ARTICLE IN PRESS

Theoretical Computer Science ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Area-time tradeoffs for universal VLSI circuits☆

Sandeep N. Bhatt [a], Gianfranco Bilardi [b], Geppino Pucci [b,*]

[a] Hewlett-Packard Laboratories, Princeton NJ08540, USA

[b] Dipartimento di Ingegneria dell'Informazione, Università di Padova, 35131 Padova, Italy

ARTICLE INFO

Keywords:
VLSI theory
Area-universal networks
Interconnection networks

ABSTRACT

An area-universal VLSI circuit can be programmed to emulate every circuit of a given area, but at the cost of lower area-time performance. In particular, if a circuit with area-time bounds (A, T) is emulated by a universal circuit with bounds (A_u, T_u), we say that the universal circuit has blowup A_u/A and slowdown T_u/T. A central question in VLSI theory is to investigate the inherent costs and tradeoffs of universal circuit designs.

Prior to this work, universal designs were known for area-A circuits with $O(1)$ blowup and $O(\log A)$ slowdown. Universal designs for the family of area-A circuits containing $O(\sqrt{A^{1+\epsilon}}\log A)$ vertices, with $O(A^{\epsilon})$ blowup and $O(\log\log A)$ slowdown had also been developed. However, the existence of universal circuits with $O(1)$ slowdown and relatively small blowup was an open question. In this paper, we settle this question by designing an area-universal circuit U_A^{ϵ} with $O(1/\epsilon)$ slowdown and $O(A^{\epsilon})$ blowup, for any value of the parameter ϵ, with $4\log\log A/\log A \leq \epsilon \leq 1$. By varying ϵ, we obtain universal circuits which operate at different points in the spectrum of the slowdown-blowup tradeoff. In particular, when ϵ is chosen to be a constant, our universal circuit yields $O(1)$ slowdown.

1. Introduction

Area-universal circuits are VLSI designs that can be programmed to emulate all the circuits of a given area A with bounded loss in area-time performance. The study of area universality can give valuable insights into general purpose multiprocessing, as well as into the exploitation of field programmable gate arrays, and other forms of reconfigurable architectures for VLSI.

Two parameters characterize the quality of a universal circuit. The *blowup*, $\alpha = A_u/A$, where A_u is the area of the universal circuit (also called *host* circuit), and A is the area of the emulated circuit (also called *guest* circuit) measures the hardware cost of a universal design. The *slowdown* $\sigma = T_u/T$, where T_u is the time taken by the host circuit to emulate T steps of the guest circuit, measures the speed penalty incurred. The tradeoffs achievable between α and σ are of great interest.

Area universality has received considerable attention in the literature. The pioneering work of Leiserson [1] introduced the first efficient universal network, the *concentrator fat-tree*, establishing its effectiveness for off-line routing, later extended to randomized on-line routing in [2]. Bay and Bilardi [3] proposed the *pruned butterfly fat-tree* and the *sorting fat-tree* for efficient deterministic on-line routing. Greenberg [4] defined the *pyramid fat-tree* and established its universality properties under various delay models for signal propagation. The blowup and slowdown of the above mentioned constructions are each polylogarithmic. Two constructions achieving constant blowup ($\alpha = O(1)$) and logarithmic slowdown ($\sigma = O(\log A)$)

☆ A preliminary version of this work appeared in *Proc. of the 20th Anniversary Conference on Advanced Research in VLSI*, Atlanta, GA USA, March 1999. This work was done while the first author was with Bellcore, Morristown, NJ USA. The research of the second and of the third author was supported, in part, by MIUR of Italy under Project *MAINSTREAM*, and by the EC under Project 15964 *AEOLUS*.

* Corresponding address: Via Gradenigo 6/B, 35131 Padova, Italy. Tel.: +39 0498277951; fax: +39 0498277799.
E-mail addresses: sandeep.bhatt@hp.com (S.N. Bhatt), bilardi@dei.unipd.it (G. Bilardi), geppo@dei.unipd.it (G. Pucci).

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.08.005

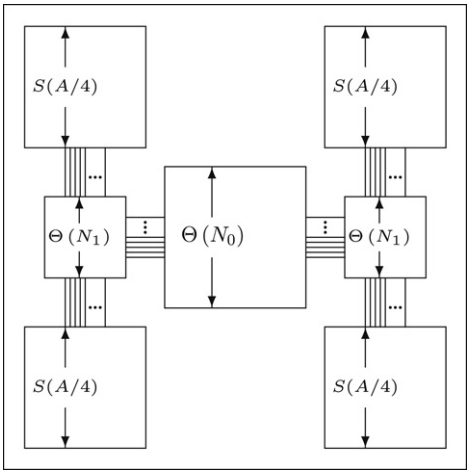

Fig. 1. Recursive definition of the H-layout of U_A^ϵ. $S(A)$ denotes the sidelength of the layout. Parameter N_i is an upper bound on the number of distinct wires entering a given fat-node at distance i from the fat-root.

for on-line routing were given by Leighton, Maggs, Ranade, and Rao [5] and by Bay and Bilardi [6], in the word and in the bit models of VLSI computation, respectively. The papers by Bilardi and Bay [7] and by Bilardi, Chauduri, Dubashi, and Mehlhorn [8] explored the slowdown-blowup tradeoff from the perspective of lower bounds. Fat-tree like networks have been adopted in some commercial multiprocessors (*e.g.*, see [9–11]).

