Pitfalls of ORION-Based Simulation

Mitchell Hayenga
University of Wisconsin-Madison
hayenga@ece.wisc.edu

Daniel R. Johnson
University of Illinois at Urbana-Champaign
djohns53@uiuc.edu

Mikko Lipasti
University of Wisconsin-Madison
mikko@engr.wisc.edu

Abstract—**ORION is the standard tool for modeling interconnect area and power within the architecture community. Despite its prevalence, a quick survey of academic work shows multiple implementations with energy and area estimates which greatly differ from ORION estimates.**

In this paper we perform an in-depth analysis of ORION's area and power modeling in order to locate potential pitfalls and sources of discrepancy. This analysis is performed on a component-wise basis, focusing primarily on the router datapath components of input buffers and crossbar switches. Overall, we observe that due to its generality and chosen circuit implementations, ORION overstates the following parameters: buffer area (5x), buffer power (1.4x), crossbar area (30x). By systematically overestimating the achievable area and power of on-chip routers, ORION overstates system-level interconnect power and the relative gain obtainable from improvements in router microarchitecture.

I. Introduction

With the continuation of Moore's Law [1] and end of Dennard scaling [2], efficiency and power density have become paramount design parameters for integrated circuits. Assisting in the characterization of these parameters, ORION [3] and its follow on ORION 2.0 [4] provide power and area estimation tools for on-chip routers and communication channels. As a valuable tool, ORION has seen wide acceptance and the majority of interconnection network papers which present area or power numbers utilize it. Between the years of 2010 and 2011 at three top conferences (ISCA, MICRO, and HPCA), nine interconnect papers [5]–[13] utilized the ORION framework whereas only five interconnect papers [14]–[18] did not.

However, despite ORION's broad acceptance it is widely known that its area and power estimates may differ greatly from actual implementations. In [19] multiple ORION authors compare against a full implementation of [20] and observe that ORION frequently overestimates router power by $\sim$2x. Although [19] notes that ORION's deficiency is due to its chosen circuit implementations, to the best of our knowledge no in-depth analysis has been performed characterizing ORION's competitiveness against alternative implementations on a component-wise basis.

In this paper we evaluate ORION's power and area estimates against other freely available tools and published designs. From this analysis we observe that in comparison to well optimized architectural components, ORION greatly overstates power and area for multiple router components. ORION's deficiencies stem primarily from three sources:

- Generality and abstraction
- Overly conservative design parameters
- Inconsistent modeling techniques

Resulting from these issues, ORION systematically overestimates buffer area (5x), buffer power (1.4x), and crossbar area (30x) over tuned alternatives.

In addition to evaluating ORION's optimality, we detail how its results affect microarchitectural tradeoffs and might impact the conclusions of academic researchers. By overestimating the relative costs of buffers and crossbars, publications utilizing ORION to justify alternative buffering mechanisms [7]–[10], crossbar designs [8], or network topologies [11]–[13] potentially overstate their gains.

The remainder of this paper is organized as follows. In Section II we examine ORION's input buffering methodology and compare against the freely available FabMem [21] memory compiler. In Section III we detail ORION's multiplexer-based crossbar and present an alternative design for comparison. Additionally we note several issues with ORION's area modeling. Finally, in Section IV we summarize the impacts of our research and conclude the paper.

All characterization is conducted at the 45nm technology node using ORION 2.0.

II. Input Buffers

As buffers are one of the primary datapath components in on-chip routers, countless publications dedicated to alternative buffer organizations and management techniques have been published. Key to these studies is the accurate accounting of buffer cost. In this section we investigate the buffer area and power models utilized by ORION.

A. Methodology

For a evaluation basis we utilize the FabMem [21] memory compiler which is part of the larger FabScalar project [22]. The FabMem memory compiler is optimized for the generation of small, multi-ported SRAM and CAM arrays at the 45nm technology node. Power and timing analyses of SRAM structures are performed through HSPICE simulation of extracted netlists.

To limit variability, we restrict our evaluation to 2-port (1 Read, 1 Write), single-bank SRAM arrays of varying width and depth.

