Unifying Mesh- and Tree-Based Programmable Interconnect

André DeHon, *Member, IEEE*

***Abstract*—We examine the traditional, symmetric, Manhattan mesh design for field-programmable gate-array (FPGA) routing along with tree-of-meshes (ToM) and mesh-of-trees (MoT) based designs. All three networks can provide general routing for limited bisection designs (Rent's Rule with $p < 1$) and allow locality exploitation. They differ in their detailed topology and use of hierarchy. We show that all three have the same asymptotic wiring requirements. We bound this tightly by providing constructive mappings between routes in one network and routes in another. For example, we show that a (c, p) MoT design can be mapped to a $(2c, p)$ linear population ToM and introduce a corner turn scheme which will make it possible to perform the reverse mapping from any (c, p) linear population ToM to a $(2c, p)$ MoT augmented with a particular set of corner turn switches. One consequence of this latter mapping is a multilayer layout strategy for N-node, linear population ToM designs that requires only $\Theta(N)$ two-dimensional area for any p when given sufficient wiring layers. We further show upper and lower bounds for global mesh routes based on recursive bisection width and show these are within a constant factor of each other and within a constant factor of MoT and ToM layout area. In the process we identify the parameters and characteristics which make the networks different, making it clear there is a unified design continuum in which these networks are simply particular regions.**

***Index Terms*—Butterfly fat tree (BFT), fat pyramid, fat tree, field-programmable gate-array (FPGA), hierarchical, hierarchical synchronous reconfigurable array (HSRA), interconnect, Manhattan, mesh, mesh-of-trees (MoT), multilevel metallization, Rent's rule, tree-of-meshes (ToM).**

I. Introduction

In the design of field-programmable gate-arrays (FPGAs), we have seen mesh based (e.g., [1]–[5]), hierarchical (e.g., [6]–[11]), and hierarchical mesh interconnection networks (e.g., [12], [13]). We have seen numerous studies showing the characteristics of these networks, how they scale, and empirically how they relate on particular designs (e.g., [7], [9], [13]). In this paper, we examine how they relate in a more fundamental manner. We ask if we can provide any guaranteed bounds on the size of one network given a routing solution in another network (e.g., given a mesh-of-trees (MoT) route, how much larger or smaller can the tree-of-meshes (ToM) route be? Mesh route?). We further ask if there are design variables that allow us to tune the design space between two different network types. This allows us to underscore the ways in which these networks are fundamentally similar and the ways in which they are fundamentally different.

A. Why Does This Matter?

Each of these networks has interesting and useful properties. We would like to know which of these properties are mutually exclusive and which properties might be simultaneously achieved in a single network. For example:

- the ToM network can be placed entirely by recursive bisection;
- the MoT network can be laid out in $\Theta(N)$ two-dimensional (2-D) area when given sufficient metal layers for routing;
- since all three networks are ultimately embedded in a mesh, the optimal mesh placement and routing will achieve minimum wire lengths or minimum total wiring.

We have some evidence that certain combinations are not possible. For example, the traditional switch population schemes for meshes require asymptotically more switches than the MoT or ToM. So, we want to know the following.

- Is it possible to achieve the simplified placement and routing of the ToM simultaneously with the MoT layout guarantee?
- How much does the strict recursive bisection placement which the ToM uses cost compared to an optimal mesh placement? Is it an asymptotically larger costs or just a constant larger?
- The MoT achieves asymptotically fewer switches than the mesh; does it require asymptotically more wiring?
- The MoT can exploit 2-D locality better than the ToM; what does this locality exploitation cost us?

By performing these equivalence mappings, we can answer these questions and show when it is necessary to compromise one good network property for another and when it is possible to achieve good properties simultaneously in a single network design.

Whether or not recursive bisection alone is sufficient for placement is an important question in system-level interconnect prediction. This shows up both in questions about the adequacy of recursive bisection in constructive placements [14] and in questions about the relationship between the pre-placement and post-placement Rent parameters [15], [16] [see (1)]. These relations help us provide, at least, an asymptotic answer. We show that the pre-placement partitioning implies a constructive layout and wiring which need never be more than a constant factor greater than the optimal, post-placement wiring. This gives us insight into the validity of using pre-placement partitioning to predict wiring requirements; the constant factor gap

Manuscript received September 10, 2003; revised February 15, 2004. This work was funded in part by the Defense Advanced Research Project Agency (DARPA) Moletronics Program under Grant ONR N00014-01-0651 and by the National Science Foundation CAREER Program under Grant CCR-0133102.

The author is with the Department of Computer Science, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: andre@cs.caltech.edu).

Digital Object Identifier 10.1109/TVLSI.2004.834237

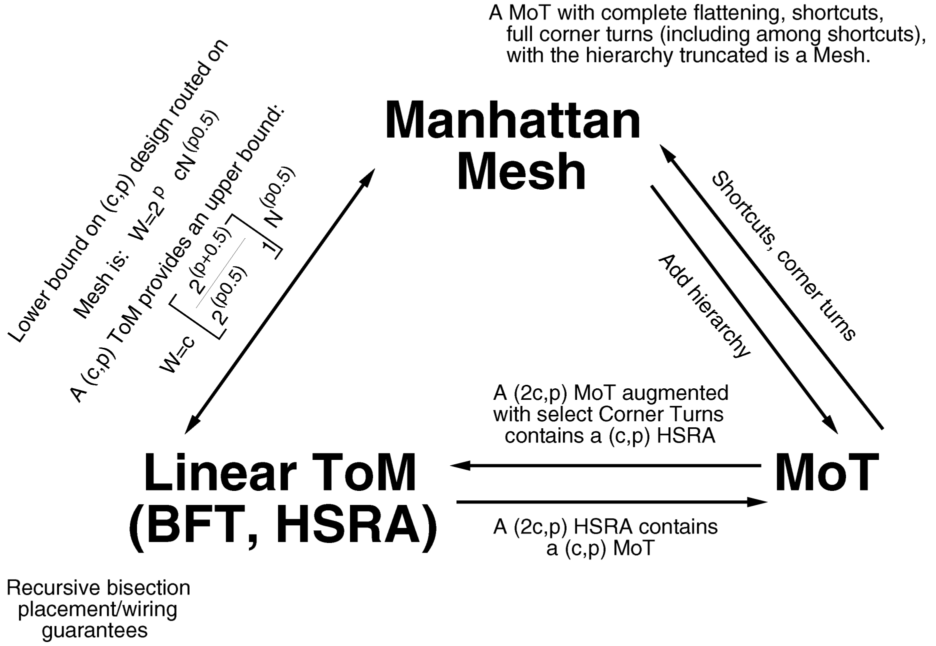


Fig. 1. Equivalence relation summary.

between the upper and lower bound helps us understand the source of variance in pre- versus post-placement wiring. This further helps us understand the wiring tradeoffs which will be involved when using fast-placement techniques for run-time placement of FPGAs [17], [18].

In earlier work [10], we showed that a properly balanced FPGA-style network may leave some compute elements unused in order to better utilize the expensive wiring. We demonstrated this on a ToM network. The question remained whether or not this result was transferable to more conventional mesh networks. Tessier later showed that it was [19]. When we know the fundamental equivalence relations between networks, we will know when results such as this must necessarily transfer between networks.

These equivalences allow us to establish the asymptotic relationships among the networks. As we consider the implications of exponentially scaling chip and system capacities, it is important to keep an eye on which resource requirements will fundamentally diverge and which are only constant factors apart. For example, the constant factor of additional wire capacity in the MoT or ToM may be a worthwhile expense to avoid paying the asymptotically growing switch requirements in conventional mesh designs.

B. Overview

Fig. 1 summarizes our key results. We start by briefly reviewing Rent's Rule and the three network styles which anchor the edges of our design space (Section II). We then show how the MoT is contained in a linear population ToM (Section III). We observe that a particular corner turn scheme, which only increases the switches in the MoT by a constant factor, gives us a MoT that contains a linear population ToM (Section IV); this shows us that the designs are within a constant factor of each other and identifies exactly the change we need to make to one network to turn it into the other. This also shows us how to layout a linear population ToM in constant area per endpoint when given sufficient metal layers—giving us a design which can both be placed easily and efficiently. We also use the MoT layout to demonstrate that the MoT and linear switch population ToM networks require, at most, a constant factor more wire channels than the mesh. The MoT/linear-ToM layout shows us an upper bound for mesh channels which is only a constant factor larger than the lower bound. This further shows the cost of the easy placement and good layout for the ToM or augmented-MoT is bounded even in wire costs compared to the mesh. We then identify the set of parameterized differences between a pure MoT and a traditional Mesh showing that there is a continuum design between these two points (Section V). Combining bounds transitively, we see that all three networks require asymptotically the same number of wires. We note that pure ToM designs require only recursive bisection for layout, so these results suggest the wiring benefit of full mesh placement versus recursive bisection placement can be, at most, a constant factor effect.

II. Background

All three network types—meshes, MoT, and ToM—are instances of limited-bisection networks. That is, rather than supporting *any* graph connectivity, like a crossbar or Beneš network, these networks are designed to exploit the fact that a typical N-node circuit or computing graph can be bisected (cut in half) by cutting less than $O(N)$ hyperedges. This is significant as the bisection width of a network, BW, directly places a lower bound on the size of the network when implemented in VLSI [20]. With a crossbar or Beneš network, the bisection width is $\Theta(N)$, as is the subsequent bisection of each half of the network. This means the horizontal and vertical width of the design, when implemented in a constant number of metal layers, must be $\Omega(N)$ which implies $\Omega(N^2)$ VLSI layout area. In contrast, a network which only has $\mathrm{BW} < O(N)$ bisection width may be implemented in less area as noted below.

A. Limited Bisection Model: Rent's Rule

A common way of summarizing the wiring requirements for circuits is Rent's Rule [21]. Landman and Russo articulate this model for relating the number of gates N and the total number of input and outputs signals, IO

$$\mathrm{IO} = cN^p. \tag{1}$$

This relationship assumes we attempt to maximize locality, i.e., we select the groups of N gates so as to minimize the number of signals (IO) which connect gates in a group to gates in other groups. Rent's observation was that this relation can be tuned to model the IO requirements for all such well chosen subgroups $N < N_{\max}$. Here c and p are parameters that can be tuned to fit the IO versus N connectivity relationship for a design; c is a constant factor offset which roughly corresponds to the IO size of the primitive elements (gates, look-up tables) in a design, and p defines the growth rate. We can view p as a measure of locality. With $p = 1$, we have a design that has $\Theta(N)$ bisection bandwidth, and hence, has little locality. Note that any group of N gates with bounded fanin, k, will have at most $(k+1) \times N$ IOs to cut even if all their nets enter and exit the partition. As p decreases, more of the possible signal nets are contained in the partition; the design has more locality and admits to smaller implementations. Landman and Russo, and a large body of subsequent work, observe that typical designs have $0.5 \leq p \leq 0.75$. Rent's Rule gives us a way of succinctly characterizing the wiring requirements for typical, limited-bisection designs.

