# COMPUTER SCIENCE PUBLICATION

OPTIMAL LAYOUT OF MIDIMEW NETWORKS

F.C.M. Lau and G. Chen

Technical Report TR-94-03

March 1994

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF ENGINEERING
UNIVERSITY OF HONG KONG
POKFULAM ROAD
HONG KONG

# Optimal Layout of Midimew Networks

F.C.M. Lau and G. Chen
Department of Computer Science
The University of Hong Kong
Hong Kong
{fcmlau,gchen}@csd.hku.hk

**Abstract**

Midimew networks [2] are mesh-connected networks derived from a subset of degree-4 circulant graphs. They have minimum diameter and average distance among all degree-4 circulant graphs, and are better than some of the most common topologies for parallel computers in terms of various cost measures. Among the many midimew networks, the rectangular ones appear to be most suitable for practical implementation. Unfortunately, with the normal way of laying out these networks on a 2-D plane, long cross wires that grow with the size of the network exist. In this paper, we propose ways to embed rectangular midimew networks in a 2-D grid so that the length of the longest wire is at most a small constant. The techniques we use are perfect shuffle and ring embedding onto a linear array. We prove that these constants are optimal under the assumption that rows and columns are moved as a whole as opposed to moving individual nodes during the embedding process.

*Keywords:* graph embedding, dilation, interconnection networks, mesh-connected computers, midimew networks.

## 1 Introduction

In designing interconnection networks, a constant low degree is generally more preferable than a high degree because of performance advantages, scalability, and implementation considerations [7, 17, 18]. Within the family of low-degree networks, degree-4 networks such as 2-D meshes and 2-D tori are among the most popular choices for processor interconnection in today's parallel computers [14, 16, 18]. The problem with these networks however is that their diameter and average distance tend to be large as the number of nodes increases, and so there has been a continuous effort in finding degree-4 or low-degree networks that have small diameters and average distances [1, 2, 4, 5, 9, 10]. One of the recent proposals is the *midimew* network (Minimum Distance Mesh with Warp-around links) [2]. This family of networks is isomorphic to a subset of circulant graphs [3] whose diameter and average distance are minimum among all degree-4 circulant graphs. For the same number of nodes, a midimew network outperforms the 2-D mesh, the 2-D torus, and the 3-D mesh, and compares favorably with the 3-D torus and the hypercube in terms of the typical cost measure of diameter $\times$ degree [2]. Other characteristics of midimew networks include regularity, vertex-symmetry, fault-tolerance, and the existence of simple routing functions for them, which obviously have positive bearings on the practical implementations of midimews.

The midimew network is of degree 4 and is most naturally laid out on a 2-D plane. The normal way of laying them out on a plane however would lead to long "cross wires" whose length grows with the size of the network (see the example in Figure 7(a) as well as those in [2]). It is the aim of this paper to show how they can be laid out in a different manner so that the longest wires are confined to a length which is a constant and is independent of the size of the network. The importance of keeping wires short is well-recognized in the design of parallel machines and VLSI systems [6, 8, 11, 15, 17, 19]. Formally, this is a problem of embedding a guest graph of a midimew network onto a host graph of a 2-D grid with minimization of the dilation. The constant dilations that we achieve through our embeddings can be proved to be optimal under the assumption of "synchronous mapping" in which an entire row or column is moved at a time during the embedding process. There are reasons to believe that these dilations are optimal or very near optimal even when this assumption is waived.

Section 2 reviews the basic properties of midimews. Section 3 presents the fundamental operations that are subsequently used to do the embedding. The actual embeddings for the various cases and their optimality are presented in Section 4.

## 2 Preliminaries

Given a number of nodes $N$, $N > 2$, the following dimension parameters determine the overall structure of a midimew network which is as shown in Figure 1 in which a filled circle represents a node, and an empty circle an unused slot on the grid.

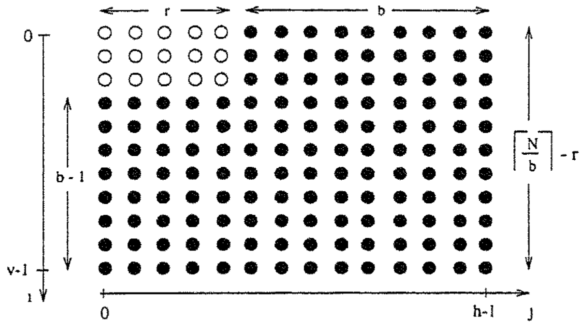



Figure 1: The dimensions of midimew

$$b = \left\lceil \sqrt{\frac{N}{2}} \right\rceil$$

$$r = \left\lceil \frac{N}{b} \right\rceil b - N, \qquad 0 \le r < b$$

$$h = b + r = \left( b \left\lceil \frac{N}{b} \right\rceil + b \right) - N$$

$$v = \left\lceil \frac{N}{b} \right\rceil - r = N - (b-1) \left\lceil \frac{N}{b} \right\rceil$$

Figure 2 shows how the structure of Figure 1 comes about. Between (a) and (b), the $r$ empty slots are moved to the side, and rotated upward. Then the entire shaded block, together with the bottom row of filled circles, is tilted upward and placed alongside the main block.

