.M. Lau is with the Department of Computer Science and Information Systems, University of Hong Kong, Pokfulam Rd., Hong Kong. E-mail: fclau@csis.hku.hk.

Manuscript received 6 May 1997; accepted 13 Dec. 1999.

For information on obtaining reprints of this article, please send e-mail to: tpd@computer.org, and reference IEEECS Log Number 105005.

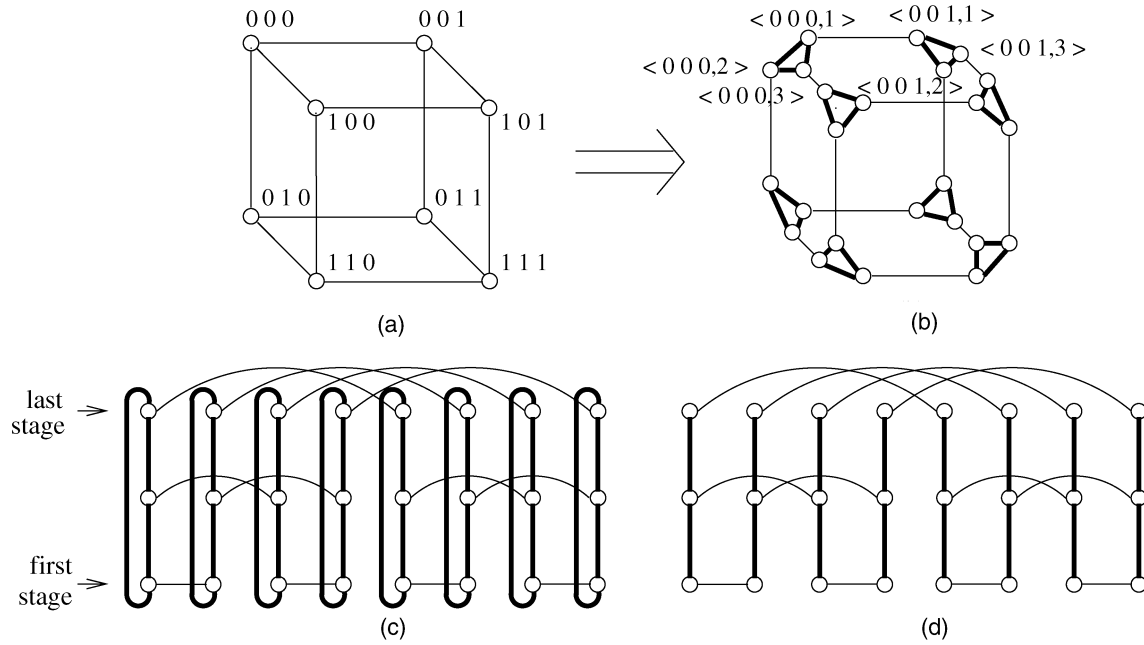


Fig. 1. (a) A 3-d hypercube. (b) A 3-d CCC. (c) Another drawing of a 3-d CCC: Cycle-edges in thick lines and cube-edges in thin lines. (d) A 3-d CCL.

stages 1 and  $s$ , are called the two end stages, and they consist of all the nodes  $\langle w, i \rangle$  for  $i = 1$  and  $i = s$ , respectively.

The CCC is closely related to the butterfly network, just as the shuffle-exchange network is to the deBruijn network. The group-theoretic relations of the four networks are well studied in [1] where the CCC and the butterfly are proven to be Cayley graphs derivable from the shuffle-exchange network and the deBruijn network, respectively; and inversely, the shuffle-exchange network and the deBruijn network are proven to be some coset graph of the CCC and the butterfly network, respectively. Feldmann and Unger proved that the CCC is a subgraph of the butterfly network, and the shuffle-exchange network is a subgraph of the deBruijn network [10].

We introduce in passing an unfolded version of the CCC. Like the butterfly network, the CCC now has the traditional folded version and the new unfolded version. For the unfolded CCC, Condition 1 in the above definition is changed to

1.  $w = w'$  and  $i - i' = \pm 1$ .

Each cycle of the CCC is replaced by a line in the unfolded CCC. We therefore call the unfolded CCC the *cube-connected lines*, denoted by CCL hereafter. A 3-d CCL is shown in Fig. 1d. We present the layout of the CCL in the Appendix.

## 2.2 The Preparata-Vuillemin Layout

Fig. 2b shows the Preparata-Vuillemin layout of a 4-d CCC, which is recursively constructed from two 3-d CCCs (identified by the dotted lines). Based on the recursive construction, it easily can be proven 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  $s \approx \log(N/\log N)$ , the chip size is  $O((N/\log N)^2)$ . In gen-

eral, we say that a network of  $N$  nodes has asymptotically-optimal layout if it can be laid out in area  $O(N^2/T^2)$ , where  $T$  is the time to execute an ascend-descend algorithm [4], [19]. CCC can execute an ascend-descend algorithm in time  $O(\log N)$  [17]. Therefore, the Preparata-Vuillemin layout is asymptotically optimal.

In more detail, for an  $s$ -d CCC with  $n = 2^s$  cycles, denoted by  $CCC(n)$  hereafter, 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. Then, for the Preparata-Vuillemin layout,

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

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,  $W(s) \times H(s)$ , is

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

For the “more economical” Preparata-Vuillemin layout which is shown in Fig. 2c,

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

The saving in the number of horizontal tracks for the case of even  $s$  comes from the overlapping of some of the  $s$ th-d