While the results of [5] and [6] show that constant blowup is achievable with polylogarithmic slowdown, a question that has remained open is whether constant slowdown is achievable and, if so, at what price in area. Initial progress in this direction was made by Kaklamanis, Krizanc, and Rao [12], who showed that a butterfly of area $A^{1+\epsilon}$ can simulate any area-A network, further constrained to have $O(\sqrt{A^{1+\epsilon}}\log A)$ vertices, with slowdown $O(\log\log A)$.

In this paper, we exhibit an area-universal circuit U_A^ϵ with blowup $O(A^\epsilon)$, for any chosen positive $\epsilon \geq 4\log\log A/\log A$, and with slowdown $O(1/\epsilon)$. When ϵ is chosen to be a fixed constant, our construction yields constant slowdown.

1.1. An overview of U_A^ϵ

The high level structure of our circuit U_A^ϵ is that of a binary fat-tree of A leaves, whose nodes, henceforth called *fat-nodes*, have size decreasing with their distance from the root. In contrast to most previously proposed fat-trees, where the leaves are in charge of computation and the internal nodes support communication, each fat-node of U_A^ϵ performs both computation and communication functions. Figs. 1 and 2 illustrate the structure and a coarse level layout of the overall fat-tree, and of one fat-node, respectively.

Specifically, a fat-node at level i in the fat-tree is equipped with a number $\ell_i = O\left(\sqrt{A/2^i}\right)$ of so called *emulation trees* of height $h/4$, where $h = (\epsilon/2)\log A$. For emulation purposes, each vertex[1] u of the guest circuit G is handled by a suitably chosen emulation tree T_u, using the technique proposed by Meyer auf der Heide in [13], which can be briefly described as follows. Each node of T_u emulates a single vertex of G, specifically, the root emulates u and if an internal node emulates some vertex v, its children emulate the neighbors of v in G (observe that the same vertex may be emulated by many nodes of the tree). The tree T_u has the following capability: if each of its nodes is initialized with the state of the corresponding guest node at (guest) step t, then T_u can produce at its root the correct state of vertex u at (guest) time $t + h/4$ in $O(h)$ steps. In this process, for $j = 1, \ldots, h/4$, during the simulation of the j-th guest step only the nodes in the top $h/4 - j + 1$ levels of T_u can update their simulated state. After simulating $h/4$ steps, a phase of global communication is needed to send the current state information from the roots of the emulation trees where it has been computed, to the nodes of the emulation trees where it is needed.

The network that accomplishes this state redistribution contains two components. First, in a fat-node ν at level i, the ℓ_i roots of the emulation trees are connected to the roots of a number of *broadcast trees*, which touch all fat-nodes within a neighborhood of ν of radius h and serve the purpose of transporting the state updates for the guest nodes emulated in ν, to those fat-nodes where such updates may be necessary to reinitialize the state of their emulation trees. Second, ν contains a network which connects the broadcast-tree nodes touching ν to the nodes of the emulation trees in ν. This latter network is a *distributor*, that is, a network of switches obtained by a simple adaptation of the Beneš permutation network [14]. The distributor can be programmed to connect each output (here, a node of an emulation tree) to at most one arbitrarily selected input (here, a node of a broadcast tree). As a consequence, a given input can be connected to zero or more outputs.

[1] To avoid confusion, we will henceforth refer to the nodes of the guest circuit G as *vertices* and reserve the term *nodes* to those of the simulating circuit.

Please cite this article in press as: S.N. Bhatt, et al., Area-time tradeoffs for universal VLSI circuits, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.005

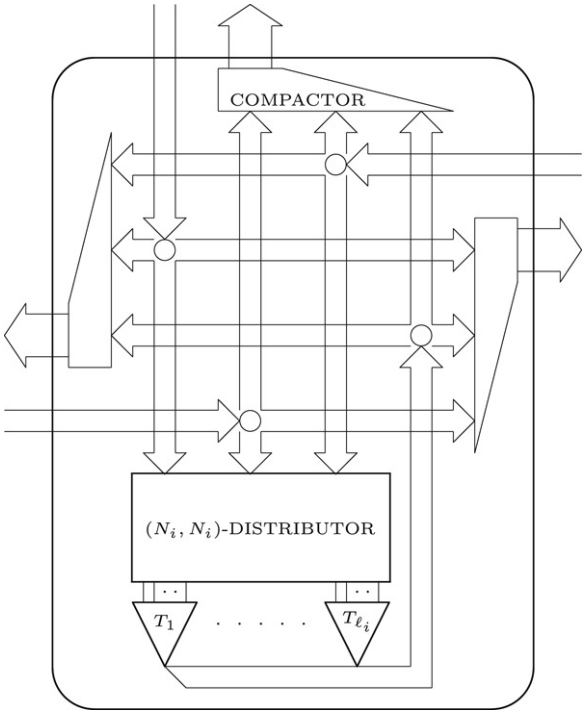

Fig. 2. A fat-node at level i. The upper portion is devoted to the broadcast trees that come from and go to the fat-node itself and its three neighbors (parent, left child, right child). The lower portion contains the ℓ_i emulation trees and the (N_i, N_i)-distributor, which establishes paths from broadcast trees to emulation trees. The roots of the emulation trees are also origins of broadcast trees. The trapezoidal areas (compactors) are wire routing regions that regroup wires continuing toward a given neighbor into consecutive tracks.