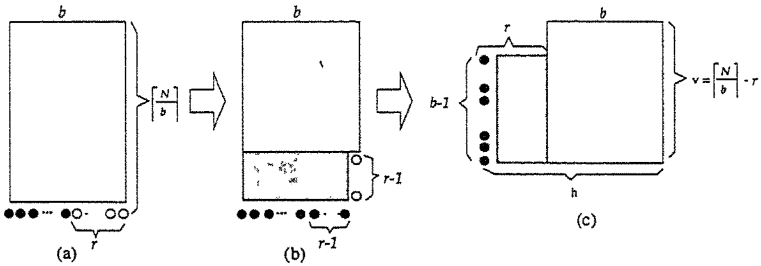



Figure 2: Laying out the nodes

After this, the connections are made, which is according to the following two rules:

**Rule 1.** Vertically each bottom node $(v-1, j)$ is connected to the top node of the column $(j + r) \bmod h$; that is, there is a deviation of $r$ for vertical wrap-around links.

**Rule 2.** Horizontally each leftmost node $(i, h-1)$ is connected to the rightmost node of the row $(i + b - 1) \bmod v$; that is, there is a deviation of $b - 1$ for horizontal wrap-around links.

When $N$ is a multiple of $b$, *i.e.*, $r = 0$, the midimew network is rectangular, such as the one in Figure 7(a) for which $N = 66, b = 6, r = 0, h = 6$, and $v = 2b - 1 = 11$. In this paper, we limit our discussion to rectangular midimews because the rectangular shape is likely to be most natural for practical implementations. Also, the routing functions for a rectangular midimew should be easier to derive than for a non-rectangular midimew. Following are some additional properties concerning midimew networks, in particular the rectangular ones.

**Lemma 1** *(See also [2].)*

1. *For each* $b$, $2(b-1)^2 < N \leq 2b^2$.
2. *For each* $b > 2$ $(N > 8)$, *there are* $4b - 2$ *midimew networks.*
3. *Out of* $4b - 2$ *midimew networks for each* $b$, *only five are rectangular.*

**Proof:**

1.

$$b = \left\lceil \sqrt{\frac{N}{2}} \right\rceil$$

*i.e.*,

$$b - 1 < \sqrt{\frac{N}{2}} \leq b.$$

Solve for $N$ and get the result.

2. The number of midimew networks is $2b^2 - 2(b-1)^2$.

3. Refering to Figure 1, there are two cases for which the resulting midimew networks are rectangular:

   - When $b + r = b$, *i.e.*, $r = 0$. This case occurs when $N$ is a multiple of b, and then the dimensions of the network are $h = b$ and $v = N/b$. According to Part 1 of the lemma, for each $b$, exactly four such rectangles appear, with $v = 2b - 3$, $v = 2b - 2$, $v = 2b - 1$, and $v = 2b$, corresponding to $N = 2b^2 - 3b$, $N = 2b^2 - 2b$, $N = 2b^2 - b$, $N = 2b^2$.

• When

$$b - 1 = \left\lceil \frac{N}{b} \right\rceil - r$$

*i.e.*,

$$r = \left\lceil \frac{N}{b} \right\rceil - b + 1.$$

This case occurs when

$$N = \left(\left\lceil \frac{N}{b} \right\rceil + 1\right)(b - 1)$$

and then the dimensions of the networks are $h = N/(b-1)$ and $v = b - 1$. Within the domain of $N$ for each $b$, only one such rectangle appears, namely with $h = 2b - 1$, that is, $N = 2b^2 - 3b + 1$ and $r = b - 1$.

■

Now we isolate these five rectangular midimew networks for a given $b$ and display them as in Figure 3. For (a) to (d), the vertical wrap-around links (by **Rule** 1 above) that join a top node

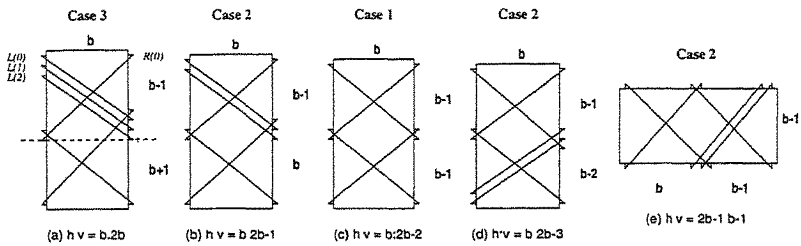



Figure 3: The five rectangular midimews (for a given $b$)

and a bottom node have a deviation of $r = 0$. They are omitted in the figure in order to highlight the other kind of links. Figure 3(e) is equal exactly to Figure 3(b) through a rotation of 90 degrees, and Figure 3(d) looks the same as Figure 3(b) when flipped vertically. For easy identification, we show only the top and bottom horizontal (vertical for (e)) wrap-around links which we refer to as "cross wires" in the sequel. From the pictures ((a) to (d)), we can see there are two sets of cross wires, the ones going from the lower left nodes to the upper right nodes, and the ones going from the upper left nodes to the lower right nodes. For the concern of embedding, they can be classified into three cases based on the difference in number between the two sets of wires: (1) Figure 3(c) for which the two sets of wires are equal, (2) Figure 3(b,d,e) with a difference of 1, and (3) Figure 3(a) with a difference of 2.