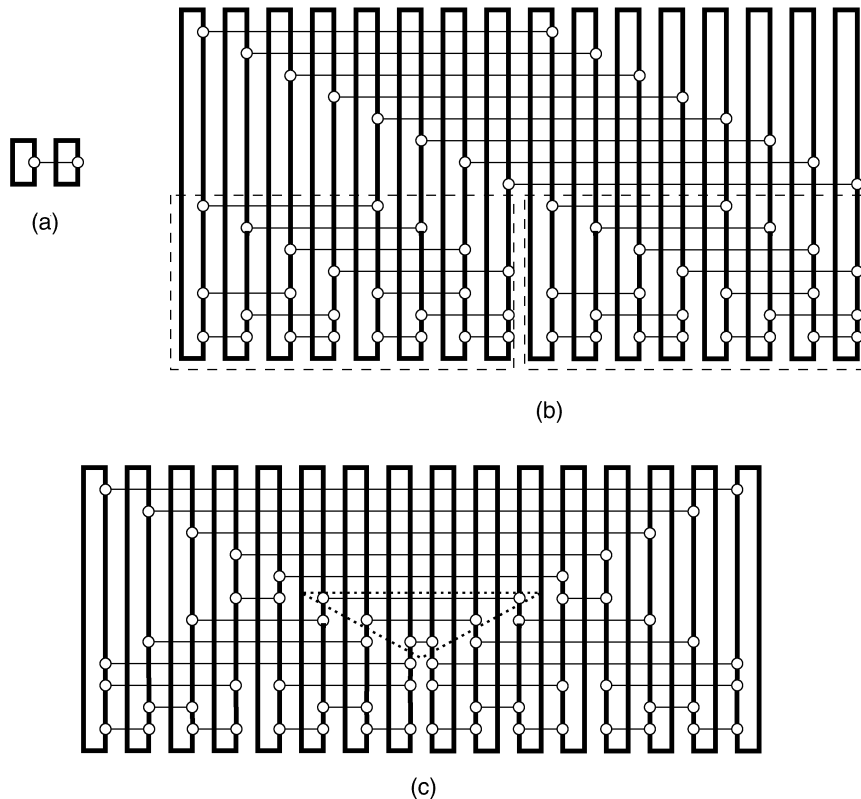


Fig. 2. (a) Preparata-Vuillemin layout of the 1D *CCC*—the base case. (b) Preparata-Vuillemin layout of the 4-d *CCC*. (c) More economical Preparata-Vuillemin layout of the 4-d *CCC*.

tracks with the embedded layouts for the  $(s - 1)$ -d *CCC* (see the dotted region in Fig. 2c). From the above, we get  $W(s) = 2^{s+1} = 2n$  and

$$\begin{aligned} H(s) &= 3 + (2 + 4 + 5 + \dots + 2^{s-2} + (2^{s-2} + 1)) \\ &= \frac{2}{3}2^s + \frac{1}{2}s + \frac{4}{3} \end{aligned}$$

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 *CCC* is asymptotically optimal, it is not the minimal layout. For real implementations, we would prefer using as tight a layout as possible. Here we give a new layout for the *CCC*. It is more compact than the Preparata-Vuillemin layout; whether it is minimal is an open question.

Referring to Fig. 2 again, there are two obvious shortcomings in the Preparata-Vuillemin layout:

- It does not try to make use of the corner positions of a cycle by putting some nodes there, and
- It places all the cycles along the same horizontal axis.

In the new layout, these two problems are corrected, and the resulting layout uses less area and has a better aspect ratio.

#### 3.1 Small *CCCs*

Figs. 3a, 3b, and 3c show the layouts of the first three *CCCs*, starting from the second dimension. These layouts use minimal areas. As our interest is in the layout of the general *CCC*, we omit the proofs of these specific cases here. Like the Preparata-Vuillemin layout, the new layout is based on recursive construction. Unlike the Preparata-Vuillemin layout, for which the recursion begins at the first dimension, the base case for recursion in the new layout is the 4-d *CCC* (Fig. 3c). The reason for this is that the layout of the 4-d *CCC* is the first one (starting from the first dimension) that puts a node in every corner of a cycle. This layout is correct in the sense that it is indeed a valid *CCC* that is being laid out. This can be easily verified by examining the connections against the labels of the cycles in Fig. 3c; similarly for the smaller cases.

#### 3.2 Recursive Construction

The procedure is as follows, for  $s \geq 5$ .

Take two copies of the layout for the  $(s - 1)$ -d *CCC*; place them side by side. Stretch every cycle vertically by an extra height of  $2^{s-3}$  for the embedding of the  $s$ -th nodes and edges.

