Embedding and Emulation Results for Static Mesh of Optical Buses

Report A/2000/2
ISBN 951-781-617-0

Ville Leppänen
Department of Computer Science, University of Turku, & Turku Centre for Computer Science (TUCS), Finland
E-mail: Ville.Leppanen@cs.utu.fi

We study emulation of common graphs on d-dimensional static mesh of optical buses. The target architecture consists of nodes arranged to a d-dimensional n-sided grid and dn^{d-1} optical buses. A bus is used to connect n nodes along a dimension. All nodes are connected to d buses – one per dimension. Wavelength division multiplexing is used to create channels into each bus. The aim in embedding graphs to such a target architecture is to minimize the number of channels used per bus, to avoid using intermediate targets in implementing the edges of the guest graph, and to minimize the physical distance between communicating nodes within each bus. We study embedding and emulation of meshes, fat meshes, coated meshes, meshes of trees, hypercubes, butterflies and fat trees. We show almost optimal embeddings for each of the graphs.

1. INTRODUCTION

Parallel computers have a routing machinery for processors to communicate with each other. Often the routing machinery has a fixed topology – e.g., a hypercube or a butterfly. Since the physical wiring of (logarithmic) networks is troublesome, optical connections within optical buses provide a tempting alternative for physical wiring [21]. Observe that it might be possible to reconfigure such a wiring easily. As is well-known, the optics provides a huge potential bandwidth both via time division multiplexing (TDM) as well as via wavelength division multiplexing (WDM). Since tunable transmitters (and receivers) are not very fast currently, we only study static connections is this paper.

We denote a d-dimensional n-sided static mesh of optical buses, where each node has ρ (fixed wavelength) receivers and transmitters per bus, by $SMOB(\langle n\rangle^d, \rho)$.

We do not consider using TDM to create channels into optical buses but assume that each channel corresponds to a unique wavelength within a bus. Moreover, we assume that a channel connects only one transmitter to one receiver. We assume no broadcasting facilities. Thus, we need $\rho \times n$ channels per bus.

We consider embedding and emulation of meshes, fat meshes, coated meshes, meshes of trees, hypercubes, butterflies and fat trees. Our aim is to minimize the number of channels used per bus, and to avoid using intermediate targets when implementing edges of the guest graph on the target architecture. A general result (Section 1.2) implies that arbitrary graphs can be emulated rather efficiently on a d-dimensional $SMOB$, but in the worst case up to $2d-2$ intermediate targets may be needed for an edge of the guest graph. We observe that in most cases it is possible to avoid using intermediate targets. Avoiding is important, since re-routing an optical signal in another bus (using a different wavelength) can be a slow operation. We also attempt to minimize the physical distance between any two nodes communicating with each other via a bus. We consider this important, since hiding the delay of transmitting a signal may involve requiring that signals are rather long, restricting the clockrate of some components, or perhaps expecting that certain amount of parallel slackness [27] is available.

1.1. Concepts

Typically, in the literature (see e.g. [2, 11, 12, 25]), embeddings and emulation operate on graphs. An embedding of a graph G into a graph H maps the nodes of G to the nodes of H and assigns a path in H for each edge of G. Through embeddings one hopes to show good functional emulation of the guest graph G on the host graph H. However, our target of embedding and emulation is the $SMOB(\langle n\rangle^d, \rho)$. Since the $SMOB$ is not an ordinary graph, we need to redefine concepts embedding and emulation.

DEFINITION 1.1. An **embedding** $\mathcal{E} = E(G, \mathcal{H})$ of communication graph G into static MOB $\mathcal{H} = SMOB(\langle n\rangle^d, \rho)$

1. assigns the nodes of G to the nodes of $\mathcal{H}$;
2. assigns the mapping of transmitter and receiver channels properly; and
3. maps for each directed edge $v_s \to v_t \in G$ a **communication path** $(u_0, \zeta_1, u_1, \zeta_2, \ldots, u_{l-1}, \zeta_l, u_l)$, where $u_0 = \mathcal{E}(v_s)$, $u_l = \mathcal{E}(v_t)$, and the ζ_i-values are channels used to transmit from u_{i-1} to u_i.

The **length** of communication path $(u_0, \zeta_1, u_1, \zeta_2, \ldots, u_{l-1}, \zeta_l, u_l)$ is l. A communication path is valid, if the nodes u_i and u_{i+1} are connected together with a bus along some axis j, and some transmitter of u_i and some receiver of u_{i+1} along the j'th axis are set to use channel ζ_{i+1}.

The efficiency of an embedding $\mathcal{E} = E(G, \mathcal{H})$ is measured with the following properties:

expansion $\varphi_e(\mathcal{E}) = \frac{\#G}{\#\mathcal{H}}$ is the ratio of the number of nodes in G to the number of nodes in $\mathcal{H}$;

load $\varphi_\ell(\mathcal{E})$ is the maximum number of nodes of G assigned to a single node of $\mathcal{H}$;

dilation $\varphi_d(\mathcal{E})$ is the length of longest path in $\mathcal{H}$ that is an image of an edge of G;

congestion $\varphi_c(\mathcal{E})$ is the maximum number of images of edges of G that share a common communication path "edge" (a channel) in $\mathcal{H}$; and

stretch $\varphi_s(\mathcal{E})$ is the length of longest stretch of a single channel in $\mathcal{H}$ (defined by f_R and f_T). The distance in a bus is measured as the distance in a linear array. In other words, if a communication channel ζ_i connects nodes $\langle x_1, \ldots, x_j, \ldots, x_d\rangle$ and $\langle x_1, \ldots, x_{j-1}, y_j, x_{j+1}, \ldots, x_d\rangle$, the stretch of ζ_i is $|x_j - y_j|$.

Besides the above parameters, we could measure, e.g., the maximum sum of channel stretches per bus (energy and bus volume) or the maximum number of channels crossing a single point in a bus (bus capacity).

DEFINITION 1.2. A stepwise **emulation** $\mathcal{F} = F(\mathcal{E}, S)$ assigns a collision-free schedule S for the movements of packets according to the communication paths. The length of schedule, **emulation time**, is denoted by $\varphi_t(\mathcal{F})$.

A schedule S solves the channel congestion problem, and therefore the property φ_s in fact supersedes the properties φ_c and φ_d. In the following, we are primarily interested in the properties φ_s and φ_t.

