ELSEVIER

Information Processing Letters 72 (1999) 137–141

Information Processing Letters

www.elsevier.com/locate/ipl

# An optimal layout of multigrid networks

Tiziana Calamoneri *, Annalisa Massini [1]

*Dipartimento di Scienze dell'Informazione, Università di Roma "La Sapienza", via Salaria 113, 00198 Rome, Italy*

Received 1 March 1999; received in revised form 1 April 1999
Communicated by S.G. Akl

**Abstract**

In this paper, we study the problem of laying out a multigrid network, $\mathcal{M}_N$, on a grid of minimum area. Precisely, we describe a layout having area $(\frac{5}{2}N-3)\times(3N-4)$, which is of the same order of magnitude as the lower bound. © 1999 Published by Elsevier Science B.V. All rights reserved.

*Keywords:* Interconnection networks; Grid layout; Multigrid network

## 1. Introduction

The mathematical problem of laying out graphs on grids abstracts a number of computational situations finding their applications in the study of the VLSI layout problem for integrated circuits [15]. Furthermore, a layout is a restricted form of embedding of a graph in the grid [5,6,12,14], hence contributes to the study of the mapping problem for parallel architectures [2,3], particularly the problem of mapping parallel programs onto mesh-structured parallel architectures [13].

One of the most relevant measures to be optimized in a layout is its area occupation. In order to pack interconnection networks into a small area, it is natural to investigate how much grid area is necessary for laying out the corresponding graph.

In this paper, the following rules for a graph layout on the grid are used:

- nodes are mapped to grid-nodes, at most one node per grid-node;
- edges are routed along edge-disjoint paths; two paths sharing an intermediate grid-node must *cross* at that node (i.e., knock-knees are not allowed); a path may touch no node, except at its end-points; edges following this rule will be called *feasible*;
- the grid is a window coordinate system, having its origin in the upper-left corner and positive coordinates towards the right ($x$) and downwards ($y$).

We define the *area* of the layout the product to be $(a+1)\times(b+1)$ where $a\times b$ is the dimension of the smallest bounding box of the layout with sides parallel to the grid-lines.

The layout area of interconnection networks has been researched intensely since the 1980s, and many networks have been laid out in optimal area: CCC [11], shuffle-exchange [9], butterfly [16,1], trivalent Cayley networks [4], Batcher and other sorting networks [7], and many others.

In this paper we will lay out an $N\times N$ *multigrid network*, $\mathcal{M}_N$. To the best of our knowledge, in the literature there are no results about the layout of multigrids, hence this seems to be the first result in this direction.

* Corresponding author. Email: calamo@dsi.uniroma1.it. Supported by the Italian Research Council (CNR).

[1] Email: massini@dsi.uniroma1.it.

PII: S0020-0190(99)00140-4

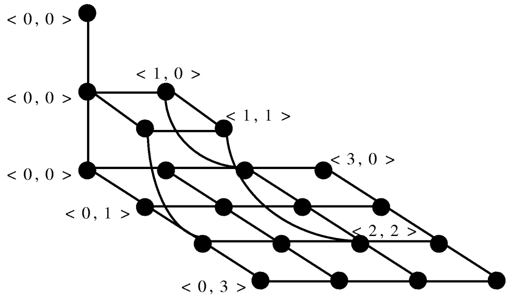

Fig. 1. A 4 × 4 multigrid network.

**Definition 1.** A *multigrid network* $\mathcal{M}_N$, where $N = 2^n$, consists of $\log N + 1$ two-dimensional arrays, each one of size $N/2^k \times N/2^k$ for $0 \leqslant k \leqslant \log N$. The arrays are interconnected so that node $\langle i, j\rangle$ on the $N/2^k \times N/2^k$ array is connected to node $\langle 2i, 2j\rangle$ on the $N/2^{k-1} \times N/2^{k-1}$ array for $0 \leqslant i, j \leqslant 2^k - 1$, and $0 \leqslant k \leqslant \log N$.

From now on, since no confusion arises, we call two-dimensional arrays simply arrays. Furthermore, for sake of simplicity of notation, we denote by $\langle i, j\rangle_k$ node $\langle i, j\rangle$ on the $N/2^k \times N/2^k$ array.