Strictly speaking, (1) captures the dominant asymptotic behavior of the design and the real IO versus group size relationship typically diverges from this at the high and low ends. Landman and Russo called the broad region where (1) held Region I and the top end where it no longer holds Region II. Stroobandt identified the divergence at the low end as Region III [22]. Because of this effect, the c in (1) may be different from the actual primitive element IO in order to better fit the Region I relationship.

Returning to our bisection based area lower bound, we can observe that

$$\mathrm{BW(chip\ half)} = \mathrm{IO(chip\ half)} = \mathrm{IO}\left(\frac{N}{2}\right). \tag{2}$$

If we place half of the primitive elements in our design on each half of the chip, (2) reminds us that the total number of wires entering each half of the chip is related to the number of primitive elements in each half. To the extent our Rent relation (1) properly captures the IO versus gate relationship, we can use it to determine the number of wires that must cross the bisection of the chip

$$\mathrm{BW(chip\ half)} = IO\left(\frac{N}{2}\right) = c\left(\frac{N}{2}\right)^p. \tag{3}$$

All of these wires must cross a line that runs the width of the chip and divides the chip into two pieces. We can use this relationship to get a lower-bound on the length of this line and hence the side length of the chip [see Fig. 2]. Without loss of generality, we assume this line is a vertical cut of the die. Then

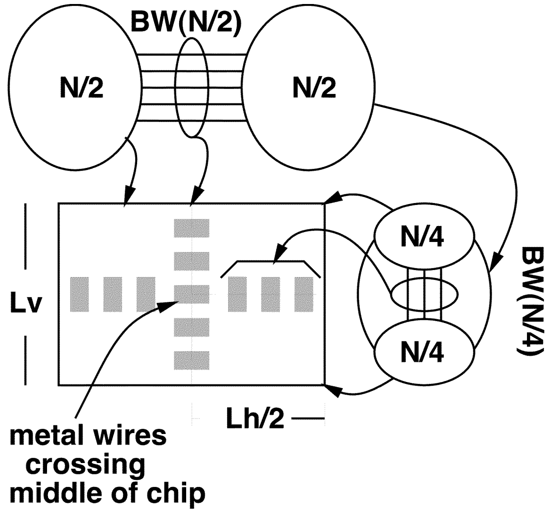


Fig. 2. Area lower bound based on bisection widths.

$$L_v > \frac{W_{\mathrm{pitch}} \times \mathrm{BW(chip\ half)}}{N_v}. \tag{4}$$

Here, W_{pitch} is the wire pitch, and N_v is the number of metal layers available for vertical wiring. Equation (4) says the best we can do is pack the wires crossing the bisection densely at the metal pitch into the available wiring layers.

Now that we have a bound on the vertical length of the chip, we can make a similar argument to bound the horizontal length of the chip. We consider cutting each half of the chip with a horizontal line that runs from the edge of the chip to the vertical cut line (see Fig. 2). This line produces two groups of size $N/4$ in each half of this chip half. Our Rent relation tells us the number of wires leaving each of these halves

$$BW(\mathrm{chip\ quarter}) = \mathrm{IO}\left(\frac{N}{4}\right) = c\left(\frac{N}{4}\right)^p. \tag{5}$$

This allows us to make a similar argument about the length of these horizontal lines

$$\frac{L_h}{2} > \frac{W_{\mathrm{pitch}} \times BW(\mathrm{chip\ quadrant})}{N_h}. \tag{6}$$

We can then put these two lower bounds together to compute a lower bound on the area of the chip due to wiring

$$\begin{aligned} A_{\mathrm{wire}} &> L_v \times L_h \\ &> \left(\frac{(W_{\mathrm{pitch}})^2}{N_v \times N_h}\right) \times \mathrm{BW}\left(\frac{N}{2}\right) \times 2 \times \mathrm{BW}\left(\frac{N}{4}\right) \\ &> \left(\frac{(W_{pitch})^2}{N_v \times N_h}\right) \times c\left(\frac{N}{2}\right)^p \times 2c\left(\frac{N}{4}\right)^p \\ &> \left(\frac{(W_{\mathrm{pitch}})^2}{N_v \times N_h}\right) \times \left(\frac{2c^2}{8^p}\right) N^{2p}. \end{aligned} \tag{7}$$

Equation (7) gives us a lower bound on the wiring requirements for *any* layout of a graph with Rent characteristics (c, p). That

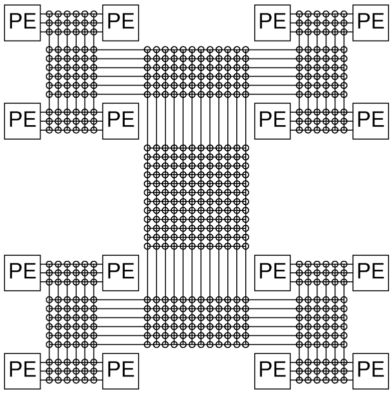


Fig. 3. ToM topology (shown as $c = 3, p = 0.5$).

is, any physical network which supports such a graph must have at least this much wiring area.

To be more precise, we can allow the IOs out of each region to exit on all four sides rather than just the one for the bisection. Note, however, that the regions will be roughly square in order to maximize the perimeter to area ratio, so the total perimeter is always a small constant factor times the edge length. Consequently, such refinements will not change the asymptotic wiring requirements. This wiring area lower bound argument is adapted from Thompson [20].

Equation (7) says our total wiring requirements are growing faster than linearly in gate count when $p > 0.5$. For any fixed number of wiring layers, N_h and N_v, this means wiring area forces the chip area to grow faster linearly for $p > 0.5$. If we allow the number of wiring layers to grow with N, perhaps we can keep A_{wire} down to an area linear in N. Our ability to do this will depend critically on our switch requirements and how the switches and wires are laid out.

B. ToM, BFT, HSRA

Leighton introduced the ToM (see Fig. 3, [23]) as a stylized, limited-bisection network which could be used as a template for the layout of any limited-bisection design and could be the basis of a configurable routing network [24]. Bhatt and Leighton use (α, F) as their parameterization rather than Rent's Rule's (c, p), but they define an equivalent space ($F = c\,(N_{\max})^p$, $\alpha = 2^p$, where $N_{\max}$ is the total number of primitive elements in the design). By tuning the child to parent channel width growth of each of the tree stages, the ToM can be parameterized to support the (c, p) wiring requirements for any circuit. Significantly, if we can recursively partition a design so that its IO versus partition size relationship does not exceed the (c, p) of a ToM network, a $(4c, p)$ ToM network will always be able to route it. Using asymptotically the same number of switches, but organizing them differently, the factor of four can be reduced. Using a crossbar type interpretation of the ToM, a $(3c/2, p)$ network supports the (c, p) design [25]. The ToM allows us to do placement only by considering recursive bisections; this is a powerful property that simplifies physical mapping.

Leiserson adapted the ToM into the fat tree [26] and defined a linear switch population version which he called the butterfly fat tree (BFT) [27] (see Fig. 4, left-hand side). Our hierarchical synchronous reconfigurable array (HSRA) [11] (see Fig. 4 right) is logically equivalent to a BFT. Both are "linearly populated" in that they have only a linear number of switches (linear in the number of child input channels) in each hierarchical switchbox rather than the quadratic number required by the full ToM (Fig. 3). One consequence of linear population is that the BFT or HSRA requires a total number of switches which is linear in the number of endpoints supported for any $p < 1$. In previous work we have identified resource and routability tradeoffs for this class of networks [25]. In [28], we showed that a $p = 0.5$ BFT could be laid out in $\Theta(N)$ area using $\Theta(\log(N))$ metal layers. However, we left open the question of whether or not a $p > 0.5$ BFT could be laid out in $\Theta(N)$ area given sufficient metal layers. Our ToM to MoT mapping (Section IV) shows that we can provide such a layout for any p, demonstrating that we have a network that can both be placed simply by using recursive bisection and be laid out in asymptotically optimal area using multilevel wiring.

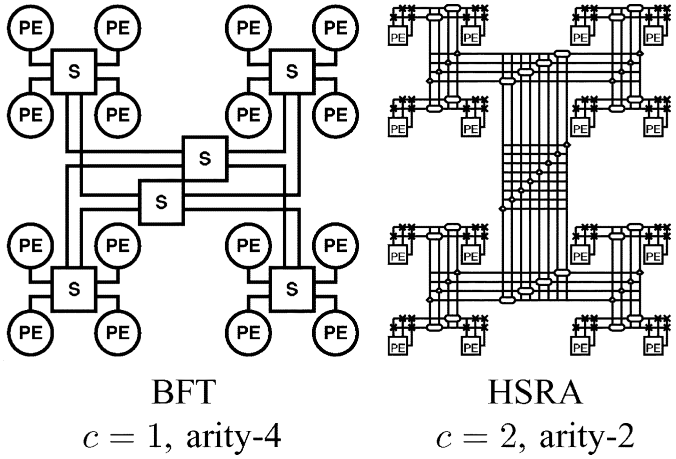


Fig. 4. BFT and HSRA topology.

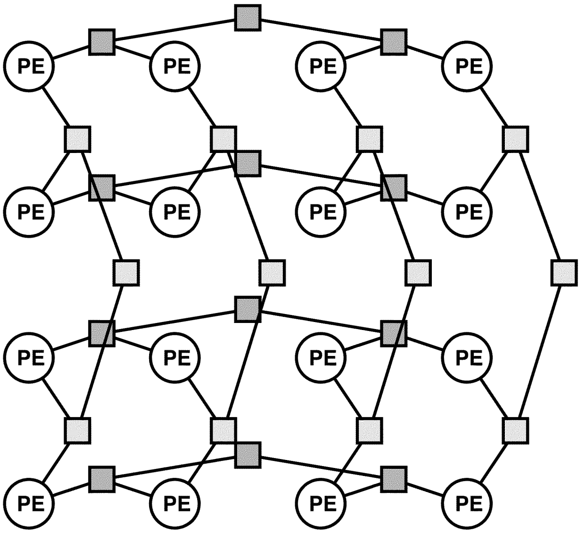


Fig. 5. Basic MoT topology (shown $c = 1, p = 0.5$).