An embedding in our context here takes a midimew and places it on the grid so that the connectivity is preserved, but the wires are shortened to the utmost. Given an original layout, the dilation of the final layout after the embedding is the number of times a wire is stretched.[1] For our case here, we consider all wires to be of length 1 before the embedding (which is possible if given a 3-D space), and therefore the dilation in question is the length of the longest wire after the embedding onto a 2-D grid. For easy visualization, we draw all wires as (shortest) straight lines, and so wires that join nodes at different rows and columns are slanted (actually the hypotenuse of a triangle). For a strict 2-D embedding, these slanted wires have to be "wired around" the nodes—*i.e.*, they become series of vertical and horizontal wire segments. The resulting new length however is at most $\sqrt{2}$ times that of the length of a direct, slanted straight wire, and hence the two length measures should be more or less equivalent as far as optimizing dilation is concerned. In this paper, we connect the nodes directly using straight wires and consider the dilation of a slanted wire equal to the sum of the two right-angled edges of the triangle.

## 3 Tools for the Embedding

**Synchronous mapping:** To arrive at an optimal embedding, we take the midimew through a number of transformations, each of them corresponds to moving one or more rows or columns from one position to another. Note that an entire row or column is moved, not individual nodes—this we refer to as *synchronous mapping*. One reason for using synchronous mapping is that it makes the analysis of the series of transformations tractable. We note at the end that the dilations resulted from our embedding procedures are but a small constant (4 or 5) which makes the need of trying out asynchronous mapping appear to be not so important.

**Perfect shuffle:** Considering the cases in Figure 3, one transformation that could be helpful is the perfect shuffle [12]. It first divides the rows into two equal halves, the upper half and the lower half, and then the two halves are shuffled together so that the rows of the two halves interleave evenly. This works perfectly if the two halves are equal, such as Figure 3(c), but not so for the other cases; some variations of the perfect shuffle are used instead. The result of the shuffling operation is that the nodes (in different halves) that are previously joined by a cross wire become much closer to each other. As such, the previously slanted cross wires are reduced in length.

[1] Readers unfamiliar with graph embedding and its related concepts may refer to the text by Leighton [13].

**Ring embedding:** Apart from the cross wires which in the original layout grow with the size of the network, there is this second kind of long wires—the vertical wrap-around wires that connect the bottom nodes to the top nodes (see Figure 7(a))—that need to be shortened as well. Notice that the column of nodes in which such a vertical wire appears is in fact a ring, and hence the obvious way to shorten these wires is through ring embedding onto a linear array. Figure 4 shows two optimal (dilation-2) embeddings of a ring of $n$ nodes onto a linear array. It is not hard to see that these are the only possible dilation-2 embeddings assuming without loss of generality that node 0 is fixed at one end and that the nodes in the upper chain are numbered first. We use these

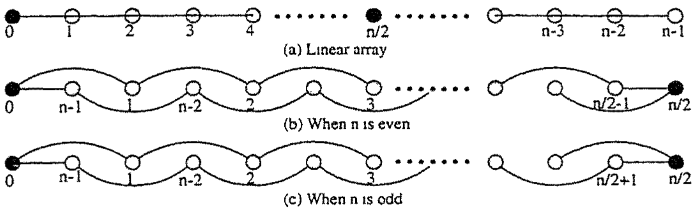



Figure 4: Dilation-2 ring embedding

embeddings as the transformation to shorten long wires whenever they appear in a ring during the layout process.

Using basically these two transformations—perfect shuffle and ring embedding—and adhering to synchronous mapping we are able to arrive at the optimal layouts of the rectangular midimew networks.

## 4 Constant-Dilation Layout of Midimews

We consider all five of the rectangular midimews for any given value of $b$, which can be separated into three cases as distinguished by the difference in number between the two sets of cross wires (Figure 3). We first present a lower bound for the dilation, which is relevant to the first two cases.

**Lemma 2** *For embedding a ring with $n$ nodes, indexed $0, 1, \cdots, n-1$, onto a linear array such that node 0 is placed at one end of the linear array, it is impossible for node $p$ ($p$ subject to a condition as indicated in the proof) to be at a distance of 3 or less from node 0 if the embedding is to achieve a dilation of 3 or less.*

**Proof:** Suppose node $p$ is placed at position 3 (see Figure 5); then node 1 and node $n-1$ must be placed at positions 1 and 2 in order to achieve a maximum dilation of 3. Nodes 2 and $n-2$ must then be placed at positions 4 and 5. Now there is one position, position 6, left to place $p$'s two neighbors, $p+1$ and $p-1$. Provided $n-3 > p > 3$, the theorem is proved for the case of $p$ being at position 3. For $p$ at position 2 or 1, the proof is similar. ■