In Fig. 1 an $\mathcal{M}_4$ is represented.

Multigrids are interesting networks since the diameter of $\mathcal{M}_N$ is $\mathrm{O}(\log N)$; as a result, multigrid algorithms often converge much more quickly than do their single-grid counterparts, using relatively little additional processing power [10].

In the following we will show how to lay out an $N \times N$ multigrid $\mathcal{M}_N$ in a grid having $\mathrm{O}(N^2)$ area; namely, we construct a layout of size $(\frac{5}{2}N - 3) \times (3N - 4)$, $\frac{1}{6}N^2 - \frac{8}{3}$ bends and $\frac{3}{2}N - 3$ maximum edge length. The result on the layout area matches the lower bound which is $\Omega(N^2)$. This lower bound can be computed using both the known general formula on the layout area of interconnection networks [15,1] and the number of nodes of the $N \times N$ multigrid network.

## 2. The layout

From the definition of a multigrid, it is straightforward to see that it has maximum degree 6; so, in a layout we represent some of its nodes as horizontal unit-length segments, instead of dots [8]. In order to make this choice consistent with the definition of grid layout, we may think that each node represented by a segment is a pair of nodes joined by a horizontal unit-length straight line.

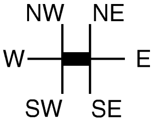

Fig. 2. Names assigned to the directions coming out from segment-nodes.

The nodes represented in the grid as segments and as dots will be called *segment-nodes* and *dot-nodes*, respectively. The position of a segment-node is specified by the coordinates of the leftmost end-point of the segment. In the following, we will denote the directions at a segment-node, as shown in Fig. 2.

For a simpler exposition, we first describe how to lay out all arrays, then we show how to connect adjacent arrays.

The resulting layout is not completely symmetric, therefore we will not give at once the final coordinates. For the sake of clearness, we prefer to describe the layout giving very regular coordinates to nodes and bends, and then to delete empty rows and columns.

### 2.1. *Layout of the arrays*

We lay out each array in a grid-like fashion, and we put node $\langle i, j\rangle_k$ in the middle of the 'square' generated by nodes $\langle 2i, 2j\rangle_{k-1}$, $\langle 2i+1, 2j\rangle_{k-1}$, $\langle 2i, 2j+1\rangle_{k-1}$, $\langle 2i+1, 2j+1\rangle_{k-1}$. In this way, the upper-left corner of the 'square' is to be connected to $\langle i, j\rangle_k$.

Formally, nodes and edges of the arrays are positioned in the following way (see Fig. 3):

**Nodes.** Represent node $\langle i, j\rangle_0$ as dot-node if either $i$ or $j$ is odd, and as segment-node otherwise. Put node $\langle i, j\rangle_0$ at coordinates $(3i, 3j)$.

Let $\langle i, j\rangle_k$ be a node, $1 \leqslant k \leqslant \log N$; it is a dot-node if $k = \log N$, and a segment-node otherwise. Put it at coordinates $(3i \cdot 2^k + 3 \cdot 2^{k-1} - 2, 3j \cdot 2^k + 3 \cdot 2^{k-1} - 2)$.

**Edges.** Draw the horizontal edges of all arrays and the vertical edges of the $N \times N$ array in the obvious way. Represent the vertical edges of the $N/2^k \times N/2^k$ array, $1 \leqslant k \leqslant \log N - 1$, as straight lines, from the SE direction of the general node $\langle i, j\rangle_k$ to the NE direction of its neighbor $\langle i, j+1\rangle_k$.

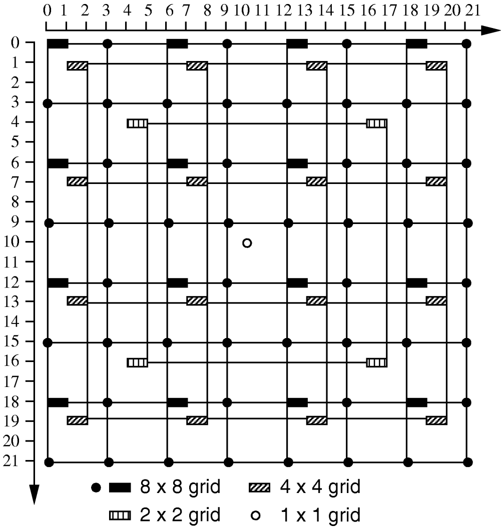

Fig. 3. How to lay out all the arrays in a $\mathcal{M}_8$.