C. MoT

Leighton also introduced the MoT [29], [23] (see Fig. 5). While the ToM–BFT–HSRA style designs have a single tree hierarchy, the MoT starts with a mesh of nodes and builds a

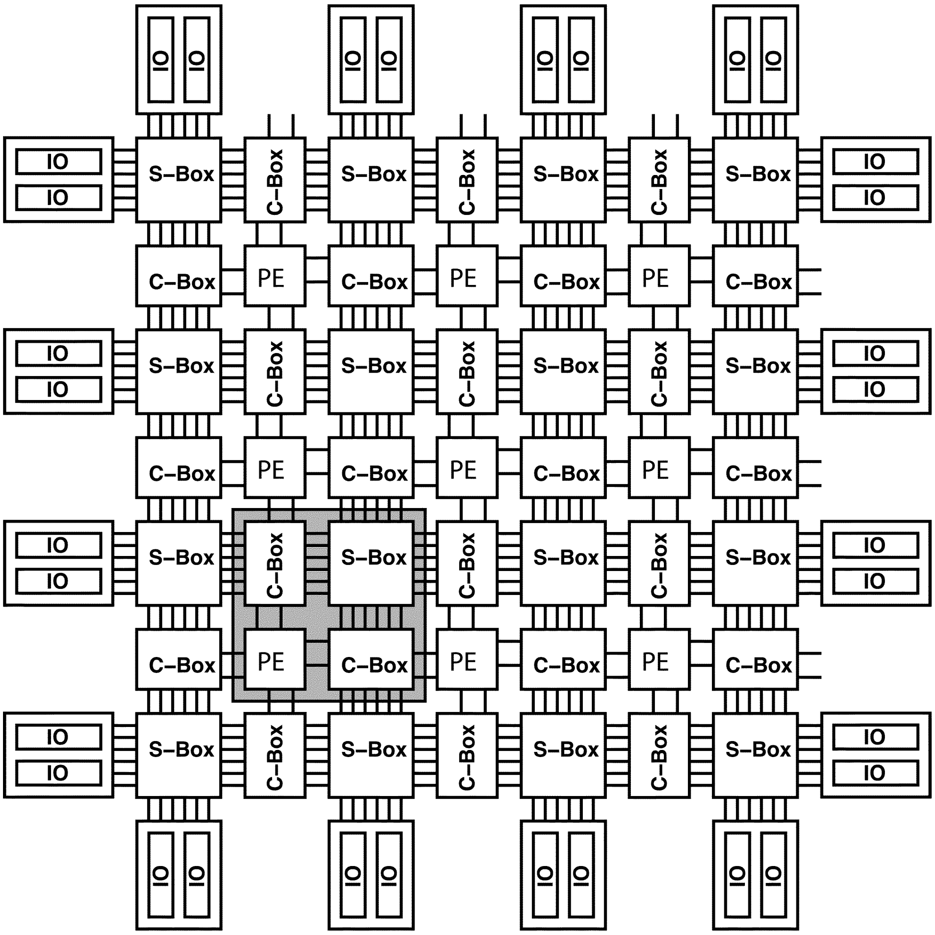


Fig. 6. Manhattan mesh interconnect model (shown as $W = 6$).

separate hierarchical network along each row and column in the mesh. The resulting network has the asymptotic linear switch property of the BFT/HSRA with greater ability to exploit two-dimensional locality. In recent work [13], we identified how to parameterize the MoT for any Rent-style wiring requirements (c, p), calculated switch and layout asymptotics, and showed that the MoT required fewer switches than conventional Manhattan mesh designs. We further showed a multilayer layout for the MoT that required only $\Theta(N)$ area for any p. [13] was the first to demonstrate constant area, multilevel metallization layouts for any of these limited-bisection networks.

D. Manhattan Mesh

Manhattan Meshes (see Fig. 6) have been most heavily studied as interconnect networks for FPGAs (symmetric [4], island-style [5]). These place a routing channel containing W wire track between every row and column of processing elements. Each node may connect to a subset of the wire tracks adjacent to it through the connection box (C-Box). At the intersection of rows and columns, there is a switch box (S-Box) which allows wires to be linked together into longer signal runs or make Manhattan corner turns between a row and a column. Traditional designs have populated the switch boxes linearly in the number of channels, W.

A mesh arranged as $\sqrt{N} \times \sqrt{N}$ primitive elements has $\sqrt{N}+1$ horizontal and vertical channels. The total bisection width of the mesh in the horizontal or vertical direction is then

$$\mathrm{BW}_{\mathrm{mesh}} = \left(\sqrt{N}+1\right) W. \tag{8}$$

To support a design characterized by Rent Parameters (c, p), the Mesh will need

$$\mathrm{BW}_{\mathrm{mesh}} \geq c\left(\frac{N}{2}\right)^{p}. \tag{9}$$

Equation (9) is the same observation as (3)—the IO out of each half of the chip must cross the bisection. Combining (9) and (8), we can related W to the number of primitive elements, N, and our Rent parameters

$$\left(\sqrt{N}+1\right) W \geq c\left(\frac{N}{2}\right)^{p}. \tag{10}$$

For simplicity, we can drop the plus one without affecting the asymptotic implications

$$\begin{aligned}\left(\sqrt{N}\right) W &\geq c\left(\frac{N}{2}\right)^{p}\\ W &\geq \left(\frac{c}{2^{p}}\right) N^{(p-0.5)}.\end{aligned} \tag{11}$$

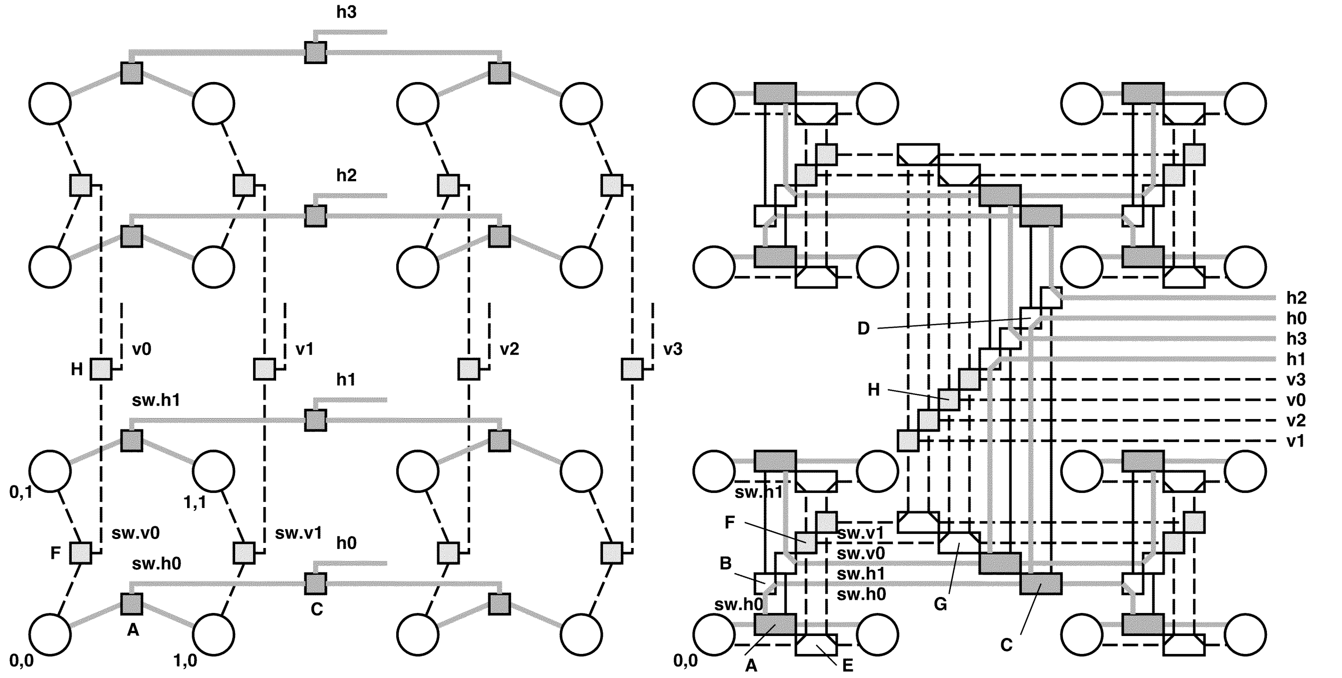


Fig. 7. Mapping between $p = 0.5$ MoT and $p = 0.5$ HSRA (BFT).

Equation (11) gives us a lower bound on channel width which a mesh will need to support a Rent characterized (c, p) design. The mesh will generally need more wire channels than this because

- this only charges for bisection wires – channels may need to be wider to hold wires in the recursive cuts;
- this assumes optimal wire spreading – it may not be possible to spread wires evenly across all channels without increasing channel widths in the orthogonal channels.

Significantly, note that the mesh channel width, W, grows with N for $p > 0.5$. This is directly related to the same asymptote we saw in (7). The total wiring requirement grows faster than linear for $p > 0.5$. In [13], we further showed that switch and area requirements must grow faster than linear in N when $p > 0.5$ for traditional mesh organizations; since switches occupy space in the substrate, this prevents us from using multilayer metallization in order to fully combat the superlinear wire growth requirements; the density of our meshes will asymptotically decrease with increasing N.

While the mesh is perhaps the most studied and most commercially exploited topology, it has the worst asymptotic growth requirements of the three designs. This is a salient example where it is important to know that the asymptotic savings which the MoT and ToM achieve is at the cost of only a constant factor in wiring compared to the mesh; this says that even if the mesh design is smallest at a particular time, if the Rent relation continues to hold while chip and system capacities continue to grow, the mesh will eventually be the largest design.

III. Mapping MoT to Linear Population ToM

We start by showing there is a direct mapping between a MoT with a given growth rate p and an HSRA/BFT with the same growth rate. Leighton observed that a MoT could be embedded in a ToM [23] where both are implicitly assumed to have $p = 0.5$. We observe that the mapping holds even if the ToM is linearly populated with switches as in the BFT or HSRA and that the mapping will hold for $0.5 \leq p \leq 1.0$.

A. Mapping

The observation is simply that we can embed each horizontal MoT tree inside a single HSRA tree (See Fig. 7). Note that the horizontal tree connecting the lowest row of the MoT (trace $(0, 0) \rightarrow \mathrm{sw}.h0 \rightarrow h.0$) is mapped to a corresponding HSRA tree (marked with same labels). Switches A and C perform the same roles in both trees. HSRA switches B and D are set into a fixed configuration as shown so that switches A and C (and corresponding switches higher in the tree) are connected together to match the MoT topology.