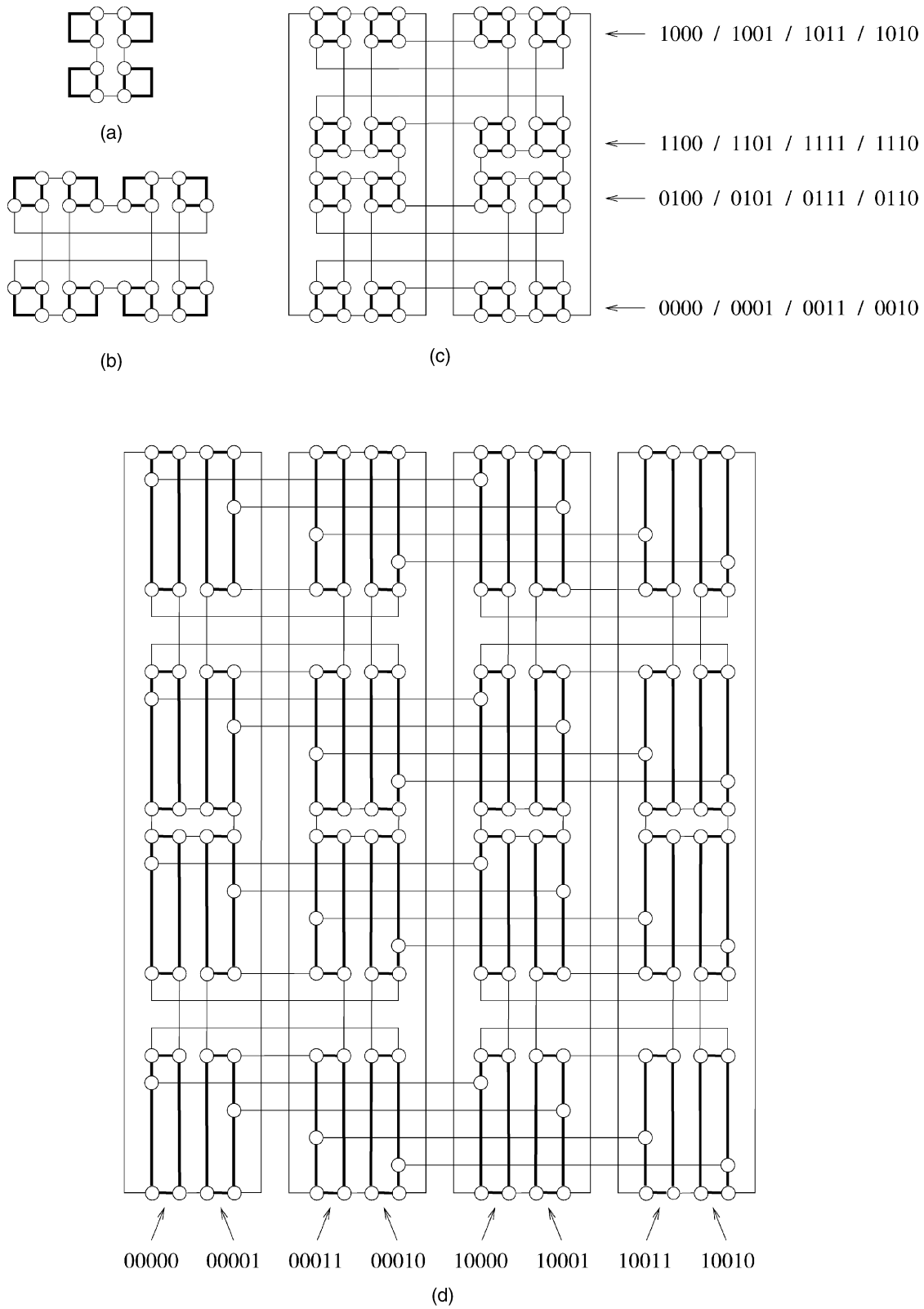


Fig. 3. New layouts of small CCC's: (a) A 2-d CCC uses area  $4 \times 4$ ; (b) a 3-d CCC uses area  $8 \times 6$ ; (c) a 4-d CCC uses area  $12 \times 12$ ; (d) a 5-d CCC uses area  $24 \times 28$ .

Since there are four rows of cycles from top to bottom, a total of  $2^{s-1}$  extra horizontal tracks are added. Note how the  $s$ th-d nodes and edges are embedded (refer to Fig. 3d and

Fig. 4) within these extra tracks: One node is added to every cycle, and its corresponding node in the other copy of the layout of the  $(s - 1)$ -d CCC is placed at the same horizontal

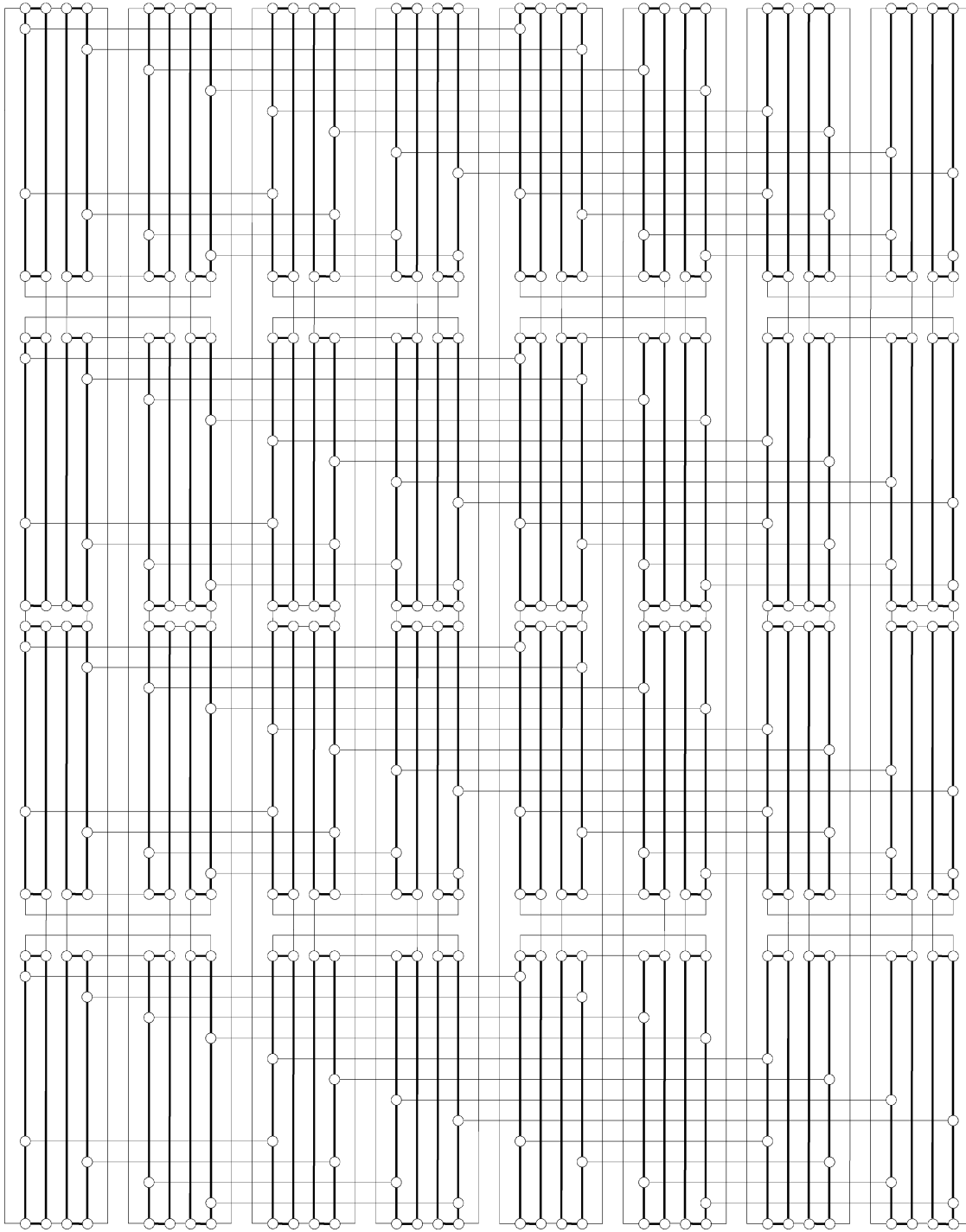


Fig. 4. New layout of the 6-d CCC with area  $48 \times 60$ .

position, and the two are joined by a horizontal wire. We label the cycles in the left copy by extending the original labels by a 0 on the left, and the cycles in the right copy by a 1 on the left. The correctness of the  $s$ -d layout immediately follows from this labeling scheme. In Fig. 3d, which is recursively constructed from Fig. 3c, the labels of the bottom row of cycles are shown.