1.2. General embedding and emulation results

Let G be some arbitrary communication graph, $\mathcal{H} = SMOB(\langle n\rangle^d, \rho)$, and $\mathcal{E} = E(G, \mathcal{H})$. Lemma 1.1 states the trivial fact that the condition $\varphi_d(\mathcal{E}) = \varphi_c(\mathcal{E}) = 1$ implies an optimal emulation schedule.

LEMMA 1.1. *If $\varphi_d(\mathcal{E}) = \varphi_c(\mathcal{E}) = 1$, there exists such an emulation $\mathcal{F} = F(\mathcal{E}, S)$ that $\varphi_t(\mathcal{F}) = 1$.*

The following Lemmas 1.2 – 1.4 state that some quantities of an emulation can be linearly traded for another quantity. The proofs are trivial. Lemmas 1.2 – 1.4 can be used to derive new emulation results from a given result.

LEMMA 1.2. (trade channels for time) *For each emulation $\mathcal{F} = F(\mathcal{E}, S)$ and c, $1 \le c \le \rho$, there exists such an emulation $\mathcal{F}' = F(\mathcal{E}', S')$, where $\mathcal{E}' = E(G, \mathcal{H}')$, on simulating machine $\mathcal{H}' = SMOB(\langle n\rangle^d, c)$ that $\varphi_d(\mathcal{E}') = \varphi_d(\mathcal{E})$, $\varphi_s(\mathcal{E}') = \varphi_s(\mathcal{E})$, $\varphi_\ell(\mathcal{E}') = \varphi_\ell(\mathcal{E})$, and $\varphi_t(\mathcal{F}') = \lceil \rho/c\rceil \varphi_t(\mathcal{F})$.*

LEMMA 1.3. (trade time for channels) *For each emulation $\mathcal{F} = F(\mathcal{E}, S)$ and $c \in I\!N$, $1 \le c \le \varphi_t(\mathcal{F})$, there exists such an emulation $\mathcal{F}' = F(\mathcal{E}', S')$, where $\mathcal{E}' = E(G, \mathcal{H}')$, on $\mathcal{H}' = SMOB(\langle n\rangle^d, c \times \rho)$ that $\varphi_d(\mathcal{E}') = \varphi_d(\mathcal{E})$, $\varphi_s(\mathcal{E}') = \varphi_s(\mathcal{E})$, $\varphi_\ell(\mathcal{E}') = \varphi_\ell(\mathcal{E})$, and $\varphi_t(\mathcal{F}') = max\{\lceil \varphi_t(\mathcal{F})/c\rceil, \varphi_d(\mathcal{E})\}$.*

LEMMA 1.4. (trade nodes for time) *For each emulation $\mathcal{F} = F(\mathcal{E}, S)$ and $c \in I\!N$, $1 \le c \le n$, there exists such an emulation $\mathcal{F}' = F(\mathcal{E}', S')$, where $\mathcal{E}' = E(G, \mathcal{H}')$,* *on* $\mathcal{H}' = SMOB(\langle\lceil n/c\rceil\rangle^d, \rho)$ *that* $\varphi_d(\mathcal{E}') \leq \varphi_d(\mathcal{E})$, $\varphi_s(\mathcal{E}') \leq \lceil\varphi_s(\mathcal{E})/c\rceil$, $\varphi_\ell(\mathcal{E}') \leq c^d\varphi_\ell(\mathcal{E})$, *and* $\varphi_t(\mathcal{F}') = c^d\varphi_t(\mathcal{F})$.

What is known of embedding an arbitrary graph into the $SMOB$? Rao [24] has shown the following result (Lemma 1.5) by using Hall's Matching theorem [6, 13]. Thus, each 2-degree graph can be emulated in 3 routing steps on a 2-dimensional $SMOB$. Lemma 1.6 states the corresponding result for a d-dimensional $SMOB$. Both results are based on the following ideas: An r-degree graph is a subgraph of a regular r-degree graph; a regular r-degree graph can be seen to consist of r perfect matchings; a perfect matching can be solved by $2d-1$ axial permutation operations (Hall's Matching theorem provides a method to "shrink" the problem size with 1 dimension by doing 2 axial permutations; the 1-dimensional case requires only one permutation); and d permutations can be solved in parallel (one per axis).

Lemma 1.5. *[24] Let* $\mathcal{H}' = SMOB(\langle\lceil\sqrt{N}\rceil\rangle^2, 1)$. *For any* N*-node* r*-degree graph* G*, there exists such an emulation* $\mathcal{F}' = F(\mathcal{E}', S)$ *on* $\mathcal{H}'$*, where* $\mathcal{E}' = E(G, \mathcal{H}')$*, that* $\varphi_\ell(\mathcal{E}') = 1$, $\varphi_d(\mathcal{E}') \leq 3$, *and* $\varphi_t(\mathcal{F}') = 3\lceil\frac{r}{2}\rceil$.

Lemma 1.6. *Let* $\mathcal{H}' = SMOB(\langle\lceil\sqrt[d]{N}\rceil\rangle^d, 1)$. *For any* N*-node* r*-degree graph* G*, there exists such an emulation* $\mathcal{F}' = F(\mathcal{E}', S)$ *on* $\mathcal{H}'$*, where* $\mathcal{E}' = E(G, \mathcal{H}')$*, that* $\varphi_\ell(\mathcal{E}') = 1$, $\varphi_d(\mathcal{E}') \leq 2d-1$, *and* $\varphi_t(\mathcal{F}') = (2d-1) \times \lceil\frac{r}{d}\rceil$.

The result of Lemma 1.6 is surprisingly good considering what is assumed of G. By using Lemma 1.3 the emulations of Lemma 1.5 and 1.6 can be speeded up. Unfortunately, Lemma 1.6 does not attempt to minimize the stretch φ_s: Only the side length $\lceil\sqrt[d]{N}\rceil - 1$ sets a trivial lower bound for it. In the following, we observe that better emulations can be found for several common graphs (than what is provided by Lemma 1.6).

2. EMBEDDING RESULTS

2.1. Mesh

Lemma 2.1 states that it is trivial to emulate a d-dimensional mesh on a d-dimensional $SMOB$. It is also appealing to study emulating a 3-dimensional mesh on a 2-dimensional $SMOB$. The case "a 2-dimensional mesh on a 3-dimensional $SMOB$" is not seen interesting, since the routing properties of a 2-dimensional mesh are clearly weaker than those of a 3-dimensional mesh of the same size. Let $G_{mesh}(\langle n\rangle^d)$ denote the underlying graph of a d-dimensional regular mesh with bidirectional connections.