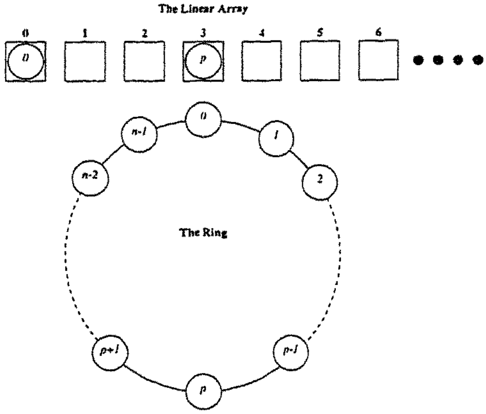



Figure 5: Dilation-3 ring embedding

**Theorem 1** *The lower bound of the dilation for laying out rectangular midimews in a 2-D grid is 4.*

**Proof:** According to **Rule 2** of the midimew construction procedure (see also [2]), there is a deviation of $b-1$ for horizontal cross wires that go from the upper leftmost nodes to the lower rightmost nodes. Now suppose we are able to shuffle the columns so that the leftmost and the rightmost column are now next to each other as in Figure 6. If we were to shorten the cross wire to a length of less than 4 (or a dilation of less than 4), the node of the right column at row $b-1$ must be moved closer to the top so that it is at a distance of 3 or less (or 2 or less if we consider dilation) from the top node. This is impossible without causing a dilation of greater than 3 for the vertical rings according to Lemma 2 if $n-2 > b > 4$, where $n$ is the number of nodes in a column—*i.e.*, $b > 4$ for all five rectangular midimews for a given $b$. ■

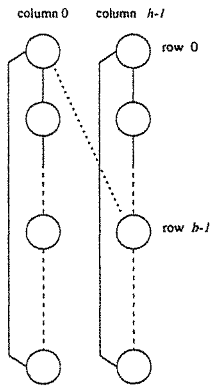


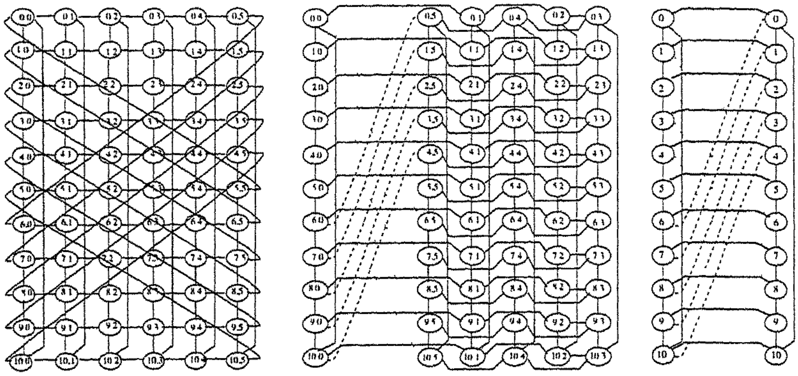

Figure 6: Lower bound for the dilation

Since cross wires are the hypotenuse of a right-angled triangle, by shortening the right-angled (vertical and horizontal) edges of the triangle, these wires are also shortened. To shorten the horizontal edge, we can do a ring embedding on the horizontal rings, as shown in Figure 7. Given that we perform only synchronous mapping, we can represent Figure 7(b) by Figure 7(c) (and abbreviate the node indices accordingly) since the latter preserves both kinds of long wires. By shortening the two long vertical long wires in Figure 7(c), all the other vertical long wires in Figure 7(b) are simultaneously shortened. For the following discussions, we will use the simplified structure consisting of two vertical columns.

### 4.1 Case 1

This is the case that the two sets of wires (going from lower left nodes to upper right and from upper left to lower right of the structure) are equal in number—corresponding to Figure 3(c). For each $b$, $v = 2b - 2$. The embedding procedure is as follows.

1. **Shuffling:** Because of the symmetric butterfly style of the cross wires, we do perfect shuffling on the columns synchronously, evenly interleaving the upper-half nodes $0, 1, \cdots, v/2 - 1$ with the lower-half nodes $v/2, v/2 + 1, \cdots, v - 1$.
2. **Node pairing:** The shuffling reduces the length of the cross wires to the utmost, but creates one extra long vertical wire for each column, that between the middle two nodes, $v/2 - 1$



(a) original layout (b) rows ring-embedded (c) an abstract view

Figure 7: A midimew example (Case 2, $b = 6$)

and $v/2$. Together with the original long vertical wire between node 0 and node $v - 1$, we have two long vertical wires in each column to shorten. These two wires in fact belong to a ring together if we group pairs of adjacent nodes in a column to form compound nodes—that is, $\{0, v/2\}, \{1, v/2 + 1\}, \cdots, \{v/2 - 1, v - 1\}$. Note that each such pair of nodes are joined to the corresponding nodes in the other column by two minimum-length cross wires. The pairing into compound nodes is meant to preserve the optimal improvement on the cross wires achieved in the previous step.

3. **Ring embedding:** We apply ring embedding to the two rings formed by compound nodes. This shortens the remaining long wires—the two long vertical wires in each column—to the utmost.

An example for $b = 5$ and $v = 2b - 2 = 8$ is shown in Figure 8. Part (g) there shows the complete look of the midimew after the above operations.