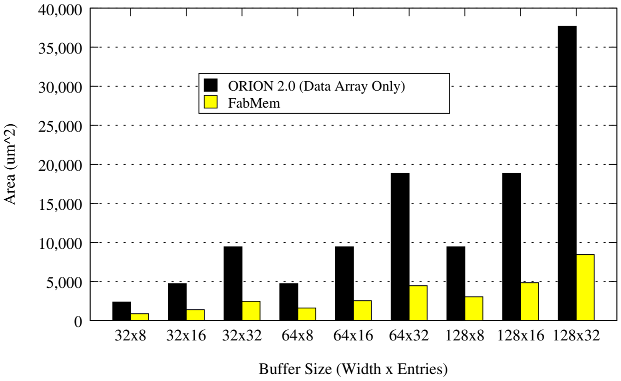


Fig. 1. Input Buffer Area. ORION reports array area only. FabMem reports decoder, array, and sense amplifier area.

TABLE I
1 Read, 1 Write SRAM Bit Cell Areas

Orion	FabMem	Difference
9.1988μm^2	1.78μm^2	5.16x

B. Area Analysis

Figure 1 shows reported SRAM buffer areas from ORION and FabMem for a range of configurations. For these SRAM buffer sizes, ORION's area estimates are 2.7x–4.5x greater than the actual designs produced by FabMem.

To identify the source of this discrepancy we examine how ORION estimates buffer area. From the ORION source[1], we note that ORION's area estimates do not account for decoders or sense amplifiers and only account for the SRAM data array. For the data presented in Figure 1, ORION area represents only the SRAM array while FabMem area includes all decoder and sense amplifier overheads. ORION's omission of decoder and output overheads would not explain the observed difference in Figure 1, and in fact would lead to an increased discrepancy.

Delving further, we observe that the primary source of disagreement originates from the assumed area required for a 2-port SRAM bit cell. Documented in Table I, ORION calculates the area of a 1R1W bit cell in 45nm as 9.2μm^2, whereas FabMem's equivalent bit cell is only 1.78μm^2. ORION's area is exceedingly conservative representing a 5.16x increase over that of FabMem. For reference, as shown in Table II, a TSMC 45nm D flip-flop would be 2.37x smaller than ORION's SRAM cell. Additionally, other academic publications present dual-ported SRAM cells in 45nm of less than 1μm^2 [23].

The calculation and parameters that ORION uses for computing the area of a bit cell are presented in Figure 2. Although the initial SRAM cell is sizable, the bit cell area is dominated ORION's assumed necessary 0.584μm spacing between word and bit lines. In comparison to FabMem's documented layouts, this spacing is unnecessarily large [21]. Of additional note, though this does not impact our results, ORION's bit cell height calculation is hard coded for two ports, whereas the width equation is generalized for any number of ports.

$$
\begin{aligned}
BitCellHeight &= RegHeight + 2 * WordLineSpacing \\
BitCellWidth &= RegWidth + 2 * BitLineSpacing \\
&\quad * (ReadPorts + WritePorts)
\end{aligned}
$$

Parameter	Value	Parameter	Value
RegHeight	1.848μm	WordLineSpacing	0.584μm
RegWidth	0.714μm	BitLineSpacing	0.584μm

Fig. 2. ORION SRAM Bit Cell Area Calculation

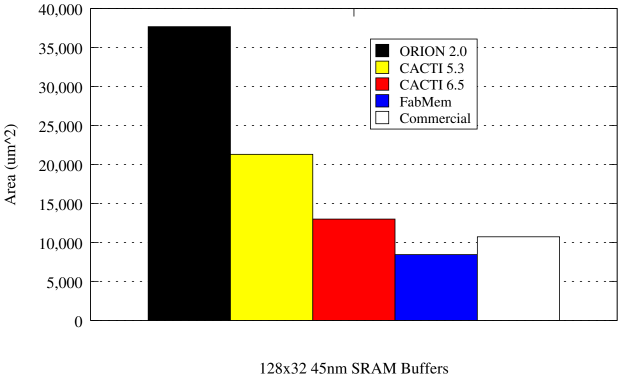


Fig. 3. SRAM Buffer Model Comparison.