Lemma 2.1. *There exists such an embedding*

$$\mathcal{E}_m^{d\mapsto d} = E\left(G_{mesh}(\langle n\rangle^d), SMOB(\langle n\rangle^d, 2)\right)$$

that $\varphi_e = \varphi_\ell = \varphi_d = \varphi_c = \varphi_s = 1$.

LEMMA 2.2. *There exists such an embedding*

$$\mathcal{E}_m^{3\mapsto 2} = E\left(G_{mesh}(\langle n\rangle^3), SMOB(\langle n^{3/2}\rangle^2, 3)\right),$$

where $\sqrt{n} \in \mathbb{N}$ *and* $2|\sqrt{n}$*, that* $\varphi_e = \varphi_\ell = \varphi_d = \varphi_c = 1$ *and* $\varphi_s = \sqrt{n}$*.*

Proof. Let $n = s^2$. The idea is to flatten the 3-dimensional mesh along the Z-axis. We reserve an $s \times s$ area for each tower of Z-axis nodes, and flatten all towers in the same way. Clearly, all connections along the X-axis and Y-axis can be implemented with dilation 1 and stretch s. And the implementation of them requires at most 2 receives (and transmitters as well as channels) per bus per node. Clearly, the connections along the Z-axis are implemented within restrictions, if we can draw such a continuous snake (dilation 1) that goes through all the nodes of an $s \times s$ mesh (completeness) and changes X/Y-direction at each node excluding the head and tail of the snake (channel restriction).

The case $s = 2$ is obvious. Assume that in the case $s = 2 \times i$ we have a snake that ends to some of the border points of the $s \times s$ mesh. The snake can be extended to case $s' = 2 \times (i + 1)$ as shown in Figure 1. The construction method clearly works, if $2|s$. ■

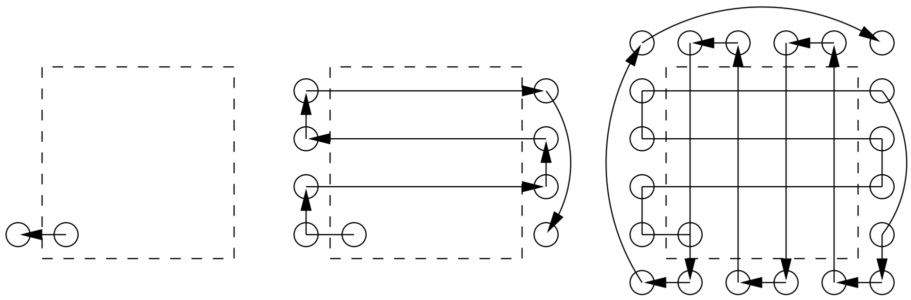

FIG. 1. Continuing a snake.

For $s = 3$, it does not seem possible[1] to prove Lemma 2.2. We conjecture that it is not possible to prove the above result, if $2 \nmid s$ (and $s > 1$). Either the number of channels ρ, the area reserved for Z-axis, or the emulation time must be increased. Lemma 2.3 implies that the result of Lemma 2.2 is asymptotically optimal.

LEMMA 2.3. *For each such embedding*

$$\mathcal{E}_m^{(\star),3\mapsto 2} = E\left(G_{mesh}(\langle n\rangle^3), SMOB(\langle n^{3/2}\rangle^2, \rho)\right),$$

for which $\sqrt{n} \in \mathbb{N}$*, and* $\varphi_\ell = \varphi_d = \varphi_c = 1$*, the stretch* $\varphi_s > \frac{2}{3}\sqrt{n}$*.*

[1]If the number of channels per node per bus is relaxed to 4, we could also have proved Lemma 2.2 simply by using the fact that every $m \times m'$ grid has a Hamiltonian path, if $mm' \equiv 0 \ mod\ 2$ (simply go through the grid row by row).

Proof. The diameter of guest graph is $3n-3$. Thus a packet from the lower left corner node of $\mathcal{H}$ reaches the higher right corner node in at most $3n-3$ hops. Since the Manhattan distance of the two nodes is $2n^{3/2}-2$,

$$\varphi_s \geq \frac{2n^{3/2}-2}{3n-3} > \frac{2n^{3/2}-2}{3n-2} > \frac{2}{3}\sqrt{n}.$$

(For every $a > b > 0$ and $x > 0$, $\frac{a}{b} > \frac{a+x}{b+x}$.) ■

We consider one more embedding, namely that of embedding a $2d$-dimensional mesh into d-dimensional $\mathcal{SMcMOB}$. This provides a very regular construction, where *the logical diameter* $\approx$ *the stretch of embedding.* Whether this kind of balance is important, is left open. Lemma 2.4 states the result. Observe that the construction wastes no transmitters or receivers.

LEMMA 2.4. *There exists such an embedding*

$$\mathcal{E}_m^{2d\mapsto d} = E\left(G_{mesh}(\langle n\rangle^{2d}), SMOB(\langle n^2\rangle^3, 4)\right),$$

where $n \in I\!N$, *that* $\varphi_\ell = \varphi_d = \varphi_c = 1$ *and* $\varphi_s = n$.

Proof. A $2d$-dimensional space can be seen to consist of a d-dimensional regular space of d-dimensional regular spaces. Thus, the embedding is basically composed of n^d meshes of size n^d. The meshes are simply put side by side in the d-dimensions to a regular cube. Corresponding routing machinery nodes in the neighboring spaces are connected to each other. Thus, the stretch equals to the side length n of a d-dimensional n^d-node regular mesh. All the connections are evidently axial, and the node degree $d \times 4$ of $SMOB$ is sufficient to implement all the connections of the $2d$-dimensional mesh nodes. ■

2.2. Fat mesh and coated mesh

A *fat mesh* [8] is a generalization of the ordinary d-dimensional n-sided mesh. The connections between nodes along the i'th axis are simply n-folded. Although the degree of nodes is $d \times n$, each node can forward all the packets passing through the node. The connections between nodes are numbered from $0, \ldots, n-1$, and once a packet is sent using the j'th link, it will traverse to its target using only that link at each node. A *coated mesh* [20] is a d-dimensional n-sided mesh of simple router nodes, whose each surface coated with processor&memory modules. Both coated meshes and fat meshes enable time-processor optimal PRAM simulation [8, 20] whereas the ordinary meshes do not.