**Theorem 2** *The dilation of embedding a Case 1 midimew network (of the form of Figure 3(c)) using the above procedure is 4 which is optimal.*

**Proof:** The horizontal wires are reduced to 2 (or 1) through ring embedding as discussed in the beginning of the section. The cross wires are shortened to $\sqrt{2}$ (dilation 3) by the perfect shuffle and then remain unchanged till the end because of node pairing. The perfect shuffle doubles the

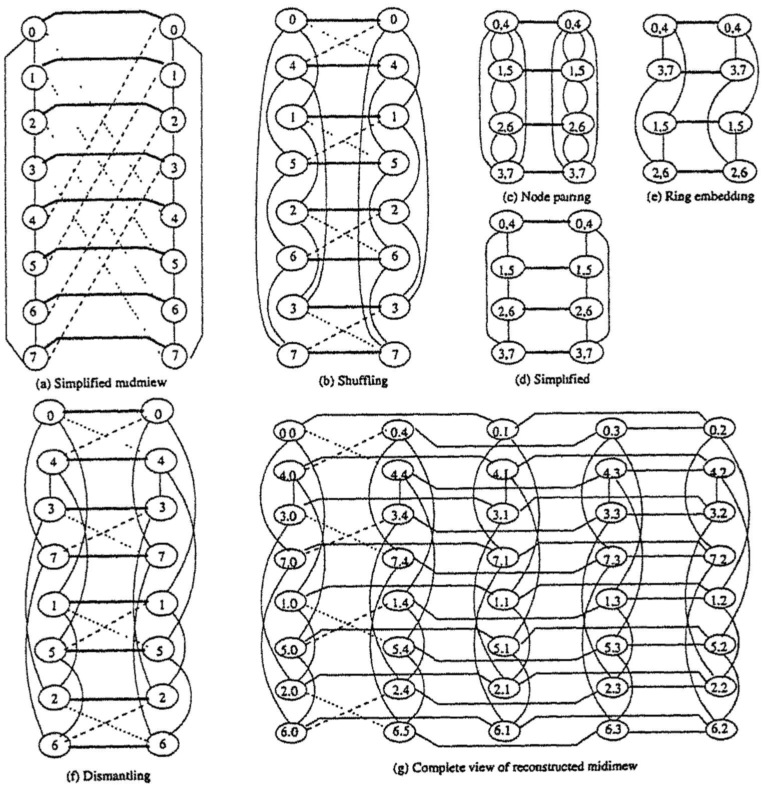



Figure 8: Case 1 ($b = 5$)

unit-length of the short vertical wires in each column (and creates one extra long wire per column). And then in shortening the long vertical wires by (dilation-2) ring embedding, some of these wires whose length has been doubled are further stretched—to a length of 4. The long vertical wires are reduced once, to a length of 2 in a ring composed of compound nodes, and since a compound node represents two adjacent real nodes, so the maximum length of these wires is 4. The longest wire is therefore of length 4. Comparing with the lower bound result (Theorem 1), the embedding is optimal. ■

Finally we note that since the two sets of cross wires are equal in number, the above procedure is applicable to and the theorem valid for both odd and even $b$.

### 4.2 Case 2

This case corresponds to Figure 3(b,d,e) of which the difference in number between the two sets of cross wires is 1. For each $b$, $v = 2b - 1$ or $2b - 3$. We use Figure 3(b) as the representative structure. Figure 3(d) and (e) can be transformed to this structure by flipping and rotating respectively. The embedding procedure is as follows.

1. **Shuffling**: Since the structure cannot be divided into two symmetric halves, we use a variation of the perfect shuffle: the lower $b - 1$ nodes $b, b + 1, \cdots, 2b - 2$ are shuffled evenly "into" the upper $b$ nodes $0, 1, \cdots, b - 1$. This reduces the length of all the cross wires except the one between node 0 and node $b - 1$ which is prolonged to the utmost.

2. **Ring embedding**: If we compound each pair of horizontal nodes in the two columns, we can see that the cross wires now form a ring. We therefore apply a ring embedding to the columns to shorten the single long cross wire as well as the long vertical wires resulted from the previous step (from 0 to $v - 1$ and from $b - 1$ to $b$).

An example for $b = 5$ and $v = 2b - 1 = 9$ is shown in Figure 9. Like the previous case, the procedure is applicable to both even and odd $b$ and to both $v = 2b - 1$ and $v = 2b - 3$ (in any case, the cross wires form a ring).

**Theorem 3** *The dilation of embedding a Case 2 midimew network (of the form of Figure 3(b,d,e)) using the above procedure is 4 which is optimal.*

**Proof:** Similar to the proof of Theorem 2 (one shuffle followed by one ring embedding). ■

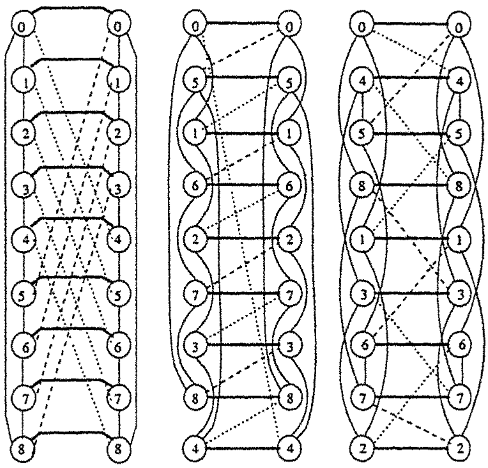



Figure 9: Case 2 ($b = 5$)

### 4.3 Case 3

This is the difficult case which corresponds to Figure 3(a). The difference in number between the two sets of cross wires is 2. If we follow the previous formula of one shuffling followed by one ring embedding, we would end up with a dilation of 7 which seems a bit far from the dilation of 4 in Case 1 and Case 2. Instead we take a different approach. Let's first consider a special property of Case 3 which is important for the proofs below. In Figure 3(a), we see that the cross wire labeled $R(0)$ meets the cross wire labeled $L(2)$ on the same row. In general, $R(i)$ meets $L(i+2)$ on the same row, for $i = 0, \cdots, b-2$. Then, focusing on the leftmost column and the rightmost column and ignoring the horizontal edges (*i.e.*, compounding pairs of nodes horizontally), we have the following lemma.