Similarly, we can embed each vertical MoT tree inside a single HSRA tree. Here switches E and G in the HSRA link up switches F and H in the HSRA so they can serve as switches F and H in the MoT.

In both cases, switches in alternate tree stages in the HSRA are simply switched into a static position (e.g., B, D, E, and G, in the called-out example) to match the topology of the MoT, while the other tree switches directly provide the switching needed by the MoT (e.g., A, C, F, and H). The MoT and the HSRA both support arbitrary c values using multiple, disjoint trees—disjoint except at the leaf where they connect to the leaves. Since we use two HSRA trees to support each MoT tree, we see that every $(2c, p)$ HSRA contains within it a (c, p) MoT. Assuming the same arity (number of children links per switchbox; see Section V-A3), a MoT route will traverse twice as many switches when implemented on the HSRA.

B. Implications

This observation says that the number of base trees (c) required for a BFT/HSRA can never be more than a constant factor larger than that for a MoT. The factor of two in leaf channels will manifest themselves as a factor of two in both the horizontal and vertical width of the HSRA, or a factor of 4 total area due to channel width. Both designs require a number of switches which is linear in the number of endpoints nodes and c. This shows that the c's will be linearly related so the total switches will be within a constant factor of each other. One might have expected that the MoTs fuller exploitation of 2-D locality would have given it an asymptotic advantage compared to the ToM; this shows the advantage is, at most, a small constant factor.

The factor of two is an upper bound. The mapped route does not fully use the switches in the HSRA (e.g., B, D, E, G), rather it takes a route which exists in the MoT based on less switching options. As a result, it is likely that any given design will route with a smaller constant factor on the HSRA ($c_{\mathrm{hsra}} < 2c_{mot}$).

This shows that if one were to come up with a particularly clever or fast way to place or route a MoT, there would be a direct way to use it for the BFT/HSRA. That is, we simply solve the problem quickly for the MoT, then use the equivalence embedding given in the previous section to identify the switch settings in the ToM necessary to implement the MoT route.

C. Technicalities

1) Leaf Composition: For the mapping to work directly, the HSRA must allow connections between trees in each leaf similar to MoT corner turns. A typical MoT network connection will route through both a horizontal and vertical tree, changing between a horizontal and vertical tree (a corner turn) at a common leaf node. Consequently a MoT route mapped to an HSRA will need to be able to exit one tree route at a leaf, switch to a different tree, and continue routing in that tree.

2) Matching Growth Rates (p's): For the simplest HSRA's and MoTs, we use arity-2 trees, and we approximate a given p by deciding whether each tree stage has single or multiple parents (e.g., in the HSRA shown on the right of Fig. 4, the lowest level tree switches have two parents, while the switches one level up have a single parent). In the single tree HSRA, for arity-2 we repeat base sequences of growths (g_i's)

$$N^p = \left(2^l\right)^p = 2^{lp} = g_0 \times g_1 \times g_2 \times \cdots g_k. \qquad (12)$$

So, for $p = 0.5$, we use the sequence (2 1)*, for $p = 0.75$, the sequence (2 2 2 1)*. For the MoT, we have separate trees in every channel contributing to the total bisection bandwidth, and each growth spans both dimensions, so we have

$$N^p = \left(\left(2^l\right)^2\right)^p = 2^{2lp} = 2^l \times g_0 \times g_1 \times g_2 \times \cdots g_l. \qquad (13)$$

The sequence (1)* realizes $p = 0.5$, and the sequence (2 1)* realizes $p = 0.75$. Redistributing the 2's

$$2^{2lp} = 2 \times g_0 \times 2 \times g_1 \times 2 \times g_2 \times \cdots \times 2g_l. \qquad (14)$$

From this, we see that given a MoT growth sequence $g_{\mathrm{mot}} = \{g_{\mathrm{mot}_0}, g_{\mathrm{mot}_1}, \ldots g_{\mathrm{mot}_k}\}^*$, we can create an HSRA growth sequence

$$g_{\mathrm{hsra}} = \{2, g_{\mathrm{mot}_0}, 2, g_{\mathrm{mot}_1}, \ldots 2, g_{\mathrm{mot}_l}\}^*. \qquad (15)$$

That is, the directly corresponding HSRA sequence includes a two before every growth factor in the MoT sequence. Thus our $p = 0.5$, (1)* MoT sequence yields our (2 1)* HSRA sequence, and our $p = 0.75$, (2 1)* MoT sequence yields our (2 2 2 1)* HSRA sequence. This arises because the MoT always effectively doubles its bandwidth in the nontree dimension simply by aggregating all the tree wires in the orthogonal channels. These are exactly the wires which have fixed switch configurations in the mapping above (see Fig. 7).

One consequence of this is that the directly mapped HSRA growth sequence for a given p corresponds to the HSRA growth sequence derived from the MoT sequence. In many cases this is the same (e.g., for $p = 0.5$: $\mathrm{MoT}(1)^* \rightarrow \mathrm{HSRA}(2\,1)^*$, for $p = 0.75$: $\mathrm{MoT}(2\,1)^* \rightarrow \mathrm{HSRA}(2\,2\,2\,1)^*$). However, for some sequences there is a simpler growth sequence which one might use on the HSRA. For example, for $p = 2/3$, the simplest MoT sequence is (2 1 1)*. The corresponding mapped HSRA sequence is (2 2 2 1 2 1)*. However, the sequence (2 2 1)* is a simpler growth sequence often used for the HSRA. If we do not use corresponding sequences in the mapping, the embedding may require a larger ratio between c_{hsra} and c_{mot}. Nonetheless, the ratio will remain a constant.

IV. Mapping Linear Population ToM to MoT

Embedding the MoT in the HSRA made it clear that the MoT has a subset of the connectivity of the HSRA. We want to identify exactly what the difference between these two networks is. *What do we have to add to transform the MoT into the HSRA*?

A. MoT Augmentation

Figs. 8 and 10 show that we need to add a strategic set of orthogonal interchanges to the trees of a single dimension of the MoT in order to achieve HSRA-equivalent connectivity. As shown in Fig. 9, we decompose the MoT into horizontal and vertical channels and concentrate on additions to the horizontal channels. We add vertical links between corresponding switching nodes in different channels (see Fig. 8). Here, "corresponding" means that a switching node at level l is connected to the switching node at the same logical tree point (same logical set of decisions among up links when there is growth) 2^l channels above (below) it.

The additional wires turn the single child per side, single parent switching nodes into 5-way switches instead of 3-way switches [the 3-way switches in Fig. 9 (S_3) turn into 5-way switches like the one shown on the right of Fig. 10 (S_5)], and turn the double parent switches into 6-way switches [see bottom level switches in the MoT on the top right of Fig. 12 (S_6)]. As shown in Fig. 10, we can reorganize the HSRA switching so that it fits inside these augmented MoT switching units while retaining all of the HSRA connectivity. This switch regrouping

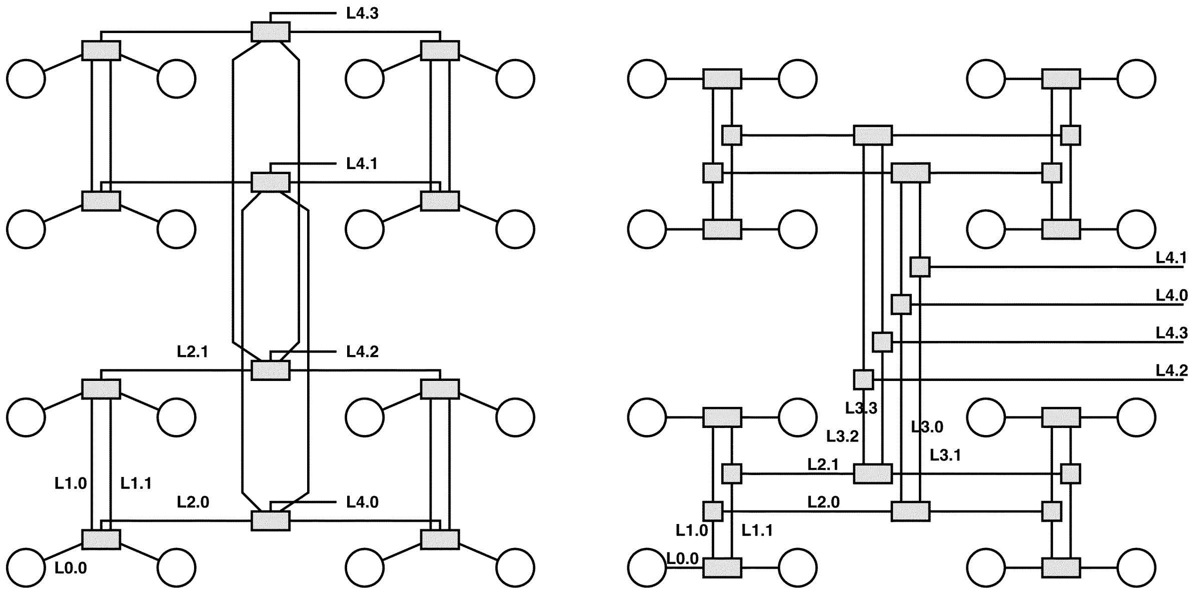


Fig. 8. Mapping between $p = 0.5$ HSRA and $p = 0.5$ augmented MoT.

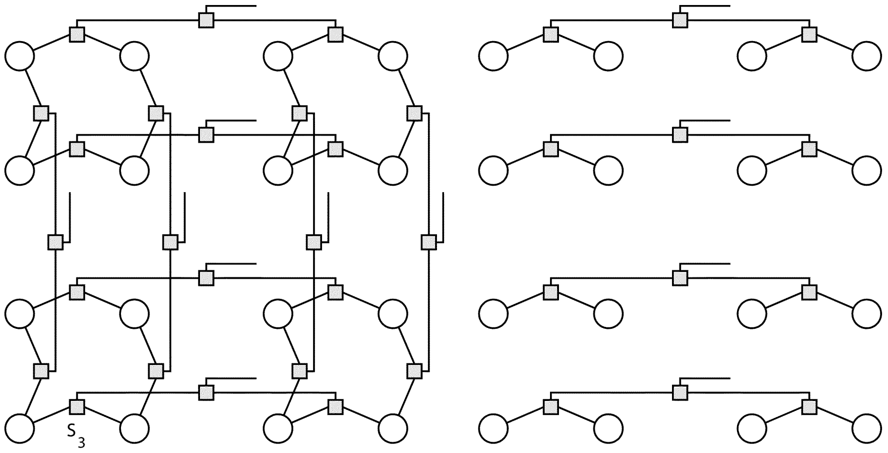


Fig. 9. Extract horizontal connectivity from $p = 0.5$ MoT.