Using the procedure, the layout of the 6-d CCC can be constructed easily. The result is shown in Fig. 4.

For an  $s$ -d CCC with  $n = 2^s$  cycles,

$$W(4) = 12,$$

$$H(4) = 12,$$

$$W(s) = 2W(s-1),$$

$$H(s) = H(s-1) + 2^{s-1}.$$

We get  $W(s) = 12 \times 2^{s-4} = \frac{3}{4}n$  and

$$H(s) = 2^s - 4 = n - 4.$$

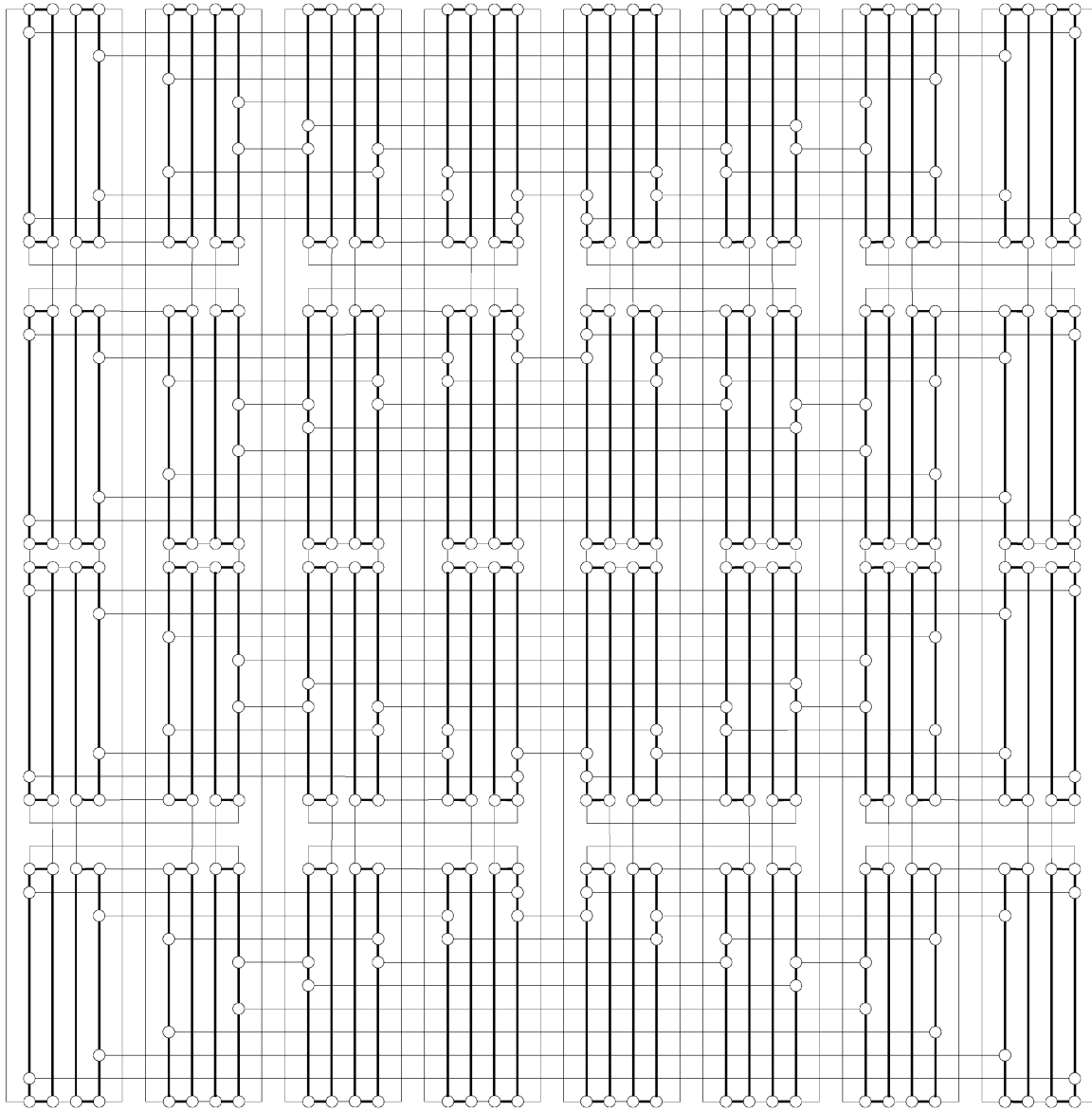


Fig. 5. More economical new layout of the 6-d CCC with area  $48 \times 48$ .

Hence, the area,  $W(s) \times H(s)$ , is

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

In the same way that a more economical layout can be derived from the Preparata-Vuillemin layout, the new layout has a more economical version. The more economical version for the 6-d CCC is shown in Fig. 5. For this improved layout,

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

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

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

### 3.3 Comparison

It is worth noting that although the construction strategy used in the new layout and that in the Preparata-Vuillemin layout are very different, the two layouts have the same recursive formulae for  $W(s)$  and  $H(s)$ , and almost the same recursive formulae for their more economical versions. The new layout, however, is based on a much better base case— $W(4) = H(4) = 12$  or  $W(4) \times H(4) = 144$ ; whereas the area using the Preparata-Vuillemin layout for the same size is  $32 \times 17 = 544$  or  $32 \times 14 = 448$  for the more economical version. As a result, the new layout has a smaller constant in front of the dominant term in its area formulae.