**Lemma 3** *For a Case 3 midimew, if $b$ is odd, the cross wires divide the nodes into two connected components: (1) a ring of even nodes $0, b+1, 2, b+3, \cdots, 2b-2, b-1$, and (2) a ring of odd nodes $1, b+2, 3, b+4, \cdots, 2b-1, b$. And if $b$ is even, the cross wires connect the nodes into a single ring, and the sequence of the nodes is: $0, b+1, 2, b+3, 4, \cdots$.*

Figure 10 and Figure 11 are two examples of Case 3 with an odd $b$ and an even $b$ respectively, including the rings that are formed by the cross links. The embedding procedure is as follows.

1. **Ring embedding:** We take the ring(s) as described in Lemma 3 and apply a ring embedding to them in order to shorten the cross wires.

2. **Shuffling:** For odd $b$ (there are two rings), we shuffle the two rings together so that the resulting nodes are in the same relative order as they are in the original structure—*i.e.*, node 1 is in between nodes 2 and 3, $\cdots$, node $i$ in between nodes $i-1$ and $i+1$. The reason for this is to bound the dilation of the vertical links. For even $b$ (there is one ring), we do the similar thing so that the resulting nodes are in the original relative order.

An example for $b = 9$ and $v = 2b = 18$ is shown in Figure 10. An example for $b = 8$ and $v = 2b = 16$ is shown in Figure 11.

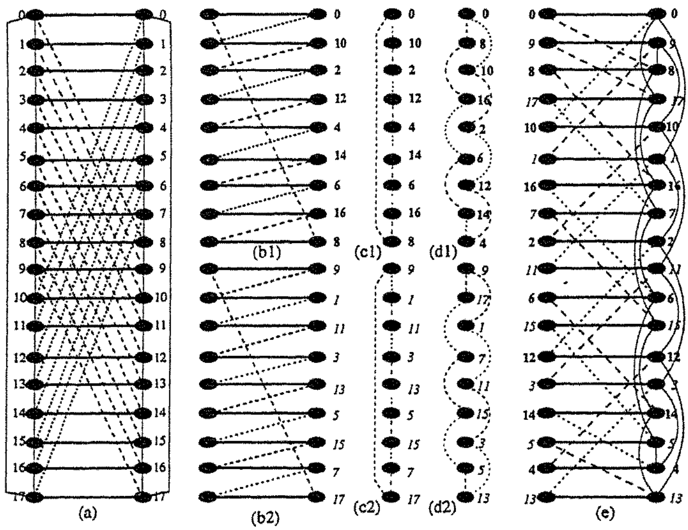



Figure 10: Case 3—$b$ odd ($b = 9$)

Given the special property of Case 3, we first derive a lower bound for the dilation which is specific to this case.

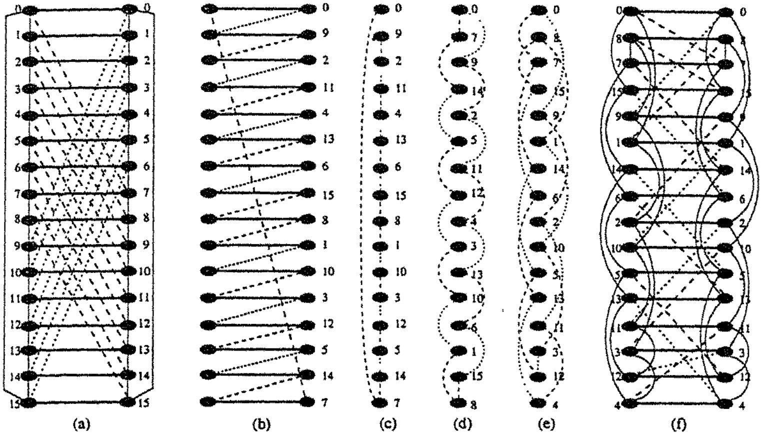



Figure 11: Case 3—$b$ even ($b = 8$)

**Theorem 4** *The lower bound for the dilation of embedding a Case 3 midimew in a 2-D grid is 5.*

**Proof:**

**When $b$ is odd:** (See also the example in Figure 10.) In order to shorten the long cross wires in the two rings identified in Lemma 3, the two rings must be embedded (in a linear array) with a reasonably small dilation. From the lemma, we note that the original order and the inter-node distances for the set of even nodes and the set of odd nodes are preserved in these rings before they are ring-embedded (Figure 10(c1,c2)). After the embedding, the inter-node distance between some pairs of even (or odd) consecutive nodes is dilated from 2 to at least 4 (Figure 10(d1,d2)). When we put these two embedded rings together back to a vertical column, this inter-node distance is expanded to at least 8 for some of the pairs. This is because we are interleaving $b$ even nodes with $b$ odd nodes, and both sets span the full length (minus one) of the vertical column. Now consider two such consecutive nodes, say node 0 and node 2 without loss of generality, as shown in Figure 12. According to Lemma 3, node $b+1$ is joined to node 0 and node 2 by cross wires. If node $b+1$ is placed in the middle, then the dilation is 5 due to the cross wires; any relocation of node $b+1$ would result in prolongment of the cross wires and a dilation of greater than 5.