For additional validation of our FabMem reference model we compared area models from ORION 2.0, CACTI 5.3, CACTI 6.5, and a commercial[2] memory compiler. Figure 3 shows the reported areas for an 128x32 SRAM-based buffer in 45nm technology for each model. Both ORION 2.0 and CACTI 5.3 report significantly more area than both memory compilers. Additionally, there is a large area difference between CACTI versions 5.3 and 6.5. Conversely, the memory compilers, FabMem and commercial, area results are within 27% of each other.

Overall, we find ORION's area modeling to be unrepresentative of 45nm SRAM technology. This is due to a combination of unnecessarily large sizing parameters, a general equation that does not accurately represent the incremental cost for additional word and bit lines, and a lack of modeling decoder and sense amplifier area.

C. Power Analysis

Figure 4 presents the dynamic energy required to perform a single read and write operation to SRAM buffers of varying sizes. This energy metric represents the per-hop dynamic energy due to buffering that a single flit would require in an interconnection network of a given bit width. Similar to the prior area analysis, we observe ORION's power estimates are higher than FabMem's ranging from 1.14x–1.66x (geomean 1.4x). In comparison to the area analysis though, the energy estimate is much closer. Unlike ORION's area model, ORION's power model computes decoder and sense amplifier overheads, resulting in more comparable numbers.

[1] SIM_router_area.c:56-64

[2] Unfortunately this memory compiler can not be named due to confidentiality agreements.

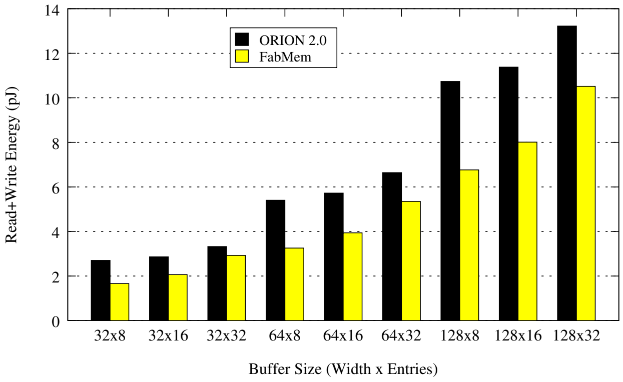


Fig. 4. Input Buffer Power

Despite properly modeling these overheads, the large bit cell assumption from the area analysis impacts the power analysis, resulting in additional capacitance due to greater bit and word line lengths.

Overall, we observe ORION's SRAM power model, although much more accurate than its area model, is also unrepresentative of $45\mathrm{nm}$ technology due to excessive bitline and wordline capacitances resulting from its area model. With an area model more characteristic of $45\mathrm{nm}$ technology, ORION's power model could be quite representative.

III. Crossbar

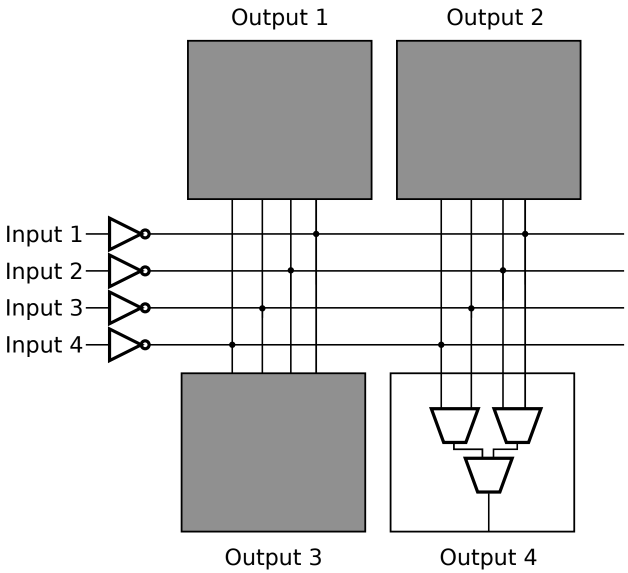
Crossbar area and power modeling are key for on-chip interconnection networks as they represent the majority of router datapath complexity. As noted earlier, many prominent interconnection network papers not only incorporate ORION's crossbar model, but utilize it to evaluate the tradeoffs between different crossbar radices. These evaluations depend on the assumption that ORION's crossbar model is scalable and that its modeling is consistent and comparable across multiple radices. In this section, after detailing ORION's modeling and an alternative crossbar model, we demonstrate that ORION's crossbar area estimates are an order of magnitude larger (30x) than a more refined design. Additionally, we show that ORION's area calculation is not consistent with the structural crossbar model. This results in dissonance between ORION's power and area models and prevents accurate modeling across different crossbar radices.