In summary, each fat-node contains (i) a number of emulation trees, (ii) one distributor, and (iii) a set of broadcast-tree nodes. Each fat-node is connected to its parent and its two children by *channels*, consisting of a collection of broadcast-tree edges (see Fig. 2).

For the emulation scheme to be feasible, guest vertices must be mapped to fat-nodes, observing two constraints: (a) since each guest node is handled by a single emulation tree, there must be at least as many emulation trees in a fat-node as the number of guest vertices mapped to the fat-node; (b) since each broadcast trees spans a neighborhood of radius h, in order to guarantee that the state updates reach all the emulation trees where they are required, vertices whose distance is at most $h/4$ in the guest circuit must be mapped onto fat-nodes, whose distance is at most h in the fat-tree. In order to establish such a mapping, we make use of a key result of layout theory, first established by Bhatt and Leighton in the bifurcator framework [15].

The rest of the paper is structured as follows. In Section 2, we review elements of bifurcator theory and emulation techniques that play a role in our construction. For added clarity, we describe the design of U_A^ϵ in two stages. First, following the ideas outlined above, in Section 3 we discuss the details of a simplified and slightly less efficient variant of U_A^ϵ, where we assume that there is a distinct broadcast tree for each emulator tree root, and analyze its area requirement in Section 4. Then, in Section 5, we show how to reduce the number of broadcast trees by means of pipelining so to obtain the stated area-time bounds for U_A^ϵ. Finally, Section 6 discusses futher ways of improving our design, while Section 7 offers a few concluding remarks.

2. Background

We will be dealing with graphs which can be laid out in Thompson's grid model for VLSI [16], hence in what follows we will restrict our attention to graphs in which no vertex has degree greater than four. The construction of our universal circuits relies on two main ingredients: "good" embeddings of such graphs into binary trees and emulation techniques based on redundant computation. The following two subsections explore each of these aspects in detail.

2.1. Bifurcators and graph embeddings

Recall that an $(F_0, F_1, \ldots, F_{r-1})$-*decomposition tree* for a graph G is a binary tree T of height r whose nodes are subgraphs of G satisfying the following constraints:

(1) The root of T is G.
(2) Each leaf of T is either the empty subgraph or an isolated node.

Please cite this article in press as: S.N. Bhatt, et al., Area-time tradeoffs for universal VLSI circuits, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.005

(3) For $0 \leq i < r$, let G_i be a tree node at level i and let G_i^0 and G_i^1 be the children of G_i. Then G_i can be partitioned into G_i^0 and G_i^1 by removing no more than F_i edges.

In [15], Bhatt and Leighton introduced the notion of *bifurcator* of a graph. Specifically, a graph G has an (F, α)-bifurcator if it has an $(F_0, F_1, \ldots, F_{r-1})$-decomposition tree with $r = \log_\alpha F + 1$ and $F_i = F/\alpha^i$, for $0 \leq i < r$. Of particular interest for layout theory are $(\cdot, \sqrt{2})$-bifurcators, as demonstrated by the following fact.

Fact 1 ([15]). *A graph G graph representing a circuit of layout area A has a $(\sqrt{A}, \sqrt{2})$-bifurcator.*

Recall that an *embedding* of a graph $G = (V_G, E_G)$ into a graph $H = (V_H, E_H)$ is a (many-to-one) mapping $\varphi : V_G \to V_H$ of the vertices of G to the nodes of H and a mapping of the edges of G to the paths in H, such that any edge $(u, v) \in E_G$ is mapped to a simple path in H whose endpoints are $\varphi(u)$ and $\varphi(v)$. When H is a tree, the path corresponding to edge (u, v) is completely determined by φ as the unique simple path in H with endpoints $\varphi(u)$ and $\varphi(v)$.

For a given embedding, the *load* of a node $v_H \in V_H$ is the number of vertices of G mapped to v_H; the *congestion* of an edge $(u_H, v_H) \in E_H$ is the number of paths realizing edges of G that pass through (u_H, v_H); and, finally, the *dilation* of an edge $(u_G, v_G) \in E_G$ is the length of the path in H corresponding to (u_G, v_G). The maximum value of each of these quantities is called load, congestion and dilation of the embedding, respectively. The following theorem is an immediate consequence of Theorem 7 of [15] and highlights an important relation between graph bifurcators and tree embeddings:

Theorem 2. *If a graph G has an $(F, \sqrt{2})$-bifurcator, then there is an embedding of G into a complete binary tree of height $\lceil 2 \log F \rceil$ with the following properties:*

(1) *Each tree node at level i has load at most $c_\ell F/\sqrt{2^i}$, for a suitable fixed constant $c_\ell > 0$.*
(2) *Each tree edge connecting a node at level i with one of its children has congestion at most $c_c F/\sqrt{2^i}$, for a suitable fixed constant $c_c > 0$.*
(3) *Each graph edge has dilation no greater than 4.*

The following is an immediate corollary of Fact 1 and Theorem 2.