### 2.2. Layout of edges connecting different arrays

Any node $\langle i, j\rangle_k$ must be connected to node $\langle 2i, 2j\rangle_{k-1}$. By construction, both $x$ and $y$ coordinates of $\langle i, j\rangle_k$ are greater than coordinates of $\langle 2i, 2j\rangle_{k-1}$. Furthermore, the NW and the SW directions are free for each node, except for nodes of the $N \times N$ array; for them, the SE direction is free. Therefore, edges connecting different arrays can be laid out as follows (see Fig. 4):

- draw a vertical segment from the NW direction of each segment-node on the $N/2 \times N/2$ array to the SE direction of its neighbor on $N \times N$ array;
- from node $\langle 2i, 2j\rangle_k$, $1 \leqslant k \leqslant \log N - 2$ to its neighbor $\langle i, j\rangle_{k+1}$ draw a three-segment line composed of a unit length vertical segment from the SW direction of $\langle 2i, 2j\rangle_k$, a horizontal segment going to the $x$-coordinate of $\langle i, j\rangle_{k+1}$, and a vertical segment to the NW direction of $\langle i, j\rangle_{k+1}$;
- from segment-node $\langle 0, 0\rangle_{\log N-1}$, draw the connection to the dot-node of the $1 \times 1$ array going from the SW direction to the $y$-coordinate of the dot-node, and to its W direction.

Fig. 4 highlights some rows and columns not used at all by nodes and edges. It is easy to assign new coordinates to nodes and bends in order to compact the layout.

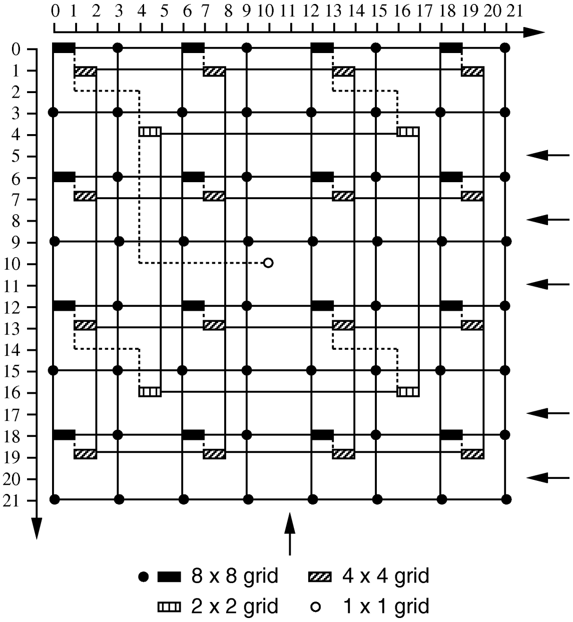

Fig. 4. How to lay out all the edges connecting different arrays in a $\mathcal{M}_8$.

**Theorem 2.** *The described method provides a grid layout of an $N \times N$ multigrid $\mathcal{M}_N$ with area $(\frac{5}{2}N - 3) \times (3N - 4)$, number of bends $\frac{1}{6}N^2 - \frac{8}{3}$, and maximum wire length $\frac{3}{2}N - 3$.*

**Proof.** First of all, let us prove that our method provides a layout. By construction, it is not difficult to see that all nodes (both segment- and dot-nodes) lie in different places. Furthermore, observe that, for each ($x$- or $y$-) coordinate occupied by a node of any array, the same coordinate can be used only by other nodes of the same array.

With respect to paths representing edges, they are feasible as proved by the following reasoning.