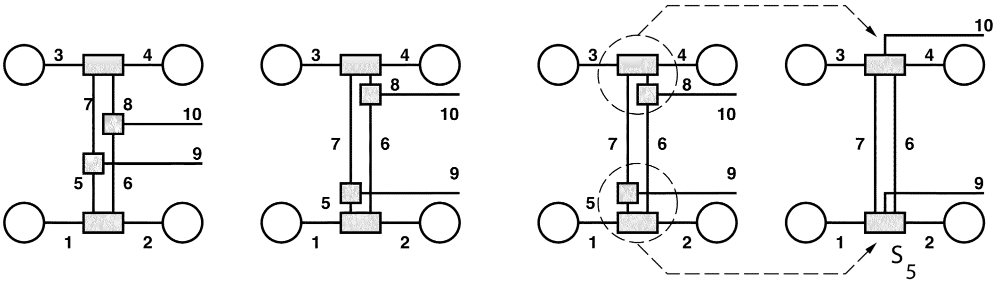


Fig. 10. Equivalence of augmented MoT switching and HSRA switching.

added to the vertical link topology recovers for us HSRA connectivity for any size HSRA. Fig. 8 marks the resulting wire correspondence.

In this transformation we simply replace every existing switching unit with one which is a constant factor larger. The net effect is to increase the total number of switches by a constant factor. The total number of switches required for this augmented MoT remains linear in the number of endpoints supported.

B. MoT Layout

Fig. 8 shows how the MoT implements the HSRA. What is not immediately demonstrated in such diagrams is how these extra wires will be laid out in the MoT. Most importantly, *when the HSRA-augmenting connections in the MoT are placed, what is the maximum channel width and maximum number of switches per node?* It turns out that we can distribute these augmented connections across the span of a hierarchical MoT segment so that there are a constant number of switches per

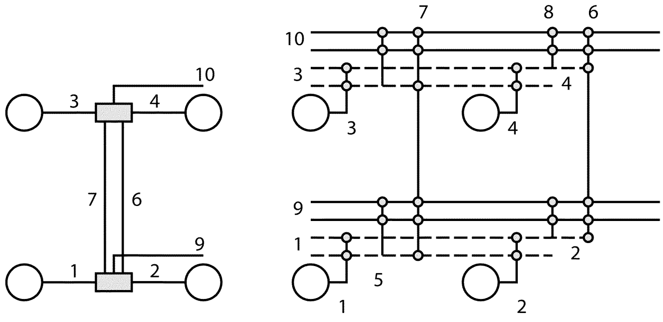


Fig. 11. Channel layout view and signal mapping for augmented MoT.

endpoint and that the number of wires per channel grows at exactly the same rate as the MoT channel wires (see middle right of Fig. 12). Further, by using the existing wire segments in the orthogonal routing channels, which we have been ignoring so far in our equivalence argument, no new wires are needed—we simply need to apply an additional set of switches which allow us to use the existing wires in this manner (see bottom row of Fig. 12).

To show switch and wire spreading, it is useful to view a more detailed view of the MoT/HSRA layout which shows channel runs and switches. Fig. 11 shows such a view alongside the logical view for a $p = 0.5$ MoT augmented with these HSRA links. Note here that we actually use a pair of MoT wires to implement each single wire into the switchboxes in order to get the full connectivity of the HSRA switching. In particular, this allows a full interchange (e.g.,$3 \rightarrow 2$, $4 \rightarrow 1$) which would not be possible if we only used a single wire. The need for two wires arises because the MoT wire is not segmented and switched at the switchbox, as in the HSRA, but rather is a continuous run and we effectively spread out the switches in the HSRA switchbox along the length of the pair of wires.

Fig. 12 shows an augmented MoT network for $p = 0.75$ alongside a $p = 0.75$ HSRA. The $p = 0.75$ case makes clearer the fact that we cannot run all the wires directly in the place where they are shown in the equivalence diagram (see top right of Fig. 12) without filling the channels unevenly. In fact, there will be $O(N^p)$ such connections at the top of the tree, whereas the MoT layout has already spread out the existing $O(N^p)$ **total** wires in its bisection among the $\sqrt{N}$ channels such that there are only $O(N^{(p-0.5)})$ wires per channel and demonstrated that these can be laid out in constant width per channel given $\Theta(N^{(p-0.5)})$ wire layers [13] [reviewed below cumulating in (22)].

Fig. 12 (middle row) shows a channel layout view for the augmented MoT. We use a $c = 2$ MoT to accommodate a $c = 1$ HSRA as suggested above. We note here that the eight wires which had crossed the bisection are now spread out so that there are two wires in each of the four channels. This is accomplished in exactly the same way we guaranteed there were only $O(1)$ switches at each endpoint [13]. Here it is important that we maximally spread out uplinks at a given level so that we do not get multiple links to the same level at the same endpoint; the geometric reduction in uplinks (wire) per endpoint as we ascend the tree guarantees that this is easy to accomplish. In fact, once we spread out the uplinks properly, we use the placement of the parent–child uplink switches as a guide for where to place these crosslink connections. Every place we have an uplink switch, we place a companion augmenting link to the associated wire in the companion stage (2^l channels above or below as previously identified). In this way, we roughly double the number of switches at each endpoint. Unlike switches, the wires do overlap. That means the number of wires per channel will grow as longer wires overlap shorter ones. The trick is to notice that the wire growth exactly matches the standard wire channel growth so that we can use existing wires for these runs.

Constructively, we note that we have a total of $g_0 \times g_1 \times \cdots g_l$ uplinks at the root of a height l row or column tree. Rearranging (13)

$$N_{\text{up}}(l) = g_0 \times g_1 \times \cdots g_l = 2^{2lp-l} = 2^{2l(p-0.5)}. \tag{16}$$

To convert this to a per channel uplink count, we have divided out the 2^l factor which results from combining across the 2^l channels contributing uplinks. Equation (16) is exactly the per channel row or column width at level l necessary to satisfy our Rent relation

$$W_{\text{mot}}(l) = \frac{N^p}{2^l} = \frac{2^{2lp}}{2^l} = 2^{2l(p-0.5)}. \tag{17}$$

These uplinks are distributed across the segment span of length 2^l, so each node gets

$$N_{\text{up\_per\_node}}(l) = \frac{N_{up}}{2^l} = \frac{2^{2l(p-0.5)}}{2^l} = 2^{2l(p-1)}. \tag{18}$$

The augmenting wires span length 2^l. Wire channel width contribution per level then is

$$W_{\text{mot}_{\text{augment}}}(l) = 2^l N_{\text{up\_per\_node}}(l) = 2^{2l(p-0.5)}. \tag{19}$$

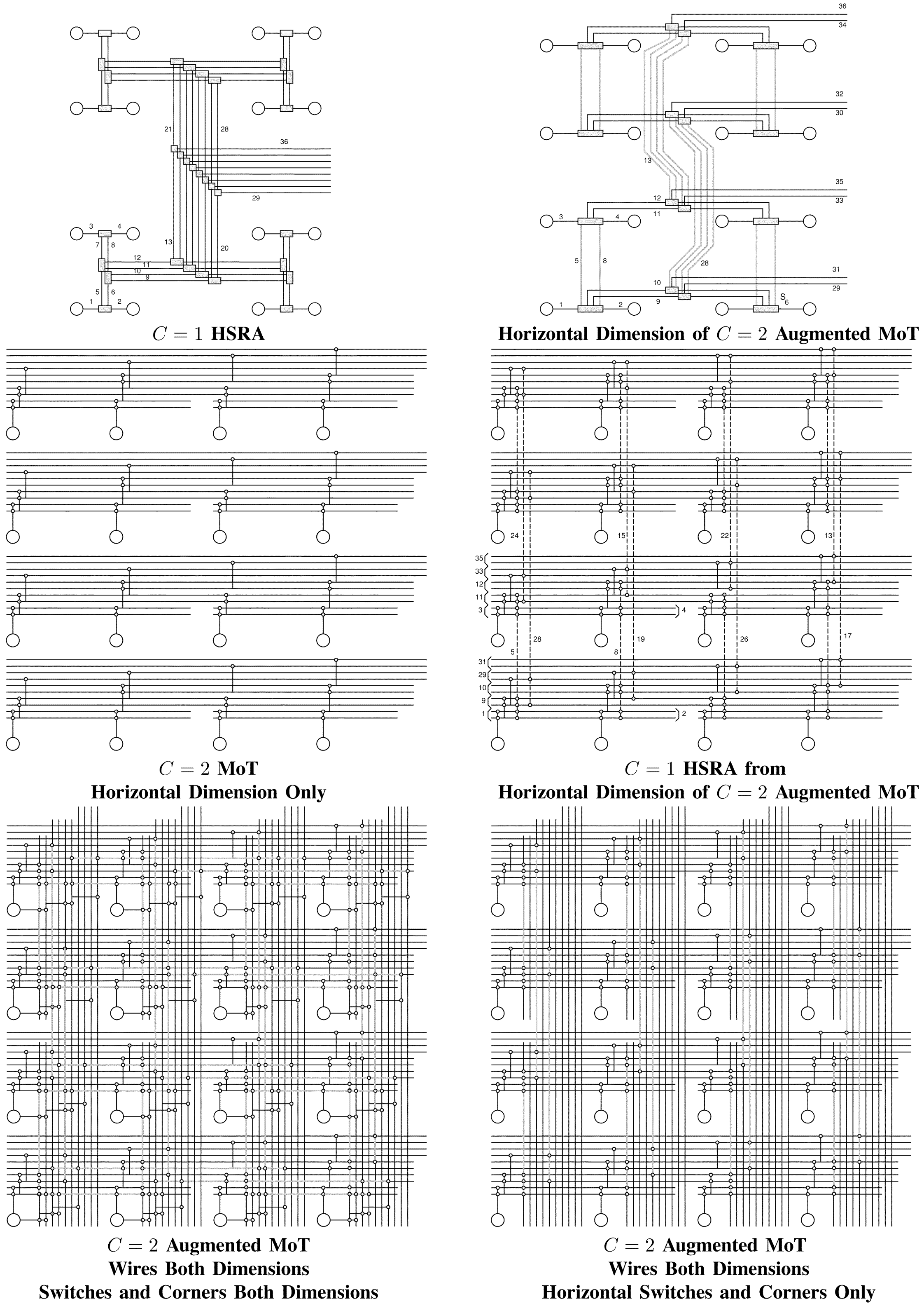
As suggested, this shows the same wire requirements as the MoT needed for this level (17).