Corollary 3. *Any graph G with a layout of area A can be embedded into a complete binary tree of A leaves with dilation at most 4, load $\ell_i \leq c_\ell \sqrt{A/2^i}$ at tree nodes at level i, and congestion $c_i \leq c_c \sqrt{A/2^i}$ at tree edges connecting a node at level i with one of its children, for suitable fixed constants $c_\ell, c_c > 0$.*

2.2. Emulations through redundant computation

In addition to the embedding technique reviewed in the previous subsection, our emulation crucially exploits the use of redundant computation, following an approach proposed in [13] for the emulation of arbitrary bounded-degree networks. Informally, we allow that a single vertex of G be emulated by a suitably large subset of nodes of H. During the emulation, some nodes in the subset may be "left behind", in the sense that their state is outdated with respect to the state of the most advanced replica in the subset. In order to "refresh" the state of lagging nodes, from time to time during the emulation all the necessary information is routed from the replicas that are up to date to those that are out of date. Constant slowdown can be achieved if the refreshing operations can be performed seldom enough that their cumulative cost is at most proportional to the number of steps of the computation being emulated.

Constant-slowdown emulations through redundant computations were first obtained by Meyer auf der Heide in [13] for processor networks. More specifically, he designed a family of bounded-degree networks $\mathcal{M} = \{M_{c,N}^\epsilon\}$ with the following property: for any value of $c, \epsilon > 0$ and arbitrary values of N, $M_{c,N}^\epsilon$ has degree $c + 4$, $O\left(N^{1+\epsilon} \log N\right)$ nodes, and can emulate any N-vertex network G of degree c with $O(c + 1/\epsilon)$ slowdown. In his construction, each vertex of G to be emulated is associated with a fixed subset of $O(N^\epsilon)$ nodes of $M_{c,N}^\epsilon$.

Since our construction borrows some ideas from the scheme of [13], we recall the main features of such a scheme. The overall structure of $M_{c,N}^\epsilon$ is very simple: its basic constituents are a set of N c-ary trees of height t, and an $(N, N \cdot c^t)$-*distributor*. The latter network has N distinguished inputs and $N \cdot c^t$ outputs, coinciding, respectively, with the roots and the leaves of the N trees. By definition, a distributor can be prepared off-line, to realize any communication pattern where each input sends the same message to a subset of outputs and each output receives exactly one message from one of the inputs. In [13], it is shown that a simple variation of the Beneš network [14] with b inputs and outputs and $b \log b$ nodes is an (a, b)-distributor for any $a \leq b$, with distribution time $O(\log b + t)$ when messages have size $O(t)$.

When emulating a given N-vertex network $G = (V_G, E_G)$ of degree c, each tree node is statically associated with a node of V_G as follows: each node u of V_G is associated with the root of a distinct tree T_u. Within T_u, if a tree node is associated with a vertex $v \in V_G$, then $g \leq c$ of its children are associated with the $g \leq c$ neighbor nodes of v in G, while the others are inactive (note the large degree of replication implied by such an assignment). It is straightforward to see that if each node of T_u contains the state of its associated graph vertex at time s, then in t steps the root of the tree is able to compute the state of vertex u at time $s + t$. Observe that at the end of the emulation the states of the nodes of T_u at level ℓ (the level of the root being $\ell = 0$) are "left behind" to time $s + t - \ell$.

Please cite this article in press as: S.N. Bhatt, et al., Area-time tradeoffs for universal VLSI circuits, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.005

The emulation algorithm proceeds in *phases*, each phase emulating t steps of G. At the beginning of Phase i, $i = 0, 1, \ldots$, each tree node contains the state at time $i \cdot t$ of its associated graph vertex. Then the emulation in each tree takes place, so that, after t steps, the N tree roots produce the sequence of state updates at times $i \cdot t + 1, \ldots, (i+1) \cdot t$ of the vertices in V_G. The distributor is configured in such a way that, for each $u \in V_G$, the root of T_u can pipeline the sequence of t state updates occurred during the phase to all the emulator tree nodes associated with u. In an additional t steps, any tree node is therefore able to compute the state of its associated graph vertex at time $(i+1) \cdot t$. Each phase requires time $O(tc + \log N)$. By choosing $t = \epsilon \log_c N$, we obtain $O(c + 1/\epsilon)$ slowdown and an overall number of $O\left(N^{1+\epsilon} \log N\right)$ nodes in $M^{\epsilon}_{c,N}$.

We remark that the need to transmit strings of t state updates arises when emulating processor networks, where the state of each vertex can be rather large. In our VLSI context, vertex states are essentially boolean values, making it sufficient to transmit the final value to each node associated to that vertex in any emulation tree.

Note that since a circuit of area A features at most A vertices of degree 4, $M^{\epsilon}_{4,A}$ can emulate any such circuit with constant slowdown. However, $M^{\epsilon}_{4,A}$ has a large area of at least $\Omega\left(A^{2(1+\epsilon)}\right)$ [17], since it contains a Beneš permutation network [14] with $\Theta\left(A^{1+\epsilon}\right)$ inputs as a subgraph. In the following sections we combine some ideas from [13] with the bifurcator-based properties of a circuit of area A, to obtain a universal circuit whose area is roughly the square root of the area of $M^{\epsilon}_{4,A}$.