By ignoring the low-order terms in (1), (2), (3), and (4), the four layout schemes of the  $CCC(n)$  that were discussed in

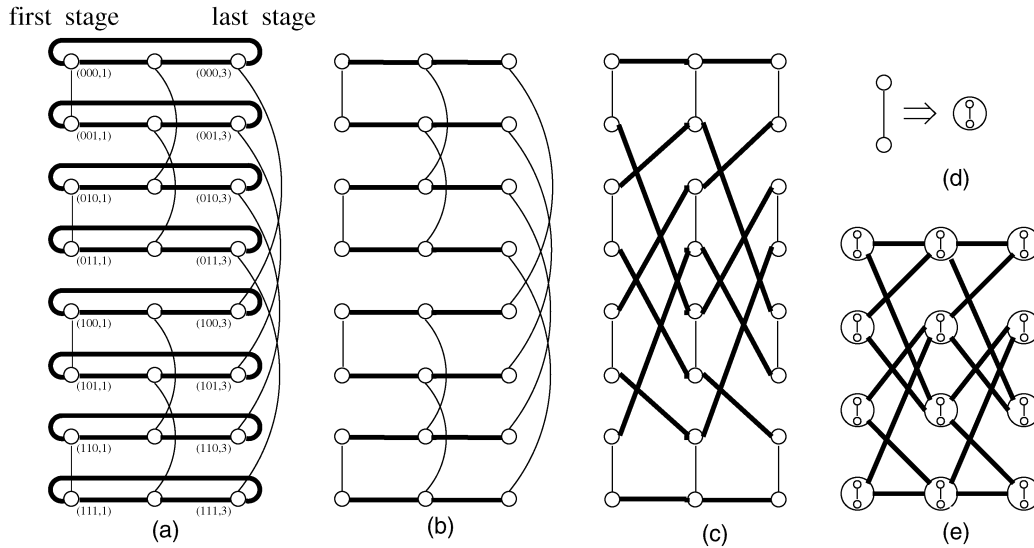


Fig. 6. An embedding of  $K_{8,8}$  into the  $CCC(n)$ .

Sections 2.2 and 3.2 take areas of approximately  $2n^2$ ,  $\frac{4}{3}n^2$ ,  $\frac{3}{4}n^2$ , and  $\frac{1}{2}n^2$ , respectively. We compare the new layout with the Preparata-Vuillemin layout, and the more economical version of the new layout with the more economical version of the Preparata-Vuillemin layout. In either case, the new layout scheme uses less than half the area of the Preparata-Vuillemin layout. The other important advantage of the new layout is that it has a more practical aspect ratio ( $W(s)/H(s)$ ), which is close to 1, whereas the aspect ratio of the Preparata-Vuillemin layout could be as large as 3. Because of a better aspect ratio, the new layout has a shorter maximum wire length than the Preparata-Vuillemin layout.

The new layout also shows the superiority of the  $CCC$  in layout area over other hypercube substitutes such as the shuffle-exchange network and the butterfly network [14]. The optimal layout of the shuffle-exchange network was due to Leighton [13]. His layout of the shuffle-exchange network, as well as the other related ones, however, is complicated, not regular or recursive. For years, the best known layout of the butterfly network with  $n$  inputs or outputs was that by Wise [20], which has area  $\simeq 2n^2$ . Recently, more compact layouts for the butterfly were found with area  $\simeq \frac{11}{6}n^2$  [9], or  $n^2 + o(n^2)$  [2]. The butterfly networks discussed in all these papers, however, are unfolded. To be fair, the folded butterfly network (i.e., the first and the last stage are merged) [10], [14] should be considered when comparing with the  $CCC$ . The corresponding areas of the folded butterfly given in [2], [9], [20] would then need to be doubled or quadrupled. On the other hand, as can be seen in the Appendix, the unfolded  $CCC$  can be laid out with area  $\simeq \frac{1}{4}n^2$ .

#### 4 LOWER BOUND ON LAYOUT AREA

We give below a lower bound of  $(\frac{1}{2}n - 1)^2$  on the layout area for the  $CCC(n)$ . Our layout of the  $CCC$  as presented in the previous sections deviates from this bound by a factor of 2.

The following construction does not take into account the cycles in the  $CCC$  (see Fig. 6). Each cycle, in fact, is treated as a line, and hence, the lower bound is also valid for the  $CCL$ . As is shown in the Appendix, the  $CCL$  can be laid out in an area of  $(\frac{1}{2}n + o(n))^2$ , which is tight when compared with the lower bound.

The lower bound of  $(\frac{1}{2}n - 1)^2$  can be easily seen from the bounding strategy invented in [19], which is in terms of the bisection width of a graph.

**Lemma 1.** For any graph  $\mathcal{G}$  with bisection width  $BW(\mathcal{G})$ ,  $AREA(\mathcal{G}) \geq (BW(\mathcal{G}) - 1)^2$  [19].

The proof of the bisection width,  $\frac{1}{2}n$ , of the  $CCC(n)$ , however, is complicated. We therefore turn to the modified bounding strategy introduced in [2], which uses something called the *minimum special bisection width*. Using this strategy, the authors of [2] were able to derive a tight lower bound for the butterfly network layout.

Let  $\mathcal{G}$  be a graph having a designated set of 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]. They used a congestion argument originated in [13], [14], which can be applied to bound unknown  $MSBW$ 's with known ones.

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

**Lemma 3.** *Let  $\mathcal{G}$  and  $\mathcal{H}$  be graphs having equal numbers of special nodes [2]. 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*

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

**Lemma 4.** *For the complete bipartite graph  $\mathcal{K}_{n,n}$ ,  $MSBW(\mathcal{K}_{n,n}) = \frac{1}{2}n^2$  when all nodes of  $\mathcal{K}_{n,n}$  are special [2].*

Now given the complete bipartite graph  $\mathcal{K}_{n,n}$  which has a known  $MSBW$ , if we could embed it into the  $CCC(n)$ , then by Lemma 3, we will have the  $MSBW$  of the  $CCC(n)$ . The next lemma gives such an embedding.