We further note that the total width of either channel is

$$W_{\text{mot}} = \sum_{l=0}^{l=\log(\sqrt{N})} \left(W_{\text{mot}}(l)\right) \tag{20}$$

$$W_{\text{mot}} = \sum_{l=0}^{l=\log(\sqrt{N})} \left(2^{2l(p-0.5)}\right) \tag{21}$$

$$\begin{aligned} W_{\text{mot}} &= 2^0 + 2^{2(p-0.5)} + 2^{4(p-0.5)} + \cdots \\ &\quad + 2^{2\log(\sqrt{N})(p-0.5)} \\ &= 2^0 + 2^{2(p-0.5)} + 2^{4(p-0.5)} + \cdots + 2^{\log(N)(p-0.5)} \\ &= 2^0 + 2^{2(p-0.5)} + 2^{4(p-0.5)} + \cdots + N^{(p-0.5)} \\ &= N^{(p-0.5)} + \left(\frac{N}{2}\right)^{(p-0.5)} + \left(\frac{N}{4}\right)^{(p-0.5)} + \cdots + 1 \\ &= N^{(p-0.5)}\left(1 + \frac{1}{2^{(p-0.5)}} + \frac{1}{2^{2(p-0.5)}} + \cdots \frac{1}{N^{(p-0.5)}}\right) \\ &< N^{(p-0.5)}\left(\frac{1}{1-\left(\frac{1}{2}\right)^{(p-0.5)}}\right). \end{aligned} \tag{22}$$



$C = 1$ **HSRA**

Horizontal Dimension of $C = 2$ **Augmented MoT**

$C = 2$ **MoT**
Horizontal Dimension Only

$C = 1$ **HSRA from**
Horizontal Dimension of $C = 2$ **Augmented MoT**

$C = 2$ **Augmented MoT**
Wires Both Dimensions
Switches and Corners Both Dimensions

$C = 2$ **Augmented MoT**
Wires Both Dimensions
Horizontal Switches and Corners Only

Fig. 12. Mapping between $c = 1, p = 0.75$ HSRA and $c = 2, p = 0.75$ augmented MoT.

For $p > 0.5$, $(1/2)^{(p-0.5)} < 1$, so the sum converges to a p-dependent constant times $N^{(p-0.5)}$, which is within a constant factor of the mesh channel width lower bound (11).

Since we have noted that the number of wires added for a stage of augmenting links is exactly the same as the number of wires in the parent stage to which they are connecting; and since we are using a $c = 2$ MoT, we can use the wires in the corresponding stage of the orthogonal tree to perform this connection simply by adding the switches necessary to allow them to serve as these augmenting links. The bottom row of Fig. 12 shows the

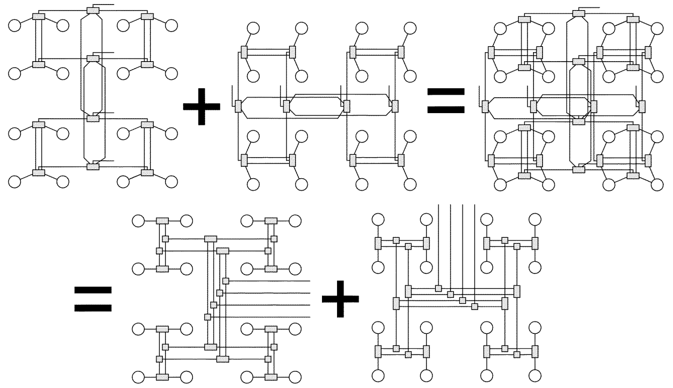


Fig. 13. $c = 4, p = 0.5$ augmented MoT in both dimensions corresponds to composing two $c = 1, p = 0.5$ HSRAs rotated 90° relative to each other.

additional switches and highlights where the augmented paths run over the orthogonal wire runs.

We have shown how to make one-dimension of a $(c_{\text{mot}} = 2c_{\text{hsra}}, p)$ MoT contain a (c_{hsra}, p) HSRA. We can populate the augmenting link switches in both dimensions as shown in Fig. 12 (bottom left). This makes a $c = 4$ MoT contain $c = 1$ HSRA composed with its transpose (see Fig. 13).

C. Implications

1) Design Unification: The augmentation that turns the MoT into the BFT/HSRA allows us to exactly understand the difference between these two networks. We can now see the BFT as a particular corner turn scheme applied to the MoT and unify these two networks into a single, parameterized design.

How we might add links between horizontal and vertical tracks in the MoT (corner turns) was an outstanding question before we discovered this mapping. If we simply allowed every wire to connect to the wires at the same tree level that cross it, we would need an asymptotically growing number of switches per node and would lose the linear switch bound of the MoT. The BFT/HSRA wiring gives us insight into how to formulate a limited corner turn scheme for the MoT. This corner turn scheme does not asymptotically increase the switches or wires in the MoT, but does provide interesting switching characteristics. Since the BFT/HSRA only has to route up and down a single tree, whereas the MoT without augmentation generally has to route up and down two trees, the augmented MoT has half the switches in the worst-case paths between a source and a destination.

A common complaint leveled at the arity–2 HSRA topology is the asymmetry between the horizontal and vertical connections; as shown on the right of Fig. 4, the horizontal nearest-neighbor is one switch away while the vertical nearest-neighbor is two. The equivalence in Fig. 13 makes it clear that the arity-2 HSRA directionality bias can be removed by overlaying the network with its transpose. Since all cases where we use MoT and HSRA networks for FPGAs have $c > 1$, we will always have multiple trees and be able to alter the orientation of the trees relative to each other. This equivalence also makes it clearer that the MoT staggering can be applied to the BFT/HSRA.

This mapping further shows us how we can apply any results on fast HSRA mapping (e.g., [30]) to the MoT.

TABLE I
SUMMARY OF RECURRING SYMBOLS USED IN ARTICLE

Symbol	Description	Intro
A_{wire}	Lower bound on chip area due to wiring requirements	Eq. 7
BW	Bisection Width	Sec. II
BW_{mesh}	Wires in Bisection for Mesh Topology	Eq. 8
c	Constant in Rent's Rule	Eq. 1
c_{hsra}	Number of Base Channels in HSRA (Fig. 4)	Sec. III-B
c_{mot}	Number of Base Channels in MoT (Fig. 5)	Sec. III-B
g_i	Growth rate at level i in MoT/ToM Tree	Eq. 12
g_{hsra}	Growth Sequence for HSRA/ToM Trees	Eq. 15
g_{mot}	Growth Sequence for MoT Trees	Eq. 16
IO	Total input/output from a region	Eq. 1
l	Level or Number of levels in MoT or ToM Tree	Eq. 12
L_h	Side length of chip in horizontal direction	Eq. 6
L_v	Side length of chip in vertical direction	Eq. 4
N	Number of Primitive elements in a region	Eq. 1
N_h	Number of horizontal wiring layers	Eq. 6
N_{up}	Number of uplinks at a specified level	Eq. 16
N_v	Number of vertical wiring layers	Eq. 4
p	Growth exponent in Rent's Rule	Eq. 1
W	Mesh Channel Width (Fig. 6)	Eq. 8
W_{mot}	Width of MoT Channel	Eq. 22
W_{pitch}	Wire Pitch	Eq. 4

TABLE II
COMMON PARAMETERIZATION AND ANALOGS

MoT/BFT/HSRA	Manhattan Mesh	Note
Flatness	complete	Standard mesh has flat connectivity up all row and column trees
Arity	L_{seg}	generally a distribution of segment lengths
Base Channels c	W	W per segment length (L_{seg}) may be a function of c,p
Growth p		
Shortcuts	Switch Box Population	Mesh designs typically assume all present
Corner Turns		
Staggering		Same idea
Domains		Similar issues

2) Asymptotics and Layout: We see from (17), (19), (22), and (11), that the $p > 0.5$ MoT, the $p > 0.5$ augmented MoT (or the HSRA), and the mesh have the same asymptotic channel width. None of the networks has more than a constant factor fewer wiring tracks than any of the others.

The equivalence transformation here tells us we can apply what we know about MoT layouts to HSRA layouts. Significantly, the construction above showed that the HSRA can be laid out in asymptotically the same channel width as the MoT. We previously showed that a $p = 0.5$ BFT/HSRA could be laid out in linear area given $\Theta(\log(N))$ wire layers [28] ; but at that point in time the general question of laying out a BFT/HSRA $(1.0 > p > 0.5)$ in linear area using multilayer metallization remained open. The equivalence above allows us to exploit our prior construction that showed how to layout the MoT for any $p > 0.5$ in linear space using $\Theta(N^{(p-0.5)})$ wire layers [13] in order to also layout any HSRA in linear two-dimensional area using $\Theta(N^{(p-0.5)})$ wire layers. This now gives us two networks that have the $\Theta(N)$ layout area property.

V. MoT TO MESH PARAMETERIZATION

In a companion article [13], we compared the MoT to a conventional, Manhattan mesh. The most fundamental difference

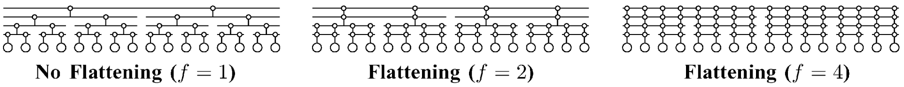


Fig. 14. Flatness parameter shown on single row (column) channel in $p = 0.5$ MoT.

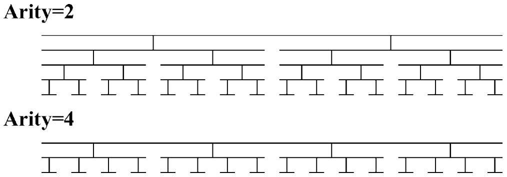


Fig. 15. Arity parameter on single row (column) channel in $p = 0.5$ MoT.