Embedding a d-dimensional fat mesh or coated mesh into a d-dimensional $SMOB$ is trivial. Similarly, embedding a $2d$-dimensional fat mesh or coated mesh into a d-dimensional $SMOB$ is straightforward. The resulting embeddings are interesting. (Due to space limitations, we omit the exact details [20].) In the fat mesh case, the stretch $\approx$ the logical diameter $\approx$ the degree of nodes ($\approx 2d$'th root of number of nodes). Respectively for the coated mesh embeddings, the stretch $\approx$ the logical diameter $\approx$ the ratio of the number of router nodes to the number of processors (but the degree of nodes is a constant).

2.3. Mesh of trees

Embedding a d-dimensional mesh of trees into a d-dimensional $SMOB$ can be done by using the fact that the d-dimensional mesh of trees is a leaf-level graph-theoretical product of d binary trees whereas the d-dimensional $SMOB$ is a product of d 1-dimensional $SMOB$s. Lemma 2.6 advances this and gives an (almost) optimal embedding with respect to the result of Lemma 2.5. Whether Lemmas 2.5 and 2.6 can be improved is left open.

Let $G_{mt}(r,d)$ be the underlying graph of an r-sided d-dimensional mesh of trees and assume that

$$\mathcal{E}_{mt}^{(\star),d\mapsto d} = E\left(G_{mt}(r,d), SMOB(\langle\lceil\sqrt[d]{(d+1)r^d - dr^{d-1}}\rceil\rangle^d, \rho)\right)$$

is an ideal embedding for which $\varphi_\ell = \varphi_d = 1$.

LEMMA 2.5. $\varphi_s(\mathcal{E}_{mt}^{(\star),d\mapsto d}) \geq \frac{r-1}{2\log r}$.

Proof. Since $G_{mt}(r,d)$ does not fit into $SMOB(\langle\lfloor\sqrt[d]{(d+1)r^d - dr^{d-1}}\rfloor\rangle^d, \rho)$, in any embedding $\mathcal{E}_{mt}^{(\star),d\mapsto d}$ there must be at least two nodes, whose Manhattan distance is at least $\delta = d \times \lfloor\sqrt[d]{(d+1)r^d - dr^{d-1}}\rfloor - d$. Since the logical diameter of $G_{mt}(r,d)$ is $2d\log r$,

$$\lceil\frac{\delta}{2d\log r}\rceil \geq \frac{d(r-1)}{2d\log r} = \frac{r-1}{2\log r}$$

is a lower bound for the stretch. ∎

LEMMA 2.6. *There exists such an embedding*

$$\mathcal{E}_{mt}^{d\mapsto d} = E\left(G_{mt}(r,d), SMOB(\langle 2r-1\rangle^d, 3)\right)$$

that $\varphi_\ell = \varphi_d = \varphi_c = 1$ *and* $\varphi_s = \lceil(r-1)/\log r\rceil$.

Proof. Heckmann, Klasing, Monien, and Unger [9] provide an optimal embedding of a complete binary tree into one-dimensional array with stretch $\lceil(r-1)/\log r\rceil$. Since the $G_{mt}(r,d)$ is a leaf-level product of d complete binary trees, $\varphi_\ell = \varphi_d = 1$ and $\varphi_s = \lceil(r-1)/\log r\rceil$. Each binary tree leaf is neighbor with one node per axis (the parent of it) and each intermediate node of a binary tree is neighbor with at most 3 nodes along one of the axes (parent + two children). Thus, $\rho = 3$ parallel connections guarantee that $\varphi_c = 1$. ∎

In [9] an almost optimal embedding of a complete binary tree into square grid is also given. It can be used to derive an efficient embedding of $G_{mt}(r,2)$ into a 3-dimensional static MOB (although it might be difficult to efficiently advance an even-sided $SMOB$).

Consider embedding a $2d$-dimensional mesh of trees into a d-dimensional $SMOB$. Since a $G_{mt}(r,2d)$ can be seen as a certain kind of product of d $G_{mt}(r,d)$'s, it is possible to advance the technique of Lemma 2.4 to pack a $2d$-dimensional mesh of trees into a d-dimensional $SMOB$. As before, the contribution of Lemma 2.7 is to shorten the logical diameter at the expense of point-to-point connection length.

LEMMA 2.7. *There exists such an embedding*

$$\mathcal{E}_{mt}^{2d\mapsto d} = E\left(G_{mt}(r, 2d), SMOB(\langle(2r-1)^2\rangle^d, 3)\right)$$

that $\varphi_\ell = \varphi_d = \varphi_c = 1$ *and* $\varphi_s = \lceil (r-1)/\log r \rceil \cdot (2r-1)$.

2.4. Hypercube

Rao [24] has studied embedding a 2^r-node binary hypercube $G_{cube}(r)$ in the 2-dimensional case. The idea is to advance a well-known and obvious VLSI layout (see e.g., [16]). Observe that if $d|r$ for some integer d, the 2^r-node cube is a graph-theoretical product of d cubes of $2^{r/d}$-nodes. Lemma 2.8 simply advances this observation.

LEMMA 2.8. *For* $r' \in I\!N$, *there exists such an embedding*

$$\mathcal{E}_{cube}^{dr'\mapsto d} = E\left(G_{cube}(dr'), SMOB(\langle 2^{r'}\rangle^d, r')\right)$$

that $\varphi_e = \varphi_\ell = \varphi_d = \varphi_c = 1$ *and* $\varphi_s = 2^{r'-1}$.

Finding as efficient embeddings for arbitrary r is not known to us. Rather efficient embeddings can be constructed for a d-dimensional $SMOB$, where $r \bmod d$ of the sides are twice as big as the others. Lemma 2.9 (trivial) states that the result of Lemma 2.8 is close to optimal with respect to the stretch φ_s. Let

$$\mathcal{E}_{cube}^{(\star)} = E\left(G_{cube}(r), SMOB(\langle 2^r\rangle^1, r)\right)$$

be any embedding for which $\varphi_\ell = \varphi_d = \varphi_e = 1$.

LEMMA 2.9. $\varphi_s(\mathcal{E}_{cube}^{(\star)}) \geq (2^r - 1)/r$.

2.5. Butterfly

The case of butterfly is especially interesting, since the SB-PRAM [1, 4, 7] and the Fluent abstract machine [22, 23] are based on the butterfly structure. Layouts for the butterfly have also been considered in several papers. The approach to the layout problem has been either purely VLSI based (see Figure 2) or a more practically oriented [3, 10, 28] (where elements are printed circuits boards, racks, cabinets, ...). It would be desirable to compare the SB-PRAM and the Fluent abstract machine implementations to other PRAM implementations based on coated DAGs [20]. Thus it would be interesting to state the embedding facts related to those butterfly implementations. Unfortunately, the physical layouts are strongly based on the VLSI cost model, and therefore it is difficult to translate the layouts to the $SMOB$ based layouts fairly. In the following, we present a lower bound result for an ideal embedding $\mathcal{E}_{bf}^{(\star),r\mapsto d}$ and upper bound results for two other embeddings.