3. Structure of the circuit and emulation algorithm

As mentioned in the previous section, the large area of $M^{\epsilon}_{4,A}$ is due to the presence of a full-blown distributor with $\Theta\left(A^{1+\epsilon}\right)$ outputs. Beside having at most A nodes of degree at most 4, a circuit of area A exhibits additional structure, as captured by the embedding derived from the bifurcator, in Corollary 3. This structure will let us substitute the large distributor of $M^{\epsilon}_{4,A}$ with several ones of much smaller size, with a considerable reduction of the overall area requirement.

Following the bifurcator-induced embedding, the basic structure of our universal circuit U^{ϵ}_{A} is that of a complete binary tree of *fat-nodes*, of height $\log A$. A fat-node at level i, $0 \le i \le \log A$, contains ℓ_i emulator trees (*i.e.*, 4-ary trees) of height $h/4$, where ℓ_i is the upper bound on the value of the load at level i in the tree-embedding of an area A circuit, and h is an integer parameter whose value will be chosen as a function of A and ϵ to govern the blowup-slowdown tradeoff of U^{ϵ}_{A}, with larger h (that is, larger ϵ) yielding a faster but larger universal circuit. The fat-node also contains a distributor of suitable size, which is used as in [13] to restore the current state in all the nodes of the emulator trees at certain times during the emulation.

Consider now an arbitrary circuit G of area A. The universal circuit is prepared for the emulation of G by assigning one emulator tree to each vertex of G. The assignment follows the embedding of G into the tree, in the sense that the emulator tree for vertex u is chosen among those residing in fat-node $\varphi(u)$. Note that an emulator tree T_u of $u \in V_G$, may contain nodes associated with vertices in V_G whose emulator trees reside in fat-nodes different from $\varphi(u)$. However, recall that the embedding provided by Corollary 3 has dilation at most 4. Since the vertices in V_G associated with nodes of T_u have distance at most $h/4$ from u in G, it follows that their corresponding emulator trees reside in fat-nodes at distance at most h from $\varphi(u)$, in the fat-tree underlying the universal circuit. We call this set of fat-nodes an *h-neighborhood* of $\varphi(u)$. Clearly, the roots of the emulator trees within an h-neighborhood of $\varphi(u)$ are the only ones that need to be connected to the nodes of the emulator trees in $\varphi(u)$ through the local distributor. The following lemmata quantify the number of distinct emulator trees in an h-neighborhood of a fat-node as a function of its level in the tree.

Lemma 4. *Let $L_{i,j}$ be the number of vertices of G embedded within a subtree of height j rooted at a fat-node at level i, with $0 \le i \le \log A$ and $0 \le j \le \log A - i$. Then*

$$L_{i,j} = O\left(\sqrt{2^j}\sqrt{A/2^i}\right).$$

Proof. The load of a fat-node at level $i+s$ in the tree is at most $\ell_{i+s} = c_\ell\sqrt{A/2^{i+s}}$. Therefore

$$L_{i,j} \le \sum_{s=0}^{j} 2^s c_\ell \sqrt{A/2^{i+s}} \le \left(c_\ell\sqrt{2}/(\sqrt{2}-1)\right)\sqrt{2^j}\sqrt{A/2^i}. \quad \square$$

Lemma 5. *Let N_i be the total number of vertices of G embedded in fat-nodes of the h-neighborhood of a fat-node at level i, with $0 \le i \le \log A$. Then*

$$N_i = O\left(h\sqrt{2^h}\sqrt{A/2^i}\right).$$

Proof. Consider a fat-node v at level i, with $0 \le i \le \log A$. Its h-neighborhood includes the two subtrees of height $\min\{\log A - i, h-1\}$, rooted at its children (if any), the first $\min\{i,h\}+1$ fat-nodes on the path from v to the fat-root and, for the s-th such node, the fat-nodes of a subtree of height $h-s-1$ (see Fig. 3). Therefore,

$$\begin{aligned} N_i &\le 2L_{i+1,\min\{\log A-i-1,h-1\}} + \sum_{s=0}^{\min\{i,h\}} \ell_{i-s} + \sum_{s=1}^{\min\{i,h-1\}} L_{i-s+1,h-s-1} \\ &\le \sqrt{A/2^i}\left(7c_\ell\sqrt{2^{\min\{\log A-i,h-1\}}} + 4c_\ell\sqrt{2^{\min\{i,h\}}} + 2c_\ell \min\{i,h-1\}\sqrt{2^h}\right) \\ &= O\left(h\sqrt{2^h}\sqrt{A/2^i}\right). \quad \square \end{aligned}$$

Please cite this article in press as: S.N. Bhatt, et al., Area-time tradeoffs for universal VLSI circuits, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.005

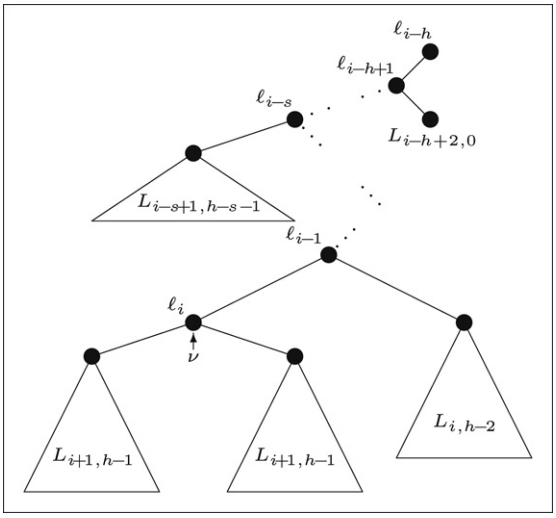

Fig. 3. The h-neighborhood of a fat-node ν at level i, with $h \le i \le \log A - h$.