- Horizontal edges connecting nodes belonging to any $N/2^k \times N/2^k$ array, $0 \leqslant k \leqslant \log N - 1$, use the same $x$-coordinate as their end-points, so they must be feasible.
- Vertical edges connecting nodes belonging to any $N/2^k \times N/2^k$ array, $1 \leqslant k \leqslant \log N - 1$ use the NE and the SE directions in order not to touch segment-nodes of the $N \times N$ array. Analogously, vertical edges of the $N \times N$ array use the NW and the SW directions in order not to touch segment-

nodes of the $N/2 \times N/2$ array. No other touching and overlapping involving these edges can arise in view of the construction, so all these edges are feasible.

- Edges connecting different arrays are, in general, zig-zag lines whose horizontal portions cannot overlap, in view of the position of nodes of each array (nodes of the next array are positioned inside the 'squares' generated by nodes of the previous array, see Fig. 3). Then, they cannot overlap with lines representing array edges, since horizontal lines of edges connecting different arrays lie, by construction, one row *below* the row containing their upper end-points. Vertical segments of edges connecting different arrays use the NW and the SW directions of their end-points, therefore they are feasible. The edge connecting the $1 \times 1$ array with the $2 \times 2$ array is treated as separate case, and it is easy to see that it is feasible.

The area occupied by the layout obtained according to the described method is $(3N - 2) \times (3N - 2)$. This value can be lowered by eliminating all rows and columns that are not used (highlighted by arrows in Fig. 4).

Utilized rows are:

- rows occupied by nodes. Of those, there are $N$ for the $N \times N$ array, $N/2$ for the $N/2 \times N/2$ array, and so on; thus, there are $2N - 1$ rows in total;
- rows used to connect adjacent arrays. There are $N/2 - 2$ such rows in total, since for $1 \leqslant k \leqslant \log N - 2$, we need exactly $N/2^{k+1}$ rows to connect the $N/2^k \times N/2^k$ array with the $N/2^{k+1} \times N/2^{k+1}$ array.

  Edges from the $N \times N$ array to the $N/2 \times N/2$ array, and from the $2 \times 2$ array to the $1 \times 1$ array do not use a row.

Thus, the total number of used rows is $\frac{5}{2}N - 3$.

About columns, one of them, in the middle of the layout (numbered by 11 in Fig. 4), is not used and therefore can be eliminated. Another column can be gained if we move the dot-node of the $1 \times 1$ array on the bend of its incident edge.

For what concerns the number of bends, it is easy to observe that the number of edges connecting arrays $N/2^k \times N/2^k$ and $N/2^{k+1} \times N/2^{k+1}$, $1 \leqslant k \leqslant \log N - 1$, is $N/2^{k+1} \times N/2^{k+1}$ and each edge contains exactly two bends, except the edge incident the array $1 \times 1$ not bending. It follows that the total number of bends is

$$\sum_{k=1}^{\log N - 2} 2\left(\frac{N}{2^{k+1}} \times \frac{N}{2^{k+1}}\right) = \frac{N^2}{6} - \frac{8}{3}.$$

Finally, it is immediate to see that the longest edges are the edges connecting nodes of the $2 \times 2$ array which are $\frac{3}{2}N - 3$ long. □

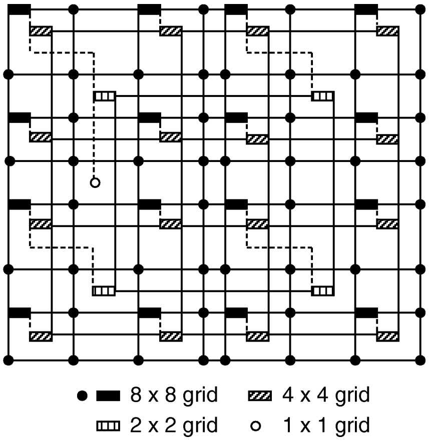

Fig. 5. Final layout of a $\mathcal{M}_8$.

It follows the concluding layout of multigrid network shown in Fig. 5.

## 3. Conclusions and open problems

In this paper, we provide a method to lay out a multigrid network in optimal area, up to a constant factor. Indeed, a lower bound for the layout area is $\Omega(N^2)$, and the area we use is $(\frac{5}{2}N - 3) \times (3N - 4)$.