Let $G_{bf}(r)$ be the underlying graph of an $(r+1) \times 2^r$ node butterfly. Imagine that there is a parametric ideal embedding

$$\mathcal{E}_{bf}^{(\star),r\mapsto d} = E\left(G_{bf}(r), SMOB(\langle\lceil \sqrt[d]{(r+1)2^r}\rceil\rangle^d, \rho)\right)$$

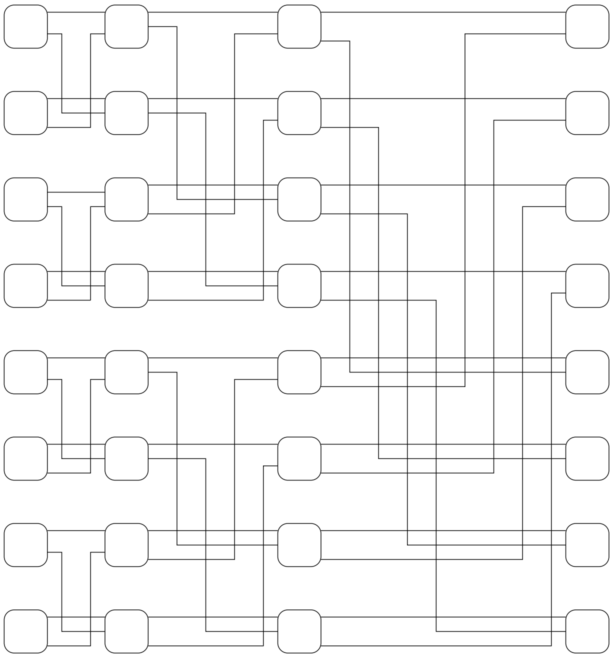

FIG. 2. A VLSI layout of the butterfly.

such that $\varphi_\ell = \varphi_d = \varphi_c = 1$.

LEMMA 2.10. $\varphi_s(\mathcal{E}_{bf}^{(\star),r\mapsto d}) \geq (d-1)2^{(r/d-\log r-1)}$.

Proof. (Proved as Lemma 2.5.) In any embedding $\mathcal{E}_{bf}^{(\star),r\mapsto d}$, there must be at least two nodes, whose Manhattan distance is at least $\delta = d \times \lfloor \sqrt[d]{(r+1)2^r} \rfloor - d$. Since the logical diameter of $G_{bf}(r)$ is $2r$,

$$\lceil \frac{\delta}{2r} \rceil \geq \frac{(d-1)2^{\frac{r}{d}}}{2r} = (d-1)2^{(r/d-\log r-1)}$$

is a lower bound for the stretch. ■

In [24] Rao shows how an $r2^r$-node cube-connected-cycles graph (see e.g., [13]) can be embedded efficiently into a 2-dimensional $SMOB$ by building a Hamiltonian cycle to a $\sqrt{r} \times \sqrt{r}$ subgraph[2] and using the hypercube style embedding on top of the small meshes. The cube-connected-cycles is very similar with the butterfly, and Rao conjectures that the butterfly can be implemented similarly. Indeed, it is wise to wrap each chain of $r+1$ nodes into a $\lceil \sqrt{r+1} \rceil \times \lceil \sqrt{r+1} \rceil$ plane (or a d-dimensional $\lceil \sqrt[d]{r+1} \rceil$-sided cube) and then attempt to apply the hypercube style embedding on top of it. How can we guarantee that (a) all the connections are axial and (b) the number of transmitters (and thus receivers and channels) reserved per node per bus is minimized? Let us call *straight edges*, those edges of the butterfly that preserve the row number, i.e., go straight along a row in Figure 2. Others are

[2] This is possible for $2|\sqrt{r}$. If the Hamiltonian cycle also has the property that the path turns at each node, then the utilization of transmitters and receivers is efficient.

called *cross edges*. The chain construction used in the proof of Lemma 2.2 clearly guarantees the condition (a) for the straight edges. To achieve the condition (a) for the cross edges, we need to agree how all the 2^r chains are arranged. We choose to use the same arrangement for each chain. Consequently, if a connection in the chain from node i to node $i+1$ takes place via the j'th dimension of the $SMOB$, then we require that the j'th dimension is also used to implement all the cross edges from level i to level $i+1$. In other words, we use the j'th axis to change the i'th bit in the binary representation (of the row number).

Consider, the hypercube embedding given in Section 2.4. In it we define that $1/d$'th of the bits are changed using the first axis, one $1/d$'th using the second, and so on. It is irrelevant, which of the bits are changed by using the j'th axis. However, if the number of bits loaded on the j'th axis is x_j, then the side length of $SMOB$ along the j'th axis will be $2^{x_j} \times r'$, where r' is the side length of a small d-dimensional cube into which the chain (of length $r+1$) is wrapped. By minimizing the maximum side length of $SMOB$, we also minimize the stretch (which is $2^{x_j - 1} \times r'$). Minimizing the side length is also related to the condition (b). Namely, if the chain turns at every node, one transmitter per node per bus is enough to implement the chain. To achieve small stretch, we require that the chain uses as evenly as possible the d axes of the $SMOB$. In the case $d = 2$, our construction of Lemma 2.2 clearly has this property. Based on the above discussion, we state Lemma 2.11. The case $d = 3$ is more complicated. It seems to be possible to show a similar result, but we must leave the exact construction of the chain open. We conjecture that in the 3-dimensional case the stretch of $G_{bf}(3r')$ can be upper bounded by $2^{r'-1} \cdot \lceil \sqrt[3]{3r'+1} \rceil$.

LEMMA 2.11. *For $r' \in \mathbb{N}$, there exists such an embedding*

$$\mathcal{E}_{bf}^{2r' \mapsto 2} = E\left(G_{bf}(2n'), SMOB(\langle 2^{r'} \cdot \lceil \sqrt{2r'+1} \rceil \rangle^2, 2)\right)$$

that $\varphi_e = \varphi_\ell = \varphi_d = \varphi_c = 1$ *and* $\varphi_s = 2^{r'-1} \cdot \lceil \sqrt{2r'+1} \rceil$.