In order to restore the correct state in all the nodes of the emulator trees within fat-node ν, *all* the roots of the emulator trees in the h-neighborhood of ν must be connected to the inputs of the local distributor, while *all* the nodes of the emulator trees in the fat-node must be connected to its outputs. Prior to the emulation, the distributor in each fat-node is prepared so that the root of each emulator tree T_u residing in the h-neighborhood can broadcast the sequence of state updates in the current phase to all the nodes associated with vertex u.

Taking a symmetric perspective, every root of an emulator tree in ν must be connected to *all* the distributors in the h-neighborhood of ν. For the sake of simplicity, let us assume for now that such connections are realized by having each root of an emulator tree be also the root of a dedicated *broadcast tree* that spans the h-neighborhood, and whose nodes are connected to the distributors. Note that there are ℓ_i distinct broadcast trees rooted at each fat-node. In Section 5, we will discuss how to employ pipelining to reduce the number of broadcast trees rooted at a fat-node, which will improve the blowup by a polylogarithmic factor in A.

From the above observations it follows that the distributor in a fat-node ν at level i has $a_i = N_i = O\left(h\sqrt{2^h}\sqrt{A/2^i}\right)$ inputs and $b_i \le (4/3)4^{h/4}c_\ell\sqrt{A/2^i} = O\left(\sqrt{2^h}\sqrt{A/2^i}\right)$ outputs. Since N_i is a factor $\Theta(h)$ greater than b_i (although no more than b_i inputs will ever be active in any emulation), an (N_i, N_i) distributor is needed to realize the required multicast operation.

Once the universal circuit is prepared for G, the emulation proceeds similarly to the network emulation of [13]. Specifically, each phase emulates $h/4$ steps of G as follows. First, in each fat-node, the emulator trees compute the initial state of their roots for the next phase. Then, a root associated with node $u \in V_G$ broadcasts its state to all the nodes of its broadcast tree, and from there through the distributors to all the nodes of the emulator trees associated with node u in the h-neighborhood of $\varphi(u)$. At this point, all the nodes of the emulator trees are ready for the next phase. Altogether, the overall time to emulate $h/4$ steps is $O(h + \log A)$. When $h = (\epsilon/2)\log A$, we obtain a slowdown of $O(1/\epsilon)$. (The factor $1/2$ in the definition of h may appear redundant in Section 4 whereas it does play a role in the construction of Section 5.)

4. Area of the universal circuit

Let us first consider the layout of a fat-node ν at level i. The area of the layout is dominated by the area of the N_i-input Beneš network realizing the local distributor. Also, no more than N_i broadcast trees are incident on ν. Hence ν admits a square layout of area $O\left(N_i^2\right)$, where the three communication channels needed to route the broadcast trees from/to ν to/from its father and its two children are incident on three distinct sides of the layout. The high-level organization of the resulting layout is shown in Fig. 2.

Being tree-structured, the circuit admits an H-tree layout [17] where the nodes of the H-tree are the layouts of the corresponding fat-nodes, and the wire channels between pairs of nodes are wide enough to route all the edges of the broadcast trees traversing the channel. The recursive structure is shown in Fig. 1.

Let $S(A)$ be the side length of the resulting H-layout. From the above considerations we conclude that

$$\begin{aligned} S(A) &= \sum_{i=0}^{\log_4 A} 2^i N_{2i} \\ &= \sum_{i=0}^{\log_4 A} O\left(\epsilon\sqrt{A^{1+\epsilon/2}}\log A\right) \\ &= O\left(\epsilon\sqrt{A^{1+\epsilon/2}}\log^2 A\right), \end{aligned} \tag{1}$$

Please cite this article in press as: S.N. Bhatt, et al., Area-time tradeoffs for universal VLSI circuits, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.005

hence the overall area requirement of the resulting design is $O\left(\epsilon^2 A^{1+\epsilon/2}\log^4 A\right)$. In the next section, we show how to reduce the area bound of U_A^ϵ by reducing the number of incident broadcast trees and the size of the distributor at each fat node.

5. Improving the area bound through pipelining

The careful reader has probably observed that both the broadcast trees and the distributors of the construction presented in Section 3 are somewhat underutilized. In fact, only one stage of them is active at any given time, which is the natural scenario for improving performance via pipelining.