As a general tool, ORION is capable of modeling both tristate and multiplexer-based crossbars. Within this work we restrict our analysis to multiplexer-based crossbars.

A. ORION's Crossbar Structure

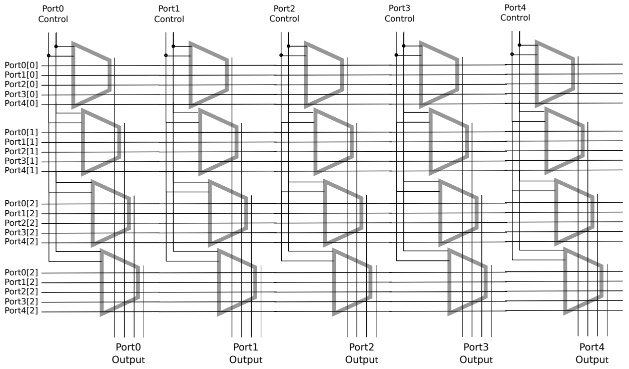
In this section we provide an overview of the multiplexer-based crossbar structure which ORION attempts to model. The definitive documentation on ORION's multiplexer-based crossbar is contained within an associated technical report [24].

Recreated from [24], Figure 5 presents the assumed crossbar structure. The overall structure consists of input wires routed horizontally that fan out vertically to output modules.



$$\begin{aligned} WireHeight &= Inputs * FlitWidth * WirePitch \\ WireWidth &= \tfrac{1}{2} Outputs * Inputs * FlitWidth \\ &\quad * TristateWidth \end{aligned}$$

Fig. 5. ORION Crossbar Structure



$$\begin{aligned} &Height = FlitWidth * Max(MuxHeight, 7 * WirePitch) \\ &Width = MuxWidth + 5 \\ &\qquad * Max((2 + FlitWidth) * WirePitch, MuxWidth) \end{aligned}$$

Fig. 6. Alternative Crossbar Structure. Specialized for a 5x5 dimension ordered network.

As all inputs are required to fan out, there are a total of $Inputs * FlitWidth$ input wires along the horizontal dimension. Output multiplexer modules may be placed above or below the input wire structure, allowing two output modules to share a vertical wire column to access a given input. If an odd number of outputs are required, the final output multiplexer is placed to the right of the input wire mesh. It is assumed that each output may need to multiplex from every input, thus there are also $FlitWidth * Inputs$ wires routed vertically to each output and every output module contains $FlitWidth$ multiplexer trees.

B. Alternative Crossbar Structure

For comparison against ORION's design, an alternative crossbar structure is presented in Figure 6. This crossbar

TABLE II
TSMC 45NM PARAMETERS

Parameter	Value
M4–M8 Minimum Width	0.07µm
M4–M8 Minimum Spacing	0.07µm
MUX4D4 Dimensions	5.18µm x 1.26µm
D Flip-Flop Dimensions	3.08µm x 1.26µm

design was utilized in [16], and was originally inspired by the crossbar structure of [25]. This instance of the design assumes a 5x5 dimension order routed (DOR) mesh network.

Similar to the ORION design, the crossbar assumes input wires are routed horizontally. Input wires are arranged in a bit-interleaved fashion and routed together in rows across output port columns. The height of an input row is equivalent to the height of a four input multiplexer or the vertical space required for seven metal tracks, whichever is greater. A four input multiplexer is sufficient for a 5x5 DOR network as no packet is ever routed out the port from which it originated. Seven metal tracks are required per row to route signals for the 5 input ports as well as 2 multiplexer control signals. It is assumed the multiplexer output may utilize one of the horizontal control wiring tracks, as the control signals should have been previously sourced.

For each output port column, two vertical wiring tracks are required to route control signals down to the horizontal tracks used for final control routing to multiplexers. Two signals are sufficient as all multiplexers in an output column are routing from the same input. Within each output column, one vertical wiring track per-row is required to route each output bit. Adjacent rows within a column are offset horizontally by a single wire pitch to align output routing.