2.6. Fat tree

As the butterfly, the fat tree is interesting due to the practically oriented interest it has received. One communication subsystem of the CM-5 machine [19] is a fat tree. The fat trees are also interesting due to certain universality results [15]. On the other hand, the fat trees have a rather high degree and the short logical diameter inevitably implies long physical connections. Moreover, the fat trees (as well as the multibutterfly variant of the butterfly) are closely related to so called expander graphs, which provably have very good routing and fault-tolerance properties [14, 17, 26].

Several definitions (see e.g., [15, 19, 18]) are given for *fat tree* in the literature. Originally, it was defined by Leiserson [18], but in Definition 2.1 we follow the definition given in [15], which defines the fat tree as a layered cross product [5] of a quad-tree and a binary tree.

DEFINITION 2.1. A **fat tree** of height ℓ consists of $\ell+1$ levels. At level k (where $0 \leq k \leq \ell$), it has nodes $V_{k,0}, \ldots, V_{k,2^{2\ell-k}-1}$. Thus, a fat tree has

$$\sum_{i=1}^{\ell} 2^{2\ell-i} = 2^{2\ell} - 2^{\ell}$$

intermediate nodes and $2^{2\ell}$ leaf nodes. Intermediate nodes, except the 2^{ℓ} root nodes, have degree 6. The root nodes have degree 4. A node $V_{k,n}$ (where $0 \leq k < \ell$ and $0 \leq n \leq 2^{2\ell-k} - 1$) is connected with bidirectional links to nodes $V_{k+1,(n \ mod \ 2^k)+(n \ div \ 2^{k+2}) \times 2^{k+1}}$ and $V_{k+1,(n \ mod \ 2^k)+2^{k+1}+(n \ div \ 2^{k+2}) \times 2^{k+1}}$. Thus, a fat tree has altogether $\sum_{i=0}^{\ell-1} 2 \times 2^{2\ell-i} = 2^{2\ell+2} - 2^{\ell+2}$ bidirectional links. The diameter is 2ℓ.

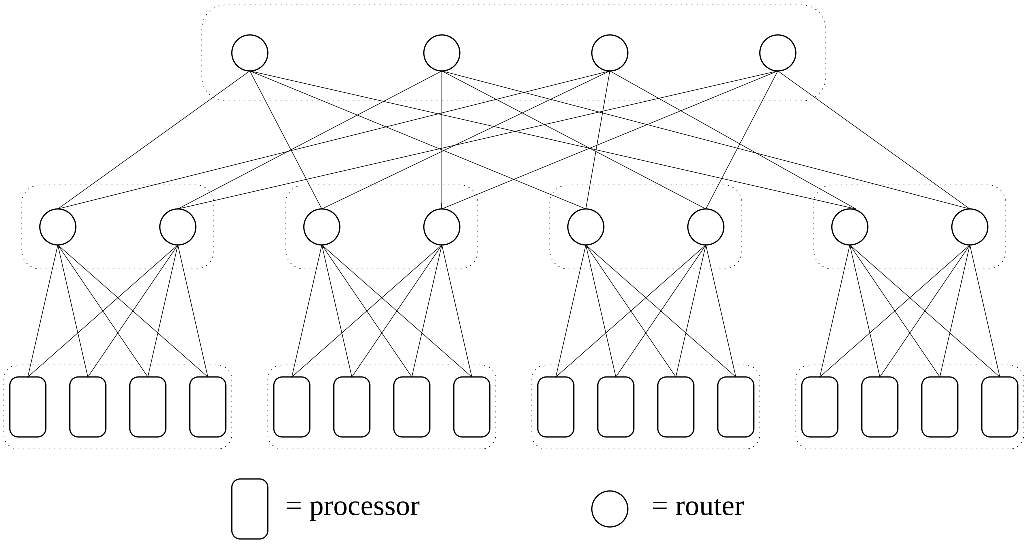

FIG. 3. A fat tree of height 2.

How to embed the underlying graph $G_{ft}(\ell)$ of an ordinary fat tree of height ℓ into a 2-dimensional $SMOB$? If it is required that $\varphi_\ell = \varphi_d = 1$, the only possible shape of the $SMOB$ is $\langle (2^{2\ell+1} - 2^{\ell}) \times 1 \rangle$ (a fat tree of height ℓ consists of $2^{2\ell} - 2^{\ell}$ intermediate nodes and $2^{2\ell}$ leaf nodes (processors)). Consider embedding a fat tree of height 1. All the leaf nodes are connected to both of the root nodes, and therefore a line is the only possible shape. The same repeats at each level. In the following, we present a straightforward adaption of the basic H-tree layout of the fat tree (see e.g., [16, Lecture 25]). Our embedding $\mathcal{E}_{ft}^{2 \mapsto 2}$, will have $\varphi_d = \varphi_c = 2$, but there is a straightforward emulation on $\mathcal{E}_{ft}^{2 \mapsto 2}$ that has $\varphi_t = 2$.

LEMMA 2.12. *There exists such an embedding*

$$\mathcal{E}_{ft}^{2 \mapsto 2} = E\left(G_{ft}(\ell), SMOB(\langle (2^{\ell+1} - 1) \times 2^{\ell} \rangle, 3)\right)$$

that $\varphi_\ell = 1$, $\varphi_d = \varphi_c = 2$ *and* $\varphi_s = 2^{\ell-1}$.

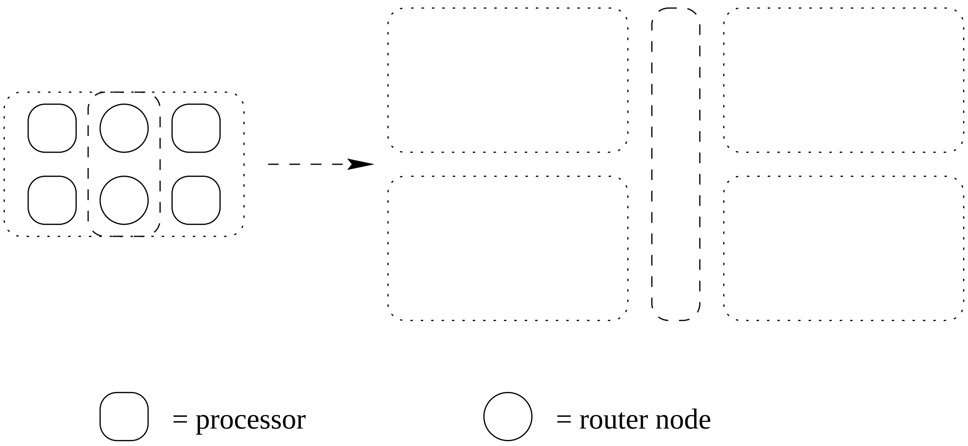

FIG. 4. Packing fat trees to a plane.