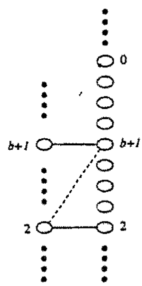



Figure 12: Lower bound for Case 3

**When $b$ is even:** (See Figure 11.) The proof is similar, except we note that the two sets of even and odd nodes are automatically evenly interleaved in the single ring formed by the cross wires. ∎

**Theorem 5** *The dilation of embedding a Case 3 midimew network (of the form of Figure 3(a)) using the above procedure is 5 which is optimal.*

**Proof:** We use the example in Figure 10 for the arguments which are valid for the general case of even or odd $b$. The rings of Figure 10(c1,c2) preserve the original physical distances of the nodes—for example, node 0 and node 2 are two edges apart, so are node 2 and node 4; similarly for the other pairs. After the ring embedding, these distances are doubled to become 3 or 4. The shuffling then expands these distances to a maximum of 8, and the shuffling is done in such a way that node 1 is placed mid-way between node 0 and node 2 and similarly for the other nodes, but since this middle position must be occupied by an even-indexed node, node 1 is placed one position away from the middle position; similarly for the other nodes. Hence, the original vertical edge between node 0 and node 1 is stretched to 5 (or 3, then the distance between node 1 and node 2 would be 5). Since the order of the nodes in the original vertical rings (Figure 10(a)) is preserved, the long vertical wire that joins node 0 and node $v-1$ in the original structure (17 in this case) becomes of length either 3 or 5. Finally, the cross wires, after the ring embedding, become of maximum length 2, and then the shuffling doubles their length to a maximum of 4; the expansion back to two columns extends their length to a maximum of $\sqrt{17} \approx 4.12$ or a dilation of 5. Hence, the maximum dilation is 5 which is optimal according the Theorem 4. ∎

# 5 Concluding Remarks

We have shown how all five types (in three cases) of rectangular midimews for a given $b$ can be embedded in a 2-D grid with constant dilation. The dilations for the three cases are 4, 4, and 5 respectively which are optimal under the assumption of synchronous mapping. Together with the other positive characteristics of midimew networks, these small constant dilations make midimew a viable candidate for network interconnection in real life. In the appendix, we give the formulas for computing the final grid positions of the nodes of a midimew according to the above embeddings. This is useful for the system implementor. The major item in the agenda for future work is *asynchronous* mapping which might or might not improve the dilations obtained in this paper.

## Appendix—Mapping Formulas

First we give the formulas for the basic operations.

**1. Ring embedding:** Without loss of generality, we adopt the dilation-2 embedding (Figure 4) in which node $v-1$ is at the second leftmost position. The mapping formula is as follows.

$$\mathrm{R}_1(v,i) = ((i+1) \bmod 2)\frac{i}{2} + (i \bmod 2)(\frac{v-i}{2} + \frac{v}{2}) \tag{1}$$

Where $v$ is the total number of nodes, $i$ $(0 \leq i \leq v)$ is the position index of the linear array. $\mathrm{R}_1(v,i)$ returns the new node number at position $i$ after the embedding. The division operator is that of integer division; the same applies to the following formulas.

**2. Node-paired ring embedding:** This is used in Case 1 where there are $v$ nodes ($v$ even), and we pair every two nodes together before ring embedding.

$$\mathrm{R}_2(v,i) = ((\frac{i}{2}+1) \bmod 2)\frac{i+1}{2} + (\frac{i}{2} \bmod 2)(v - 1 - \frac{i}{2} + i \bmod 2) \tag{2}$$

**3. Shuffling:** For even $v$ nodes $(0, 1, 2, \cdots, v-1)$, we divide them into two equal halves. Each node of the second half $(v/2, v/2+1, \cdots, v-1)$ is inserted immediately under the corresponding node of the first half $(0, 1, 2, \cdots, v/2-1)$. For odd $v$ nodes, the two halves have a difference of 1 in the number of nodes. Each node of the second (smaller) half is inserted into the corresponding gap of the first half. Their formulas are as follows.

$$\mathrm{S}_1(v,i) = ((i+1) \bmod 2)\frac{i}{2} + (i \bmod 2)(\frac{i}{2} + \frac{v}{2}) \tag{3}$$

$$S_2(v,i) = ((i+1) \bmod 2)\frac{i}{2} + (i \bmod 2)(\frac{i}{2} + \frac{v+1}{2}) \quad (4)$$

Next the formulas for the three cases of rectangular midimews.