The equations listed at the bottom of Figure 6 compute the crossbar dimensions. As can be seen, the crossbar height is dictated by the wire pitch or the height of a multiplexer depending on the technology parameters. For crossbars of sufficient bit width, the total width is dictated by the allowable pitch between vertical wires.

In the evaluation that follows we assume the TSMC 45nm parameters from Table II. The standard-cell based multiplexer's use through metal layer M3. Horizontal and vertical wires are routed along M5 and M6, leaving M4 for final connections to I/O pins.

C. Crossbar Comparison

Figure 7 presents the areas of 64-bit 5x5 mesh crossbars for our alternative crossbar model with 1x and 3x minimum wire spacing (ALT-1x and ALT-3x), the elastic-buffer (EB) router of [26], and ORION's multiplexer and tri-state models in 45nm technology. The alternative model and ORION numbers represent the crossbar area only, while the EB router area was taken directly from [26] and is for an entire router (crossbar, latches, and control). As seen in Figure 7, ORION's multiplexer-based crossbar requires 124,492μm^2 whereas ALT-1x requires only 4,143μm^2, representing a 30x area difference. At 14730μm^2, the entire EB router design is 8.5x smaller than what ORION estimates for the crossbar alone.

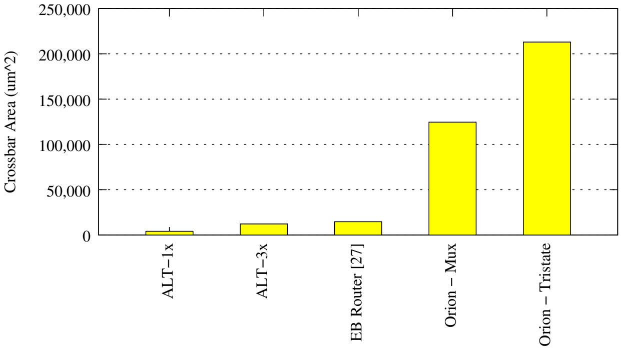


Fig. 7. Alternative 45nm 64-bit, 5x5 Crossbar Implementations. The EB Router [26] is a full router implementation while all other designs are crossbars only.

To see how such great area differences are possible, we revisit the layouts from Figures 5 and 6. Assuming inputs are approximately equal to outputs and that the widths and heights of the crossbars are mostly determined by the number of wires in each dimension, we can perform a complexity analysis using the equations listed at the bottom of these figures. For n ports of width w, the height of ORION's crossbar is $O(nw)$ and its width is on order of $O(n^2w)$, resulting in an area on the order of $O(n^3w^2)$. For the alternative layout, the height is approximately $O(w(n+\log n))$, due to the n signals per row and $\log n$ control signals needed. The alternative design's width is proportional to $O(nw)$, resulting in an area of $O(n^2w^2 + nw^2 \log n)) \approx O(n^2w^2)$. As ORION's n^3 term dominates for the 5x5 switch, the alternative layout results in less area.

Additionally, the TSMC wire parameters in Table II used for the alternative design are more aggressive than ORION's assumed spacing parameters, resulting in even greater area differences. ORION's assumed 45nm wire pitch is 0.174µm whereas TSMC allows for a 0.14µm pitch. Also, in the horizontal dimension ORION is limited not by wire pitch, but by the width of a tri-state gate[3], $TristateWidth$. As $TristateWidth$ is equal to 1.442µm, vertical wires are spaced 10x further apart than in the alternative design.

In total, the advantage due to better complexity and layout-related spacing parameters are why we observe such less area for the alternative crossbar in comparison to ORION-based crossbars.