**Lemma 5.**  *$\mathcal{K}_{n,n}$  can be embedded into the  $CCC(n)$  with congestion  $2^s = n$ .*

**Proof.** The embedding of  $\mathcal{K}_{n,n}$  into  $CCC(n)$  is such that the inputs of  $\mathcal{K}_{n,n}$  are mapped to the first stage (Stage 1) of the  $CCC(n)$  and outputs of  $\mathcal{K}_{n,n}$  are mapped to the last stage (Stage  $s$ ) of the  $CCC(n)$ . The edges of  $\mathcal{K}_{n,n}$  are mapped to various paths that go from some first-stage node to some last-stage node in the  $CCC(n)$ —i.e., from left to right in Fig. 6a. Hence, the special nodes of the  $CCC(n)$  are all the nodes in the first and the last stage.

Without loss of generality, Fig. 6 shows an embedding of  $\mathcal{K}_{8,8}$  into a  $CCC(8)$ . The  $CCC(8)$  has two columns of eight nodes each, which correspond respectively to the first stage and the last stage of nodes in Fig. 6e. Since the long wrap-around cycle-edges of the  $CCC$  are not used for routing in the embedding, Fig. 6a is simplified as Fig. 6b, which is actually a  $CCL(8)$ . Fig. 6b can be isomorphically arranged to become Fig. 6c, in which all stages of nodes, except the first stage, are reordered so that a pair of nodes connected by a cube-edge are placed together, just like those in the first stage. As a result, the cycle-edges at each stage would be in an unshuffle-connection pattern [1], [8]. Fig. 6c can be transformed into Fig. 6e by replacing every pair of nodes by a complex node, as shown in Fig. 6d. Fig. 6e is a reverse omega network (or a flip network [3]). Hence, we have transformed the original  $CCC(n)$  into a reverse omega network with  $\frac{1}{2}n$  inputs and  $\frac{1}{2}n$  outputs.

The reverse omega network (with  $\frac{1}{2}n$  inputs and  $\frac{1}{2}n$  outputs) has the banyan property [11]: 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  $1 \leq k \leq s - 1$ . One end point of  $e$  reaches precisely  $2^{s-k-1}$  distinct output nodes while the other end point of  $e$  reaches precisely  $2^{k-1}$  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  $e$  lies on precisely  $2^s$  input-output paths—i.e., its congestion is  $2^s = n$ .

A further look reveals that the congestion of cube-edges of the  $CCC(n)$ , shown as a thin edge in Fig. 6d, is also  $2^s = n$ , since from each input, exactly half of

the paths will go through the cube-edge of a complex node.  $\square$

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

**Proof.** Directly from Lemmas 3, 4, and 5.  $\square$

Combining Lemma 6 and Lemma 2 yields the desired lower bound on the area of  $CCC(n)$  layouts:

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

## 5 CONCLUSION

We have given a simple, regular, and more compact layout scheme for the  $CCC$ , which takes less than half of the area of the Preparata-Vuillemin layout. We have also derived a lower bound on the layout area of the  $CCC$ . Our layout deviates from the lower bound by a constant factor of 2. The lower bound, however, is not the tightest possible because its construction does not take into account the laying out of the cycles in a  $CCC$ . It is, more appropriately, a lower bound for the  $CCL$ . On the other hand, our tight layout of the  $CCL$  can give rise to a layout of the  $CCC$ , but the area will be four times that of the  $CCL$  (consider Fig. 7, and the width and the height of the grid will need to be doubled to accommodate the cycles of the  $CCC$ ). We conjecture, therefore, that the layout of the  $CCC$  as we have proposed in this paper is optimal. Further work will be directed to deriving a lower bound for the  $CCC$  that would take the cycles into account.

Our layout of the  $CCL$  as given in the Appendix reveals the superiority of the  $CCL$  over the unfolded butterfly network, since the former takes only one-fourth of the layout area of the unfolded butterfly network [2]. Another merit of the  $CCL$  is that a  $CCC$  can be embedded into a  $CCL$  with congestion 2 and dilation 2 due to the well-known fact that a cycle can be embedded into a line with congestion 2 and dilation 2. Hence, the  $CCL$  can be a good substitute for the  $CCC$ , and can execute an ascend-descend algorithm with a small constant slowdown.

Another important measure of a layout is the maximum wire length [4], [12]. We have recently succeeded in deriving a layout of the  $CCC$  which has no long wires and yet preserves the asymptotic-optimality of the area [7]. Our next task is to consider the trade-off [4], [6] between area and maximum wire length for the  $CCC$  layouts.

## APPENDIX

### LAYOUT OF $CCL$

In this Appendix, we give a tight layout of the  $CCL$ . Avior et al. [2] gave a tight layout of the unfolded butterfly network with area  $(n + o(n))^2$ . We borrow their technique and apply it to the  $CCL$ , resulting in the desired tight layout of the  $CCL$  with area  $(\frac{1}{2}n + o(n))^2$ .

Using the Preparata-Vuillemin layout (Fig. 2b), if we replace all the cycles by lines, we can lay out a  $CCL(n)$  in an  $n \times (n - 1)$  grid because nodes can now be placed at the two ends of a line.