**Case 1:** For the horizontal direction, we do one ring embedding operation. For the vertical direction, we do a shuffling followed by a node-paired ring embedding.

$$< i,j > \Longrightarrow < S_1 \cdot R_2(i), R_1(j) > \quad (5)$$

**Case 2:** For the horizontal direction, we do one ring embedding operation. For the vertical direction, we do a shuffling followed by a ring embedding.

$$< i,j > \Longrightarrow < S_2 \cdot R_1(i), R_1(j) > \quad (6)$$

**Case 3—odd** $b$: First, all even-numbered nodes and all odd-numbered nodes take part in shuffling into two rings respectively. The mapping formula is (7). Then, the two rings are ring-embedded, and the corresponding formula is (8). The last step is the shuffling operation, which puts the two rings together back as a single column. The combined formula is (9).

$$\mathbf{DS}(v,i) = ((v-i-1) \bmod \frac{v}{2}) \cdot 2S_2(\frac{v}{2}, i) + (i \bmod \frac{v}{2})((\frac{v}{2} + 2S_2(\frac{v}{2}, i - \frac{v}{2})) \bmod \frac{v}{2}) \quad (7)$$

$$\mathbf{DR}(v,i) = ((v-i-1) \bmod \frac{v}{2}) \cdot R_1(\frac{v}{2}, i) + (i \bmod \frac{v}{2}) \cdot (R_1(\frac{v}{2}, i - \frac{v}{2}) + \frac{v}{2}) \quad (8)$$

$$< i,j > \Longrightarrow < \mathbf{DS} \cdot \mathbf{DR} \cdot S_1(i), R_1(j) > \quad (9)$$

**Case 3—even** $b$: First, all even-numbered nodes are fixed at where they are, and all odd-numbered nodes are cyclically shifted $v/2$ positions; the corresponding mapping formula is (10). The next two steps are ring embedding operations. The combined formula is (11).

$$\mathbf{C}(i) = ((i+1) \bmod 2) \cdot i + (i \bmod 2)((i + \frac{v}{2}) \bmod v) \quad (10)$$

$$< i,j > \Longrightarrow < \mathbf{C} \cdot R_1 \cdot R_1(i), R_1(j) > \quad (11)$$

# References


[1] Arden, B.W. and K.W. Tang, Representations and Routing for Cayley Graphs, *IEEE Trans. on Communications*, Vol. 39, No. 11, November 1991, 1533-1537.

[2] Beivide, R., E. Herrada, J.L. Balcázar, and A. Arruabarrena, Optimal Distance Networks of Low Degree for Parallel Computers, *IEEE Trans. on Computers*, Vol. 40, No. 10, October 1991, 1109-1124.

[3] Boesch, F.T. and R. Tindell, Circulants and Their Connectivities, *J. of Graph Theory*, Vol. 8, 1984, 487-499.

[4] Braun, H. and F.C. Stephan, On Optimizing Diameter and Average Distance of Directed Interconnected Networks, *IEEE Trans. on Computers*, Vol. 42, No. 3, March 1993, 353-358.

[5] Campbell, L. *et al.*, Small Diameter Symmetric Networks from Linear Groups, *IEEE Trans. on Computers*, Vol. 41, No. 2, February 1992, 218-220.

[6] Dally, W.J., Wire efficient VLSI Multiprocessor Communication Networks, *Proceedings of Stanford Conference on Advanced Research in VLSI*, 1987, 391-415.

[7] Dally, W.J., Performance Analysis of $k$-ary $n$-cube Interconnection Networks, *IEEE Trans. on Computers*, Vol. 39, No. 6, June 1990, 775-785.

[8] Feldman, Y. and E. Shapiro, Spatial Machines: A More Realistic Approach to Parallel Computation, *CACM*, Vol. 35, No. 10, October 1992, 61-73.

[9] Gessesse, G.A. and S. Chalasani, New Degree Four Networks: Properties and Performance, *Proceedings of 7th International Parallel Processing Symposium*, April 1993, 168-172.

[10] Hatz, V., Interconnection Networks Based on Block Designs, *Proceedings of CONPAR 92–VAPP V*, September 1992, 37-42.

[11] Hillis, W.D., *The Connection Machine*, MIT Press, 1985.

[12] Hwang, K., *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, 1993.

[13] Leighton, F.T., *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann Publishers, 1992.

[14] Ni, L.M. and P.K. McKinley, A Survey of Wormhole Routing Techniques in Direct Networks, *IEEE Computer*, Vol. 26, February 1993, 62-76.

[15] Paterson, M.S., W.L. Ruzzo, and L. Snyder, Bounds on Minimax Edge Length for Complete Binary Trees, *Proceedings of 13th ACM Symposium on Theory of Computing*, 1981, 293-299.

[16] Ramanathan, G. and J. Oren, Survey of Commercial Parallel Machines, *ACM Computer Architecture News*, Vol. 21, No. 3, June 1993, 13-33.

[17] Ranade, A.G. and S.L. Johnsson, The Communication Efficiency of Meshes, Boolean Cubes and Cube Connected Cycles for Wafer Scale Integration, *Proceedings of 1987 International Conference on Parallel Processing*, 1987, 479-482.

[18] Varma, A. and C.S. Raghavendra, *Interconnection Networks for Multiprocessors and Multicomputers: Theory and Practice*, IEEE Computer Society Press, 1994.

[19] Vitányi, P.M.B., Locality, Communication, and Interconnect Length in Multicomputers, *SIAM J. on Computing*, Vol. 17, No. 4, August 1988, 659-672.