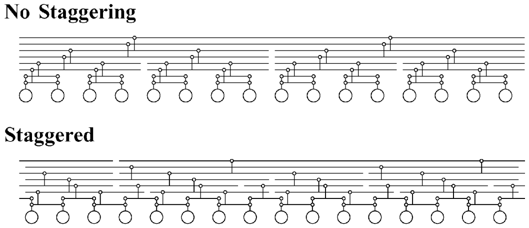


Fig. 16. Staggering single row (column) channel in $p = 0.5$ MoT.

between the Manhattan mesh and the MoT is the flat endpoint connectivity on the mesh. That is, the mesh C-Box connects the compute element's inputs or outputs to all of (a constant fraction of) the wires in the channel, whereas the MoT only connects to the base level tree channels and uses the tree connections to climb up the tree to reach longer segments. This has the immediate impact that the MoT needs only a linear number of switches, while the Mesh needs $O(W)$ switches per endpoint. Since we have already established that W grows with N for $p > 0.5$ [(11)], we see that the mesh requires asymptotically more switches than the MoT.

We can parameterize this difference and the other traditional differences between the MoT and the mesh in order to define a continuum space between the extremes. Table II summarizes the variables. As noted in Fig. 1, we can view a mesh as a special, degenerate case of the MoT where we tune several of the parameters to their extreme values.

A. *Parameterization*

1) Flatness: We parameterize flatness by the number of parent tree levels to which we connect each child node. In the MoT, we connect a child at level l to a parent at level $l + 1$. In the Mesh, we connect the leaf child at level 0 to all levels above it. In general, we could provide direct connections among a group of f levels; that is, we connect a child at level $n \cdot f$ to levels $n \cdot f + 1, n \cdot f + 2, \ldots n \cdot f + f = (n+1) \cdot f$ (see Fig. 14).

2) Segment Distribution: The Rent relation (1) can be applied strictly to define a set of segment distributions. We see from the MoT designs, that we get c length 1 segment, $c \times g_{\mathrm{mot}_0}$, length 2 segments, $c \times g_{\mathrm{mot}_0} \times g_{\mathrm{mot}_1}$, length 4 segments, and so on. Recall from (13), that we pick the growth rates to correspond to our target p value. This same idea could be applied to the selection of mesh segment lengths and segment length distributions. As we noted earlier [13], if these lengths are chosen geometrically in this manner, and if corner turns are only allowed at segment ends, the mesh only needs a total number of S-Box switches which is linear in the number of nodes supported by the design.

Conventional mesh designs have often chosen to truncate their hierarchy—stopping after a given segment length or jumping from one segment length to full row/column length lines rather than including all of the geometric wire lengths.

3) Arity: For simplicity, we have, so far, described and shown binary trees for the MoT and HSRA. We showed an arity-4 BFT in Fig. 4. In general, we can build trees with any number of children levels to a parent level. For example, Fig. 15 shows a MoT row with an arity of 4 as contrasted with an arity-2 MoT row. The arity tunes the rate of segment growth. So an arity-4 MoT has segments of length $1, 4, 16 \ldots$ rather than $1, 2, 4, 8, 16 \ldots$ In this way the combination of arity and p defines segment distribution.

4) Staggering: When we have multiple segments of non-unit length longer than one, it is useful, both for switch placement and for routing, to spread out the switch placements (e.g., [31]–[33]). In the MoT, this is true as well [13]. For the MoT, we stagger the alignment of the trees relative to each other (see Fig. 16). In both cases, staggering minimizes the cases where a route must use a significantly longer (higher) link than it should take to span the distance between the source and sink.

5) Shortcuts: In the strict tree structure of the MoT and BFT/HSRA, there are cases where two nodes are physically close in the layout but logically distant in the tree. This effect is mitigated by staggering. It can be eliminated entirely by adding shortcut connections which allow segments at the same level and in the same channel to be connected to their immediately adjacent neighbors. These shortcuts, which only requires a constant factor more switches than the base MoT, now guarantee that the physical distance one must travel in the MoT or BFT/HSRA is never more than a constant factor larger than the Manhattan distance; this was the key innovation of the Fat Pyramid [34]. Further, the number of switches on the path will be logarithmic in distance, making it asymptotically fewer than any bounded-segment length mesh scheme. These shortcuts perform exactly the same switching as the end-to-end segment switching (E $\leftrightarrow$ W, N $\leftrightarrow$ S) which appears in the switchpoints of standard, Manhattan, switchbox designs (see Fig. 17). That is, in the standard diamond switchbox, the switch which connects a segment to a single segment of the same length in the same channel on the other side of the switchbox, is essentially the same as the shortcut switches which we may or may not include in MoT or ToM designs.

6) Corner Turns: Corner turn parameterization defines where and how routes may turn between orthogonal channels (from horizontal to vertical routing or *vice-versa*). In a standard mesh switchbox, a segment has a corner turning switch to a single orthogonal segment when it crosses that segment or to

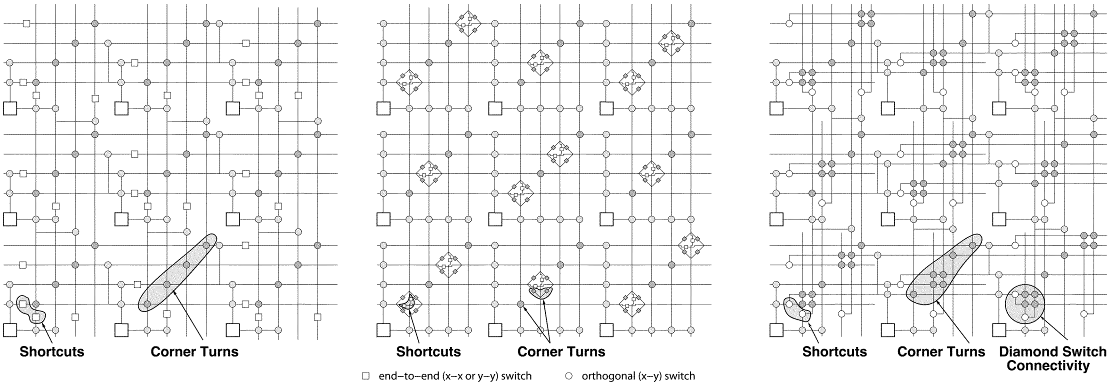

Shortcuts
Corner Turns
Shortcuts
Corner Turns
□ end–to–end (x–x or y–y) switch
○ orthogonal (x–y) switch
Shortcuts
Corner Turns
Diamond Switch Connectivity

Fig. 17. Shortcut, corner turn, and switchbox population.

one segment in each direction when it arrives at a switchbox coincident with a segment break in its corresponding orthogonal segment. In a standard diamond switch configuration, these are the NW, NE, SW, and SE switches (see Fig. 17). These corner turns make up the remaining 1–2 switches which are normally attached to the end of each segment (compare $F_s = 3$ in the Toronto Model [3], [4]).

Since W is growing with N for $p > 0.5$ (11), as long as we have at least one corner turn per segment per switchbox, the number of switches per switchbox is growing, so the total number of switches in all the switchboxes are growing faster than linearly. It is asymptotically desirable to avoid this level of corner turn population. The fact that we can lay out MoT and BFT/HSRA designs using asymptotically similar wiring requirements but without such extreme corner turn population certainly suggests there is a viable alternative and it will eventually be beneficial to exploit it. In the MoT, we consider whether corner turns should be limited to the leaves or whether we need some, limited scheme for higher level corner turns (Section IV-C1 and [13]). In general, all the Mesh and MoT corner turn variations make up a rich parameterized design space.

The typical MoT layout, shown on the left of Fig. 17, allows only a single corner turn at each switchbox. On the right of Fig. 17, we show an alternate layout that overlaps adjacent segments in the same channel so that we can use simple switches between orthogonal lines to support corner turns and allow this inclusion of the pair of corner turns (e.g., both NE and NW from the north input to a switchbox) typical in mesh switch populations.

7) Switchbox Population and Domains: It should be clear that the general issues, which the mesh considers in terms of switchbox population can be decomposed into shortcut and corner turn issues above and shown in Fig. 17. In the preceding, we described the most popular mesh and MoT designs where there are a constant number of wires connecting the end (or internal points in the mesh case) of segments. More generally, we might ask about fuller switchbox population (e.g.,[35], [25]) and there are similar questions in both designs. The traditional mesh design has disjoint domains which are only connected at the leaves; similarly, the primary MoTs we have described only allow turns between separate row and column trees at the leaves and typically can only change among corresponding row and column trees at a corner turn, giving them this domain structure as well.

B. Implications

1) Unified Design Space: Unifying the design space gives us greater insight into how we can tune designs. Reconciling the Mesh with the MoT introduces new design parameters to explore for tuning the Mesh and the MoT. It also sheds some light on the assumptions we tend to make in these designs—assumptions which may merit reexamination, e.g., would the MoT be better off adopting the rich segment endpoint switching of a typical mesh, or would the mesh be better off omitting some of these switches? Is flat-endpoint connectivity a good assumption to keep as we scale up to million compute-node designs? Note that the mesh's flat-endpoint connectivity and full corner-turn population are each, individually, sufficient to force the number of switches in the mesh to grow superlinearly in the number of nodes. The MoT shows us a plausible alternative to avoid the asymptotic growth arising from flat-endpoint connectivity, and the ToM to MoT mapping show us a plausible alternative to full corner-turn population.

2) Shortcuts: Our experience with MoT designs to date suggests that shortcut connections may offer marginal additional value compared to staggering [13]. Shortcuts do reduce the total channel width required to route the design when we do not have staggered segments, but only at a net increase in the total number of switches. Once we add staggering, even the wire reduction benefit is marginal. It may be interesting to consider shortcut depopulation in the Manhattan mesh.

3) Pre- and Post-Layout Rent Characteristics: As shown in the previous section, we can layout a BFT/HSRA with asymptotically the same channel width as a Mesh. We can use the same layout strategy for the ToM, giving us a generalization of Leiserson and Greenberg's fold-and-squash layout [36]. We know the ToM can accommodate layouts simply by recursive bisection. As long as the bisection cuts do not exceed the tree bandwidths, the recursive bisection design will be routable on the ToM, which we now know can be laid out with asymptotically the same channel width as the mesh.