**Lemma 7.** *A  $CCL(n)$  can be laid out in an  $(n + o(n)) \times (n + o(n))$  grid.*



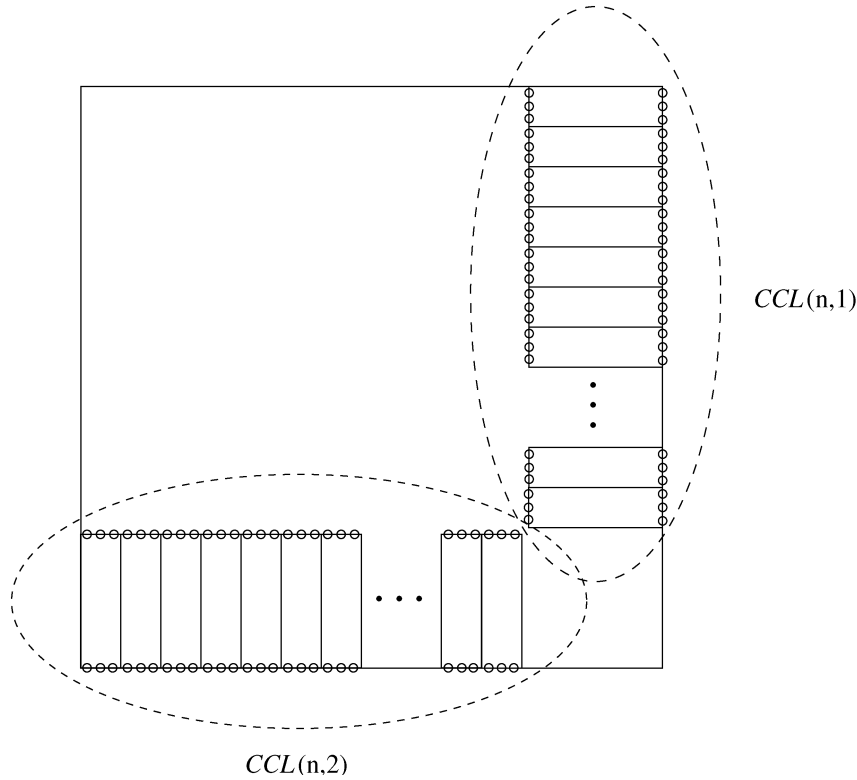


Fig. 7.  $CCL(n,1)$  and  $CCL(n,2)$  inside  $CCL(n)$ .

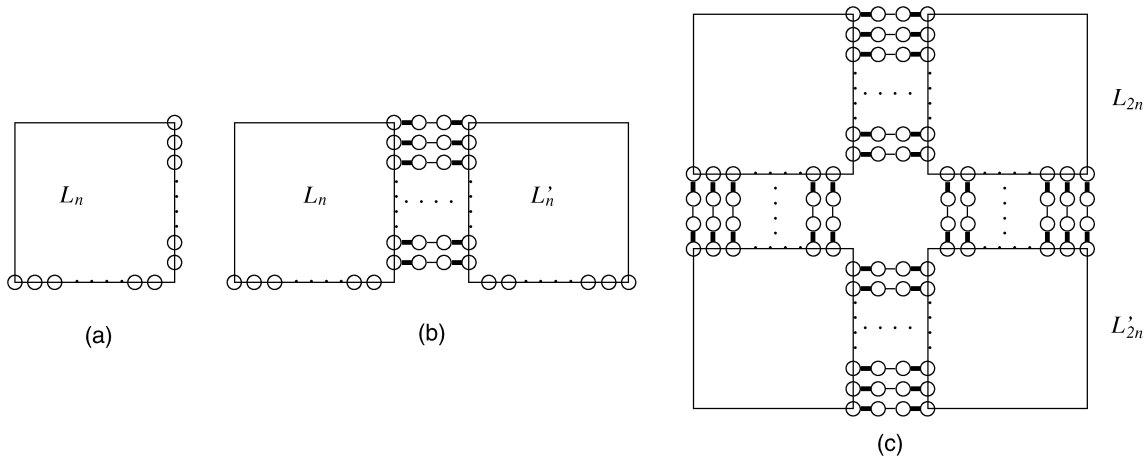


Fig. 8. (a)  $L_n$ . (b)  $L_{2n}$ . (c)  $L_{4n}$ .

**Proof.** Let the stages of a  $CCL(n)$  be numbered  $1, 2, \dots, s-1, s$ , where  $n = 2^s$ . Let  $k = s/2$ . By cutting the edges between Stage  $k$  and Stage  $k+1$ , we decompose the  $CCL(n)$  into

- $CCL(n,1)$ : the subgraph of  $CCL(n)$  bounded by Stage 1 and Stage  $k$ , and
- $CCL(n,2)$ : the subgraph of  $CCL(n)$  bounded by Stage  $k+1$  and Stage  $s$ .

Let  $L$  be a (Preparata-Vuillemin) layout of  $CCL(2^k)$  in a  $2^k \times (2^k - 1)$  grid. We stack  $2^k$  copies of  $L$ , one above another, along the right side of the grid for  $CCL(n)$ . This takes care of  $CCL(n,1)$  using a space of  $n \times (2^k - 1)$  (Fig. 7). By basic properties of the hypercube,  $CCL(n,2)$  is isomorphic to  $CCL(n,1)$  by a suitable relabeling of the

lines. We then do the same for  $CCL(n,2)$  along the bottom side of the grid as we just did to  $CCL(n,1)$ . It can be easily seen that the smallest grid for the  $CCL(n)$  that can hold both  $CCL(n,1)$  and  $CCL(n,2)$  is of area  $(n + o(n)) \times (n + o(n))$ .

We must then connect  $CCL(n,1)$  and  $CCL(n,2)$  to re-create the  $CCL(n)$ . This can be accomplished by routing a specific bijection between the two subgraphs of the  $CCL(n)$ . It is obvious that the unpopulated area that is left behind after placing the  $CCL(n,1)$  and  $CCL(n,2)$  is sufficient for any such bijection to be routed.  $\square$

Note that by a suitable orientation, the two end stages of nodes of the  $CCL(n)$  would occupy one horizontal side and one vertical side of the grid, as shown in Fig. 8a.