The idea behind the use of pipelining is quite simple. Specifically, we partition the ℓ_i roots of the emulator trees of a fat-node at level i into $\lceil \ell_i/h \rceil$ groups of (at most) h roots each and let each group use a single broadcast tree to distribute state updates to their h-neighborhood. Simple calculations suffice to show that the number of distinct broadcast trees incident on a node at level i is now

$$N_i' = O\left(\sqrt{2^h}\sqrt{A/2^i} + 2^h\right)$$

(the second term accounts for the number of distinct fat-nodes in an h-neighborhood of a fat node at level i). Also, N_i' is still an upper bound on the total number of nodes of the ℓ_i emulator trees residing in the fat-node. As a consequence, equipping the fat node with an (N_i', N_i')-distributor is still sufficient to perform the required multicast operation in $O(h)$ time, by pipelining h batches of state updates. As a consequence, the slowdown is still $O(h + \log A) = O(1/\epsilon)$.

Let us now consider the area requirement of this improved design. The area of the layout of a fat-node at level i is still dominated by the area of the (N_i', N_i')-distributor. Observe that each switch of the distributor must now accommodate the extra space required for storing the direction bits needed for routing the h batches of state updates. However, this is achievable by adding only $O\left(h \log N_i'\right) = O\left(N_i'\right)$ grid lines to one side of the standard layout of the Beneš networks (h lines per stage) to place storage elements for the required control bits. As a consequence, this augmentation yields a layout with both sides of length still linear in N_i'.

By plugging the new values N_i' into Eq. (1) we obtain

$$S(A) = \sum_{i=0}^{\log_4 A} 2^i N_{2i}' = O\left(\sqrt{A^{1+\epsilon/2}}\max\{\log A, A^{\epsilon/4}\}\right),$$

hence the overall area requirement of our final universal circuit U_A^ϵ is $O\left(A^{1+\epsilon/2}\max\{\log^2 A, A^{\epsilon/2}\}\right) = O\left(A^{1+\epsilon}\right)$ whenever the parameter ϵ is such that $4\log\log A/\log A \le \epsilon \le 1$.

We have thus proved the main result of this paper, stated in the following theorem.

Theorem 6. *For any A and ϵ such that $4\log\log A/\log A \le \epsilon \le 1$, there is an area-universal VLSI circuit U_A^ϵ of area $O\left(A^{1+\epsilon}\right)$ which can be programmed to emulate any VLSI circuit of area A with slowdown $O(1/\epsilon)$.*

The above construction is valid even for nonconstant values of the parameter ϵ. For instance, setting $\epsilon = 4\log\log A/\log A$ yields a circuit of area $O\left(A\log^2 A\right)$, which can emulate all circuits of area A with $O(\log A/\log\log A)$ slowdown. This area is the same as that of the *concentrator fat-tree* of [1] while the slowdown is a $\Theta(\log\log A)$ factor smaller.

6. Further directions of improvement

It is straightforward to check that, in terms of (blowup, slowdown) bounds, most previous universal circuits (*e.g.*, [1,2,12,3]) are subsumed by some instantiation of our circuit U_A^ϵ. However, our flexible design is not able to match the constant blowup and logarithmic slowdown networks of [5] (word model) and [6] (bit model), since the circuit of Theorem 6 does not admit a realization with constant blowup.

It is natural to wonder whether our construction can be further improved to yield a suitable parametrized universal circuit, whose (blowup, slowdown) bounds can smoothly range from $(\alpha = O(1), \sigma = O(\log A))$ to $(\alpha = O(A^\epsilon), \sigma = O(1/\epsilon))$. In fact, there is still a number of enhancements that can be applied to our construction, which yield extra area savings and push down the minimum value of ϵ, for which significant tradeoffs can be achieved, to $O(1/\log A)$. However, since the improved design still falls short of subsuming the circuit in [6], here we discuss the necessary modifications only at a high level, and leave the details to the interested reader.

In the first place, observe that the pipelining feature introduced in Section 5 is underutilized whenever $h = o(\log A)$ (*i.e.*, nonconstant values of ϵ), since the cost of distribution at the root of the fat tree is always $\Omega(\log A)$. Therefore, we can afford that $\Theta(\log A)$ (rather than $\Theta(h)$) roots of emulator trees use the same broadcast tree. Also, observe that the h-neighborhood of fat-nodes grows exponentially smaller as we move towards the leaves, hence we can have shallower emulation trees for vertices embedded closer to the leaves, at the cost of more frequent redistributions, whose total cost would still be amortized by the number of steps being emulated.

Pushing the level of pipelining to $\Theta(\log A)$ requires a more compact layout for a fat-node at level i. An area $O\left((N_i/\log A)^2 + N_i\log N_i\right)$ can be obtained by employing the *Cube-Connected-Cycles* (CCC) interconnection of [18], featuring $N_i/\log A$ cycles of $\log A$ nodes each and capable of performing the required distribution in $O(\log A)$ time. Finally, in order to

Please cite this article in press as: S.N. Bhatt, et al., Area-time tradeoffs for universal VLSI circuits, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.005

avoid wasting area towards the leaves of the H-layout, we must stop the embedding stated in Corollary 3 at a level i', where the total number of vertices that would be embedded in a subtree rooted at level i' is about $\log^2 A$, and simply map all these vertices to the fat-nodes at level i'. A simple mesh layout is then sufficient for these fat-nodes.

To quantify the impact of the above changes, simple calculations show that for $\epsilon = 1/\log A$ the resulting universal circuit features an area of $O(A \log\log A)$ with slowdown $O(\log A)$. The blowup is a mere factor $O(\log\log A)$ worse than the one achieved by the circuit of [6] with the same slowdown.