Put together, these observations imply that the *a posteriori* global route Rent exponent for a Manhattan layout should be the same as the *a prior* Rent exponent. That is, while there may be difference in the layout-based partitions (e.g.,[15], [16]), these should, at most, be placement shuffles to reduce the constant factors associated with tree overlap among tree levels and will not change the asymptotic growth rate. The MoT and ToM layout described above tell us how to take any Rent characterized (c,p) design and lay it out with $O\left(cN^{(p-0.5)}\right)$ Manhattan channel width. This gives us an upper bound on the global channel width required to route a (c,p) design on a mesh; this upper bound is within a constant factor of the lower bound we previously derived on mesh channel width ((11)). Note that the $c_{\text{mot}} = 2c_{\text{hsra}}$ construction above already more than accounts for the downlink conflicts that forced us to use a $(1.5c,p)$ ToM to accommodate a (c,p) design, so the channel width is no higher than $2c$ times the per channel width of the MoT derived in (22)

$$\begin{aligned} W &\leq 2cN^{(p-0.5)}\left(\frac{1}{1-\left(\frac{1}{2}\right)^{(p-0.5)}}\right) \\ &\leq \left(\frac{2^{(p+0.5)}}{2^{(p-0.5)}-1}\right) cN^{(p-0.5)}. \end{aligned} \tag{23}$$

This suggests that there is no fundamental reason for the post-placement Rent exponent for a design to be larger than the pre-placement Rent. However, while asymptotically tight, the bounds are loose in absolute terms; for example, the ratio between the lower bound in (11) and this upper bound ((23)) is around 30 for $p = 2/3$. Consequently, this leaves room for large constant factor differences between pre-placement and post-placement IO ratios, and it may take very large designs for the asymptotic effects to dominate. Our construction is unlikely to be the tightest possible, so we leave open the question of how much it is possible to tighten the constant factors in this mapping.

The guarantee above is made in terms of a global route and the full population ToM, rather than the mesh detailed route and the HSRA or BFT, because the mapping ratio for linear population designs remains an important, open question [35], [25]. The result is directly applicable more to custom routing than FPGA routing for this reason. However, if we can establish a constant mapping ratio for some variant of the HSRA/BFT, then these observations would allow us to apply this result to these detailed networks as well.

VI. Summary

To build efficient switching networks for typical circuits, we must use networks which allow us to exploit the locality structure which exists in these circuit graphs. Manhattan meshes, MoT, and ToM style networks are all examples of limited-bisection switching networks which support this locality exploitation. While these networks are different in formulation, we see that they have the same asymptotic wiring requirements—all requiring $O(N^{(p-0.5)})$ wires per channel in 2-D layouts when $p > 0.5$. Using embeddings, we have demonstrated equivalence mappings between the networks (MoT embedded in HSRA, HSRA embedded in augmented MoT with corner turns, and MoT embedded in 2-D mesh); all of these mappings require at most a constant scale factor in wires. The MoT to ToM and ToM to MoT embeddings are made with only a constant scale factor in switches. From these mappings we now see how to layout linear-population ToM designs of any p (e.g., BFT and HSRA) in constant area using multilayer metallization and how to produce constructive global mesh routes which are known to be within a constant factor of optimal. We can view these networks within a larger, unified design space which helps us understand the tradeoffs which each network makes and aides our search for network parameters which meet design goals.

References

[1] W. S. Carter, K. Duong, R. H. Freeman, H.-C Hsieh, J. Y. Ja, J. E. Mahoney, L. T. Ngo, and S. L. Sze, "A user programmable reconfigurable logic array," in *Proc. IEEE 1986 Custom Integrated Circuits Conf.*, May 1986, pp. 233–235.

[2] S. Trimberger, "A reprogrammable gate array and applications," *Proc. IEEE*, vol. 81, pp. 1030–1041, July 1993.

[3] J. Rose and S. Brown, "Flexibility of interconnection structures for field-programmable gate arrays," *IEEE J. Solid-State Circuits*, vol. 26, pp. 277–282, Mar. 1991.

[4] S. D. Brown, R. J. Francis, J. Rose, and Z. G. Vranesic, *Field-Programmable Gate Arrays*. Norwell, MA: Kluwer, 1992.

[5] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*. Norwell, MA: Kluwer, 1999.

[6] *FLEX 8000 Handbook*, Altera Corporation, San Jose, CA, 1994.

[7] A. A. Agarwal and D. Lewis, "Routing architectures for hierarchical field programmable gate arrays," in *Proc. 1994 IEEE Int. Conf. Computer Design*, Oct. 1994, pp. 475–478.

[8] V. C. Chan and D. Lewis, "Area-speed tradeoffs for hierarchical field-programmable gate arrays," in *Proc. Int. Symp. Field-Programmable Gate-Arrays*, Feb. 1996, pp. 51–57.

[9] Y.-T. Lai and P.-T. Wang, "Hierarchical interconnection structures for field programmable gate arrays," *IEEE Trans. VLSI Syst.*, vol. 5, pp. 186–196, June 1997.

[10] A. DeHon, "Balancing interconnect and computation in a reconfigurable computing array (or, why you don't really want 100% LUT utilization)," in *Proc. Int. Symp. Field-Programmable Gate-Arrays*, Feb. 1999, pp. 69–78.

[11] W. Tsu, K. Macy, A. Joshi, R. Huang, N. Walker, T. Tung, O. Rowhani, V. George, J. Wawrzynek, and A. DeHon, "HSRA: High-speed, hierarchical synchronous reconfigurable array," in *Proc. Int. Symp. Field-Programmable Gate-Arrays*, Feb. 1999, pp. 125–134.

[12] R. Sidhu, S. Wadhwa, A. Mei, and V. K. Prasanna, "A self-reconfigurable gate array architecture," in *Proc. Int. Conf. Field-Programmable Logic Applications*, Villach, Austria, Aug. 2000, pp. 106–120.

[13] R. Rubin and A. DeHon, "Design of FPGA interconnect for multilevel metallization," *IEEE Trans. VLSI Syst.*, vol. 12, pp. 1038–1050, Oct. 2004.

[14] A. E. Caldwell, A. B. Kahng, and I. L. Markov, "Can recursive bisection produce routable placements?," in *Proc. IEEE Design Automation Conf.*, 2000, pp. 477–482.

[15] P. Verplaetse, J. Dambre, D. Stroobandt, and J. V. Campenhout, "On partitioning vs. placement rent properties," in *Proc. 2001 Int. Workshop on System-Level Interconnect Prediction (SLIP)*, Mar. 2001, pp. 33–40.

[16] X. Yang, E. Bozorgzadeh, and M. Sarrafzadeh, "Wirelength estimation based on rent exponents of partitioning and placement," in *Proc. 2001 Int. Workshop System-Level Interconnect Prediction*, Mar. 2001, pp. 25–31.

[17] A. DeHon and J. Wawrzynek, "Reconfigurable computing: What, why, and design automation requirements?," in *Proc. 1999 Design Automation Conf.*, June 1999, pp. 610–615.

[18] S. W. Gehring and S. H.-M. Ludwig, "Fast integrated tools for circuit design with FPGAs," in *Proc. Int. Symp. Field-Programmable Gate-Arrays*, Feb. 1998, pp. 133–139.

[19] R. Tessier and H. Giza, "Balancing logic utilization and area efficiency in FPGAs," in *Proc. Int. Conf. Field-Programmable Logic and Applications*, R. W. Hartenstein and H. Grübacher, Eds., Aug. 2000, pp. 535–544.

[20] C. Thompson, "Area-time complexity for VLSI," in *Proc. 11th Annual ACM Symp. Theory Computing*, May 1979, pp. 81–88.

[21] B. S. Landman and R. L. Russo, "On pin versus block relationship for partitions of logic circuits," *IEEE Trans. Computers*, vol. 20, pp. 1469–1479, 1971.

[22] D. Stroobandt, "On an efficient method for estimating the interconnection complexity of designs and the existence of region iii in Rent's rule," in *Proc. Great Lakes Symp. VLSI*, Mar. 1999, pp. 330–331.

[23] F. T. Leighton, "New lower bound techniques for VLSI," in *Proc. Annual Symp. Foundations Computer Science*, Nashville, TN, 1981, pp. 1–12.

[24] S. Bhatt and F. T. Leighton, "A framework for solving VLSI graph layout problems," *J. Comput. Syst. Sci.*, vol. 28, pp. 300–343, 1984.

[25] A. DeHon, "Rent's rule based switching requirements," in *Proc. System-Level Interconnect Prediction Workshop* , Mar. 2001, pp. 197–204.

[26] C. E. Leiserson, "Fat-trees: Universal networks for hardware efficient supercomputing," *IEEE Trans. Computers*, vol. C-34, pp. 892–901, Oct. 1985.

[27] R. I. Greenberg and C. E. Leiserson, "Randomized routing on fat-trees," in *Randomness in Computation*, MIT/LCS/TM-307 ed. Greenwich, CT: JAI Press, 1988, vol. 5, Advances in Computing Research.

[28] A. DeHon, "Compact, multilayer layout for butterfly fat-tree," in *Proc. 12th ACM Symp. Parallel Algorithms Architectures*, July 2000, pp. 206–215.

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

[30] R. Huang, J. Wawrzynek, and A. DeHon, "Stochastic, spatial routing for hypergraphs, trees, and meshes," in *Proc. Int. Symp. Field-Programmable Gate-Arrays*, Feb. 2003, pp. 78–87.

[31] M. Pedram, B. S. Nobandegani, and B. T. Preas, "Design and analysis of segmented routing channels for row-based FPGAs," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 1470–1479, Dec. 1994.

[32] S. Brown, M. Khellah, and Z. Vranesic, "Minimizing FPGA interconnect delays," *IEEE Des. Test Comput.*, vol. 13, pp. 16–23, Jan. 1996.

[33] K. Compton and S. Hauck, "Track placement: Orchestrating routing structures to maximize routability," in *Proc. Int. Conf. Field-Programmable Logic Applications*, Lisbon, Portugal, Sept. 2003, pp. 121–130.

[34] R. Greenberg, "The fat-pyramid and universal parallel computation independent of wire delay," *IEEE Trans. Computers*, vol. 43, pp. 1358–1365, Dec. 1994.

[35] Y.-L. Wu, S. Tsukiyama, and M. Marek-Sadowska, "Graph based analysis of 2-d FPGA routing," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 33–44, Jan. 1996.

[36] R. I. Greenberg and C. E. Leiserson, "A compact layout for the three-dimensional tree of meshes," *Appl. Math Lett.*, vol. 1, no. 2, pp. 171–176, 1988.

André DeHon (S'92–M'96) received the S.B., S.M., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1990, 1993, and 1996, respectively.

From 1996 to 1999, he was Co-Head of the BRASS Group, Computer Science Department, University of California, Berkeley. Since 1999, he has been an Assistant Professor of Computer Science at the California Institute of Technology. He is broadly interested in how we physically implement computations from substrates, including VLSI and molecular electronics, up through architecture, CAD, and programming models. He places special emphasis on spatial programmable architectures (e.g., field-programmable gate-arrays) and interconnect design and optimization.