Proof. The basic idea of the embedding is described in Figure 4. At each level i, the construction consists of 4 level $i-1$ constructions that are connected to the array of intermediate nodes in the middle. At level i, there are 2^i intermediate nodes (except at level 0 there is only one processor and no intermediate nodes). Thus, the vertical side length of the resulting construction is 2^ℓ. The horizontal side length is $s(i) = 2s(i-1)+1$, which has solution $s(i) = 2^{i+1}-1$, since $s(0) = 1$.

If the intermediate nodes $V_{i,j}$, $0 \leq j < 2^{2\ell-i}$, at each level i are arranged from the top to the bottom in increasing order of j (to arrays $V_{i,x2^i}, \ldots, V_{i,(x+1)2^i}$ for an integer x), then one of the connections from each level $i-1$ intermediate node to level i intermediate node is straight. This can be seen from Definition 2.1. We reserve a channel for both of these "horizontal" connections. The stretch of these straight connections is $s(i-2)+1 = 2^{i-1}$. The other connection from each level $i-1$ intermediate node is drawn via a (unique) node in the same column with the level $i-i$ intermediate node. This increases the load of each horizontal channel by 1, since each level i intermediate node receives exactly 2 connections from the left (right) side. All in all, an intermediate node needs 3 "horizontal" channels but only one "vertical" channel. It is not difficult to determine that the vertical stretch of the two-phase connections is also 2^{i-1}. ■

It is also possible to define a "3-dimensional" variant of the fat tree [15]. The topology of a 3D fat tree is given in Definition 2.2 (it can be seen as a layered cross product[3] of a 2^3-ary tree and a 2^{3-1}-ary tree).

DEFINITION 2.2. A **3-dimensional fat tree** of height ℓ consists of $\ell+1$ levels. At level k (where $0 \leq k \leq \ell$), it has nodes $V_{k,0}, \ldots, V_{k,2^{3\ell-k}-1}$. Thus, a 3D fat tree has

$$\sum_{i=1}^{\ell} 2^{3\ell-i} = 2^{3\ell} - 2^{2\ell}$$

[3]Naturally, it would be possible to consider other kind of "fat trees". In general, an LCP of a k_1-ary tree (downwards) and a k_2-ary tree (upwards) forms a "fat tree", if $k_1 > k_2 \geq 2$.

intermediate nodes and $2^{3\ell}$ leaf nodes. Intermediate nodes, except the $2^{2\ell}$ root nodes, have degree $8+4=12$. The root nodes have degree 8. A node $V_{k,u}$ (where $0 \le k < \ell$ and $0 \le u \le 2^{3\ell-k}-1$) is connected to nodes

$$V_{k+1,(u \ mod \ 2^{2k})+x\times 2^{2k}+(u \ div \ 2^{2k+3})\times 2^{2k+2}},$$

where $x = 0,1,2,3$, with bidirectional links. Thus, a 3D fat tree has altogether $\sum_{i=0}^{\ell-1} 2^2 \times 2^{3\ell-i} = 2^{3\ell+3} - 2^{2\ell+3}$ bidirectional links. The diameter is 2ℓ.

How is the "3-dimensional" variant of the fat tree (see Definition 2.2) embedded into a 3-dimensional $SMOB$? Basically, an embedding with similar structure is easy to construct. The starting point of construction is an 8-processor 3-dimensional fat mesh having 4 intermediate nodes. The processors are arranged to corners of a $3 \times 2 \times 2$ space and the intermediate nodes are arranged to a 2 by 2 plane (between the two groups of corners). The construction is extended similarly by gluing together 8 corner block with one plane of intermediate nodes. Clearly, the resulting construction has size $2^{\ell+1}-1 \times 2^{\ell} \times 2^{\ell}$.

In the initial case, we must implement a connection from each corner node to each of the four intermediate nodes. Thus, $\varphi_d = 3$. (Observe that we are still doing better than Lemma 1.6.) Now, we have two choices: Either to attach two "horizontal" channels to each corner node (emulation requires $max\{2, \varphi_d\}$ steps), or pipeline the 4 virtual channels through one "horizontal" channel (emulation takes 4 steps). We choose the latter option. Thus, to implement the 8 channels, we attach 2 "horizontal" channels to each intermediate node for this purpose. Each intermediate node has 6 packets to forward. It is not difficult to see that by attaching 2 channels per node for both of the remaining axes, the packets can be routed to their target in 3 steps (thus, the emulation succeeds in 4 steps). The above described routing path designing and scheduling succeeds similarly in the higher levels. It is also easy to see that the maximum stretch along each dimension at level i is 2^{i-1}. We state the above in form of Lemma 2.13. There exists an emulation on top of $\mathcal{E}_{ft}^{3\mapsto 3}$ that requires 4 steps.

LEMMA 2.13. *There exists such an embedding*

$$\mathcal{E}_{ft}^{3\mapsto 3} = E\left(G_{ft,3}(\ell), SMOB(\langle(2^{\ell+1}-1)\times 2^{\ell}\times 2^{\ell}\rangle, 3)\right)$$

that $\varphi_\ell = 1$, $\varphi_d = 3$, $\varphi_c = 4$, *and* $\varphi_s = 2^{\ell-1}$.

The discussed embeddings $\mathcal{E}_{ft}^{2\mapsto 2}$ and $\mathcal{E}_{ft}^{3\mapsto 3}$ leave a little to improve. We leave open, whether an ordinary fat tree has an efficient embedding into a 3-dimensional $SMOB$ (considering the case $3 \mapsto 2$ is perhaps not as interesting). One could also consider embedding other kind of fat trees.

3. CONCLUSIONS

We studied embedding of meshes, fat meshes, coated meshes, meshes of trees, hypercubes, butterflies and fat trees into a d-dimensional regular static mesh of optical buses. We observed that in all cases it was possible to find a better embedding (and thus emulation) than what is suggested by the general results of Lemmas 1.5 and 1.6. For all but the fat tree embeddings, $\varphi_d = \varphi_c = 1$ and thus by Lemma 1.1 there exists such an emulation on each of the embeddings that $\varphi_t = 1$. For the embeddings $\mathcal{E}_{ft}^{2\mapsto 2}$ and $\mathcal{E}_{ft}^{3\mapsto 3}$ there exist emulations requiring 2 and 4 steps, respectively.