Further improvements might possibly be obtained by blending the techniques of [6] with those of the current paper, but the required adaptations do not appear to be straightforward. In fact, in [6] the area-A layout is partitioned into $\log A \times \log A$ cells, whose internal communications are emulated by small meshes of the universal circuit and whose external communications are emulated by a concentrator fat-tree that has the meshes placed at its leaves. To achieve locality within the small meshes, it is crucial that *all* the vertices of the emulated circuit be embedded there, at the leaves of the fat-tree. This is in contrast with the approach of the current paper, where some vertices of the emulated circuit are also embedded in the internal fat-nodes, to achieve locality (constant dilation) within the fat-tree.

7. Conclusion

We have developed the first efficient area-universal circuit with constant slowdown. In fact, our construction is more general, since it embodies a design parameter ϵ and yields slowdown $\sigma = O(1/\epsilon)$ and area blowup nearly proportional to A^ϵ, namely $\alpha = O(A^\epsilon)$, within the range $4 \log\log A/\log A \leq \epsilon \leq 1$. Therefore, the area blowup is exponential in the inverse of the slowdown.

More substantial reductions of the area of the universal circuit, possibly leading to polylogarithmic blowup for constant slowdown, pose a considerably more challenging problem, which appears to require significant progress in our current understanding of constant slowdown simulations of all bounded degree networks.

The ideas developed here might find, most likely after careful fine tuning, applications to the design of versatile and/or reconfigurable hardware architectures. In fact, the potential to simulate any architecture of a given area, guarantees that a reconfigurable architecture with universal properties can execute any task nearly as efficiently as any circuit specialized for that task. A particularly interesting direction to explore in this context is the possibility of developing automatic tools to efficiently map any given application on an area-universal circuit.

References

[1] C. Leiserson, Fat-trees: Universal networks for hardware-efficient supercomputing, IEEE Transactions on Computers C-34 (10) (1985) 892–901.
[2] R. Greenberg, C. Leiserson, Randomized routing on fat-trees, in: S. Micali (Ed.), Randomness and Computation, JAI Press, 1989, pp. 345–374.
[3] P. Bay, G. Bilardi, Deterministic on-line routing on area-universal networks, Journal of the ACM 42 (3) (1995) 614–640.
[4] R. Greenberg, The fat-pyramid and universal parallel computation independent of wire delay, IEEE Transactions on Computers C-43 (12) (1994) 1358–1364.
[5] F. Leighton, B. Maggs, A. Ranade, S. Rao, Randomized routing and sorting on fixed-connection networks, Journal of Algorithms 17 (1) (1994) 157–205.
[6] P. Bay, G. Bilardi, An area-universal VLSI circuit, in: Proc. of the 1993 Symp. on Integrated Systems, 1993, pp. 53–67.
[7] G. Bilardi, P. Bay, An area lower bound for a class of fat-trees, in: Proc. of the 2nd European Symposium on Algorithms, in: LNCS, vol. 855, Springer-Verlag, 1994, pp. 413–423.
[8] G. Bilardi, S. Chauduri, D. Dubhashi, K. Mehlhorn, A lower bound on area-universal graphs, Information Processing Letters 51 (1994) 101–105.
[9] C. Leiserson, Z. Abuhamdeh, D. Douglas, C. Feynman, M. Ganmukhi, J. Hill, W. Hillis, B. Kuszmaul, M.St. Pierre, D. Wells, M. Wong, S.-W. Yang, R. Zak, The network architecture of the Connection Machine CM-5, in: Proc. of the 4th ACM Symp. on Parallel Algorithms and Architectures, 1992, pp. 272–285.
[10] Q.S. World, CS-2 architecture overview, Tech. Rep., QSW, Bristol, UK, 1996.
[11] F. Petrini, W. Feng, A. Hoisie, S. Coll, E. Frachtenberg, The quadrics network: High-performance clustering technology, IEEE Micro 22 (1) (2002) 46–57.
[12] C. Kaklamanis, D. Krizanc, S. Rao, Universal emulations with sublogarithmic slowdown, in: Proc. of the 33th IEEE Symp. on Foundations of Computer Science, 1993, pp. 341–350.
[13] F. Meyer auf der Heide, Efficient simulations among several models of parallel computation, SIAM Journal on Computing 15 (1) (1986) 106–119.
[14] V. Beneš, Optimal rearrangeable multistage connecting networks, Bell System Technical Journal 43 (1964) 1641–1656.
[15] S. Bhatt, F. Leighton, A framework for solving VLSI graph layout problems, Journal of Computer and System Sciences 28 (2) (1984) 300–342.
[16] C. Thompson, A complexity theory for VLSI, Ph.D. Thesis, Dept. of Computer Science, Carnegie-Mellon University, Ph.D. Thesis, Tech. Rep. CMU-CS-80-140, Aug. 1980.
[17] T. Lengauer, Combinatorial Algorithms for Integrated Circuit Layout, John Wiley & Sons, Chichester, UK, 1990.
[18] F. Preparata, J. Vuillemin, The cube-connected-cycles: A versatile network for parallel computation, Communications of the ACM 24 (5) (1981) 300–309.

Please cite this article in press as: S.N. Bhatt, et al., Area-time tradeoffs for universal VLSI circuits, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.005