Two interesting open problems arise from this work. First of all, we can notice that many rows are used only by edges, and many columns cannot be eliminated, since otherwise edges touch some nodes that are not their end-points. This implies a rather big constant factor of the resulting value of the area. On the contrary, the constant factor of the lower bound is $\frac{4}{3}$ if we bound it with the number of nodes, and it is almost 2 if we consider the additional condition that some nodes have degree greater than 4 and therefore they need at least 2 grid-points to be represented. We wonder whether to close the gap in the constant between the upper and the lower bound.

Another question comes from the fact that the maximum wire length of our layout is $\mathrm{O}(N)$. With three-dimensional layout one aims to shorten wires, so an open problem is to lay out an $N \times N$ multigrid in a three-dimensional grid, so that its volume is minimum, i.e., $\mathrm{O}(N^2)$, and its maximum wire length is less than $\mathrm{O}(N)$.

## References

[1] A. Avior, T. Calamoneri, S. Even, A. Litman, A.L. Rosenberg, A tight layout of the butterfly network, Theory of Computing Systems (Math. Systems Theory) 31 (1998) 475–487. Preliminary version in: Proc. 8th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA '96) (ACM Press, New York, 1996) 170–175.

[2] F. Berman, L. Snyder, On mapping parallel algorithms into parallel architectures, J. Parallel Distrib. Comput. 4 (1987) 439–458.

[3] S.N. Bhatt, C.E. Leiserson, Minimizing the longest edge in a VLSI layout, MIT VLSI Memo, Cambridge, MA, 1982, pp. 82–86.

[4] T. Calamoneri, R. Petreschi, Visual representations of trivalent Cayley networks, in: Proc. 11th Internat. Symp. on Computer and Information Sciences (ISCIS XI), 1996, pp. 555–564. To appear on J. on Foundation of Comput. Sci.

[5] G. Di Battista, P. Eades, R. Tamassia, J. Tollis, Algorithms for drawing graphs: An annotated bibliography, Computational Geometry: Theory and Applications 4 (5) (1994) 235–282.

[6] G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, F. Vargiu, An experimental comparison of four graph drawing algorithms, Computational Geometry: Theory and Applications 7 (5–6) (1997) 303–325.

[7] S. Even, Layout area of sorting networks, Manuscript, 1997.

[8] S. Even, G. Granot, Grid layouts of block diagrams—Bounding the number of bends in each connection, in: Proc. Graph Drawing '94 (GD '94), Lectures Notes in Comput. Sci., Vol. 894, Springer, Berlin, 1995, pp. 64–75.

[9] D. Kleitman, F.T. Leighton, M. Lepley, G.L. Miller, New layouts for the shuffle-exchange graph, in: Proc. 13th Annual ACM Symposium on Theory of Computing (STOC '81), 1981, pp. 278–292.

[10] F.T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays, Trees and Hypercubes, Morgan Kaufmann Publ., San Mateo, CA, 1992.

[11] F.P. Preparata, J. Vuillemin, The cube-connected cycles: A versatile network for parallel computation, Comm. ACM 24 (5) (1981) 300–309.

[12] A.L. Rosenberg, Issues in the study of graphs embeddings, in: Proc. 6th International Workshop on Graph-Theoretic Concepts in Computer Science (WG '80), Lectures Notes in Computer Sci., Vol. 100, Springer, Berlin, 1980, pp. 150–176.

[13] L. Snyder, Type architectures, shared memory, and the corollary of modest potential, Ann. Rev. Comput. Sci. 1 (1986) 289–317.

[14] R. Tamassia, Planar orthogonal drawings of graphs, in: Proc. IEEE International Symposium on Circuits and Systems, 1990.

[15] C.D. Thompson, A complexity theory for VLSI, Ph.D. Thesis, Carnegie-Mellon Univ., Pittsburgh, PA, 1980.

[16] D.S. Wise, Compact layouts of banyan/FFT networks, in: H.T. Kung, B. Sproull, G. Steele (Eds.), VLSI Systems and Computations, Computer Science Press, Rockville, MD, 1981, pp. 186–195.