REFERENCES

1. F. Abolhassan, R. Drefenstedt, J. Keller, W.J. Paul, and D. Scheerer. On the Physical Design of PRAMs. *The Computer Journal*, 36(8):756 – 762, 1993.
2. F. Annexstein, M. Baumslag, and A.L. Rosenberg. Group Action Graphs and Parallel Architectures. *SIAM Journal on Computing*, 19(3):544 – 569, 1990.
3. R. Beigel and C.P. Kruskal. Processor Networks and Interconnection Networks without Long Wires (Extended Abstract). In *Symposium on Parallel Algorithms and Architectures SPAA*, volume 1, pages 42 – 51, 1989.
4. R. Drefenstedt and D. Schmidt. On the Physical Design of Butterfly Networks for PRAMs. In *Proceedings, Fourth Symposium on the Frontiers'92*, pages 202 – 209. IEEE Computer Society Press, USA, 1992.
5. S. Even and A. Litman. Layered Cross Product – A Technique to Construct Interconnection Networks. In *SPAA'92, 4th Annual Symposium on Parallel Algorithms and Architectures, San Diego, California*, pages 60 – 69, June 1992.
6. R.P. Grimaldi. *Discrete and Combinatorial Mathematics – An applied introduction*. Addison-Wesley, 1994.
7. T. Grün, T. Rauber, and J. Röhrig. The Programming Environment of the SB-PRAM. In *Proceedings, 7th IASTED/ISMM International Conference on Parallel and Distributed Computing and Systems*, pages 504 – 509, 1995.
8. T.J. Harris. *Shared Memory with Hidden Latency on a Family of Mesh-like Networks*. PhD thesis, Department of Computer Science, University of Edinburgh, May 1995.
9. R. Heckmann, R. Klasing, B. Monien, and W. Unger. Optimal Embedding of Complete Binary Trees into Lines and Grids. In *Proceedings, 17th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'91), LNCS 570*, pages 25 – 35, 1991.
10. J. Keller. Regular Layouts of Butterfly Networks. *Integration – The VLSI Journal*, 17(3):253 – 263, 1994.
11. R. Koch, T. Leighton, B. Maggs, S. Rao, and A. Rosenberg. Work-Preserving Emulations of Fixed-Connection Networks. In *Proceedings, 21th ACM Symposium on Theory of Computing STOC'89*, pages 227 – 240, 1989.
12. R. Koch, T. Leighton, B. Maggs, S. Rao, A. Rosenberg, and E. Schwabe. Work-Preserving Emulations of Fixed-Connection Networks, 1996. To appear in Journal of the ACM.
13. F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays • Trees • Hypercubes*. Morgan Kaufman, San Mateo, CA, 1992.
14. F.T. Leighton and B. Maggs. The Role of Randomness in the Design of Interconnection Networks. In J. van Leeuwen, editor, *IFIP, Information Processing'92*, volume I, Algorithms, Software, Architecture, pages 291 – 305, 1992.
15. F.T. Leighton, B.M. Maggs, A.G. Ranade, and S.B. Rao. Randomized Routing and Sorting on Fixed-Connection Networks. *Journal of Algorithms*, 17(1):157 – 205, July 1994.
16. T. Leighton, C.E. Leiserson, and R. Blumofe. Theory of Parallel and VLSI Computation, Lecture notes for 18.435J/6.848J. Technical Report MIT/LCS/RSS 18, MIT, Laboratory for Computer Science, July 1992.
17. T. Leighton and B. Maggs. Expanders Might be Practical: Fast Algorithms for Routing Around Faults on Multibutterflies. In *Proceedings, 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 384 – 389, 1989.
18. C.E. Leiserson. Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing. *IEEE Transactions on Computers*, C-34(10):892–900, October 1985.

19. C.E. Leiserson, Z.S. Abuhamdeh, D.C. Douglas, C.R. Feynman, M.N. Ganmukhi, J.V. Hill, W.D. Hillis, B.C. Kuszmaul, M.A. St. Pierre, D.S. Wells, M.C. Wong, S-W. Yang, and R. Zak. The Network Architecture of the Connection Machine CM-5. *SPAA'92, 4th Annual Symposium on Parallel Algorithms and Architectures, San Diego, California*, pages 272 – 285, 1992.

20. V. Leppänen. *Studies on the Realization of PRAM.* PhD thesis, TUCS, Department of Computer Science, University of Turku, November 1996. TUCS Dissertation, No 3.

21. W.F. McColl. General Purpose Parallel Computing. In A M Gibbons and P Spirakis, editors, *Lectures on Parallel Computation. Proc. 1991 ALCOM Spring School on Parallel Computation*, Cambridge International Series on Parallel Computation, pages 337–391. Cambridge University Press, 1993.

22. A.G. Ranade, S.N. Bhatt, and S.L. Johnsson. The Fluent Abstract Machine. Technical Report BA87-3, Thinking Machines Corporation, Technical Report Series, 1987.

23. A.G. Ranade, S.N. Bhatt, and S.L. Johnsson. The Fluent Abstract Machine. In *Proceedings, 5th MIT Conference on Advanced Research in VLSI*, pages 71–93, 1988.

24. S.B. Rao. Properties of an Interconnection Architecture based on Wavelength Division Multiplexing. Technical Report TR-92-009-3-0054-2, NEC Research Institute, Princeton, January 1992.

25. A.L. Rosenberg. Better Parallel Architectures via Emulations. In F. Meyer auf der Heide, B. Monien, and A.L. Rosenberg, editors, *Proc. of Parallel Architectures and Their Efficient Use, First Heinz Nixdorf Symposium, LNCS 678*, pages 30 – 36. Springer-Verlag, 1992.

26. E. Upfal. An $O(\log N)$ Deterministic Packet Routing Scheme. In *Proceedings, 21th Annual ACM Symposium on Theory of Computing, STOC'89*, pages 241 – 250, 1989.

27. L.G. Valiant. General Purpose Parallel Architectures. In *Algorithms and Complexity, Handbook of Theoretical Computer Science*, volume A, pages 943–971, 1990.

28. D.S. Wise. Compact Layouts of Banayan/FFT Networks. In *Proceedings, CMU Conference on VLSI Systems and Computations*, pages 186 – 195, 1981.