**Theorem 2.** *There is a layout of the  $CC\mathcal{L}(n)$  with area  $(\frac{1}{2}n + o(n))^2$ .*

**Proof.** We construct a layout of a  $CC\mathcal{L}(4n)$  from four copies of a layout of  $CC\mathcal{L}(n)$ . By Lemma 7, a  $CC\mathcal{L}(n)$  can be laid out in an  $(n + o(n)) \times (n + o(n))$  grid (Fig. 8a). We refer to this layout of the  $CC\mathcal{L}(n)$  as  $L_n$ . We flip  $L_n$  horizontally to produce  $L'_n$ . Using  $n$  extra nodes (in two columns), we can form  $CC\mathcal{L}(2n)$  from  $L_n$  and  $L'_n$  (Fig. 8b). We refer to this layout of the  $CC\mathcal{L}(2n)$  as  $L_{2n}$ .

Next, flip  $L_{2n}$  vertically to produce  $L'_{2n}$ . Add extra stages of nodes, and then join  $L_{2n}$  and  $L'_{2n}$  by connecting these stages of nodes to produce  $L_{4n}$ , a layout for  $CC\mathcal{L}(4n)$  (Fig. 8c). Clearly, layout  $L_{4n}$  resides in a  $(2n + o(n)) \times (2n + o(n))$  grid.  $\square$

## ACKNOWLEDGMENTS

We thank the editor and the anonymous referees for their helpful suggestions and comments. Guihai Chen's research was supported by the China National Science Foundation under grant 69803005.

## REFERENCES

- [1] F. Annexstein, M. Baumslag, and A.L. Rosenberg, "Group Action Graphs and Parallel Architecture," *SIAM J. Computing*, vol. 19, no. 3, pp. 544-569, 1990.
- [2] A. Avior, T. Calamoneri, S. Even, A. Litman, and A.L. Rosenberg, "A Tight Layout of the Butterfly Network," *Theory of Computing Systems*, vol. 31, no. 4, pp. 475-488, 1998.
- [3] K.E. Batcher, "The Flip Network in STARAN," *Proc. Int'l Conf. Parallel Processing*, pp. 65-71, 1976.
- [4] R. Beigel, C.P. Kruskal, "Processor Networks and Interconnection Networks without Long Wires," *Proc. ACM Symp. Parallel Algorithm and Architecture*, pp. 42-51, 1989.
- [5] S.N. Bhatt and F.T. Leighton, "A Framework for Solving VLSI Graph Layout Problem," *J. Computer and System Sciences*, vol. 28, no. 2, pp. 300-343, 1984.
- [6] N. Blum, "An Area-Maximum Edge Length Tradeoff for VLSI Layout," *Information and Control*, vol. 66, no. 1/2, pp. 45-52, 1984.
- [7] G. Chen and F.C.M. Lau, *Layout of CCC without Long Wires*, Technical Report 97-09, Dept. of Computer Science and Information Systems, University of Hong Kong, 1997.
- [8] G. Chen and F.C.M. Lau, "Comments on A New Family of Cayley Graph Interconnection Networks of Constant Degree Four," *IEEE Trans. Parallel and Distributed Systems*, vol. 8, no. 12, pp. 1,299-1,300, 1997.
- [9] Y. Dinitz, *A Compact Layout of Butterfly on the Square Grid*, Technical Report 873, Technion-Israel Inst. of Technology, Haifa, Israel, Nov. 1995.
- [10] R. Feldmann and W. Unger, "The Cube-Connected Cycles Network Is a Subgraph of the Butterfly Network," *Parallel Processing Letters*, vol. 2, no. 1, pp. 13-19, 1992.
- [11] C.P. Kruskal and M. Snir, "A Unified Theory of Interconnection Network Structure," *Theoretical Computer Science*, vol. 48, no. 1, pp. 75-94, 1986.
- [12] F.C.M. Lau and G. Chen, "Optimal Layouts of Midimew Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 7, no. 9, pp. 954-961, Sept. 1996.
- [13] F.T. Leighton, *Complexity Issues in VLSI: Optimal Layout for the Shuffle-Exchange Graph and Other Networks*. MIT Press, 1983.
- [14] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercube*. Morgan Kaufmann, 1992.
- [15] C. Mead and L. Conway, *Introduction to VLSI Systems*. Addison Wesley, 1980.
- [16] K. Mehlhorn, F. P. Preparata, and M. Sarrafzadeh, "Channel Routing in Knock-Knee Mode: Simplified Algorithms and Proofs," *Algorithmica*, vol. 1, pp. 213-221, 1986.
- [17] F.P. Preparata and J. Vuillemin, "The Cube-Connected Cycles: A Versatile Network for Parallel Computation," *Comm. ACM*, vol. 24, no. 5, pp. 300-309, 1981.
- [18] C.D. Thompson, "Area-Time Complexity for VLSI," *Proc. 11th Ann. Symp. Theory of Computing*, pp. 81-88, May 1979.
- [19] C.D. Thompson, *A Complexity Theory for VLSI*, PhD thesis, Carnegie Mellon Univ., Computer Science Dept., 1980.
- [20] D.S. Wise, "Compact Layouts of Banyan/FFT Networks," *VLSI Systems and Computations*, H.T. Kung, R. Sproull, and G. Steele, eds., pp. 186-195, Springer-Verlag, 1981.



internet computing, and parallel algorithms.



research interests are in parallel and distributed computing, object-oriented programming, and operating systems. Dr. Lau is a member of the IEEE and an active member of the IEEE Computer Society, where he has served as the 1999 vice president for Chapters Activities.

**Guihai Chen** received the BS degree from Nanjing University, China, in 1984, the ME degree from Southeast University, China, in 1987, and the PhD degree from the University of Hong Kong in 1997, all in computer science. In 1998, he visited Kyushu Institute of Technology, Japan, as a research fellow. He is currently an associate professor at Nanjing University. His main research interests include interconnection networks, advanced computer architecture,

**Francis C.M. Lau** received the BSc degree from Acadia University, Canada, and the MMath and PhD degrees from the University of Waterloo, Canada. He joined the Department of Computer Science and Information Systems at the University of Hong Kong in 1987, where he is now an associate professor and a leader of the Systems Research Group. One of the current projects of the group is to develop a Java-based platform for cluster computing. Dr. Lau's main