D. Model Consistency

In the previous section we lied to you. Although the arguments for why the alternative crossbar layout is better than ORION's multiplexer-based crossbar are valid, the data presented within Figure 7 was generated with ORION's area model. In this section we show that ORION's area model

[3] See [24] for details of why this is necessary.

$$\begin{array}{rlcl}
1) & Depth & = & ceil(\log_4 Inputs) \\
2) & NumMux & = & \frac{4^{Depth}-1}{3} \\
3) & MuxArea & = & 1.5 * NumMux * AreaMux4 \\
4) & CrossbarArea & = & FlitWidth * MuxArea * Outputs \\
5) & & & *Inputs
\end{array}$$

Fig. 8. ORION Degree-4 Multiplexer Crossbar Calculation

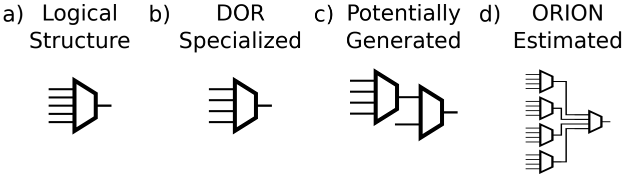


Fig. 9. Possible 5-input Multiplexer Implementations.

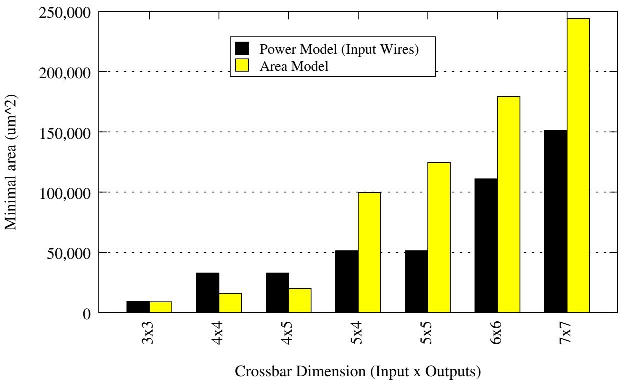


Fig. 10. ORION 2.0 Power and Area Model Comparison

is inconsistent with the structure presented in Figure 5 and ORION's own power model. Additionally we show that ORION's area model has other issues related to optimality and comparing crossbars of different dimensions.

Figure 8 shows how ORION's area model calculates crossbar area for degree-4 multiplexer-based crossbars[4]. We must step through this calculation to demonstrate how ORION's area model is unsuitable for comparing across multiple crossbar dimensions and is unrepresentative of ORION's crossbar model.

The general structure of Figure 8, is to compute the area of an output multiplexer structure in lines 1-3 and then use this area to compute the full crossbar area in lines 4-5. Line 4 computes the total area required for all output multiplexer trees by multiplying the multiplexer area by the total number of trees, $FlitWidth * Outputs$. However, line 5 performs the final area computation by multiplying the total output multiplexer area by the number of inputs. We believe this final multiplication step to be in error as we can find no justification for it from a structural or any other point of view.

There are also problems with how the first three lines of Figure 8 compute the area required for an n-wide multiplexer constructed out of 4-input multiplexers. The problems with this computation due to its non-optimal and general nature. It is suboptimal because line 2 considers only the depth of the multiplexer tree, thus ORION builds the same structure for a 16-input multiplexer as it does for a 5-input multiplexer. Additionally, ORION's generality also obscures potential optimizations. If dimension ordered routing had been assumed in a 5x5 crossbar, a 4-input multiplexer would suffice as no packet would ever travels out its arrival port. ORION only considers the general case though, and utilizes a 5-input multiplexer. Figure 9 shows the logical structure, alternative possible implementations, and what ORION generated for a 5-input multiplexer. These output multiplexer equations are why it is dangerous to use ORION's area model to compare crossbars of different radices. The previously mentioned suboptimality the line 2's $NumMux$ calculation causes ORION's area model to report that a 5x5 crossbar is more than 5x larger than a 4x5 crossbar.

Figure 10 demonstrates how the flaws in ORION's area model cause it to be dissonant with respect to the power model. To properly compute the capacitance of the input wires, ORION's power model utilizes the height and width equations displayed at the bottom of Figure 5. By multiplying these dimensions together, we effectively compute the area of the input wire mesh displayed in Figure 5. In Figure 10 we display the area of the input wires, as computed by ORION's power model, versus the area of the crossbar, as computed by ORION's area model. According to the structure of Figure 5, the input wire area should always be less than the total crossbar area. This constraint is broken as the area model reports less area than the power model for the 4x4 and 4x5 crossbars.

Additionally, scaling from the 4x5 to the 5x5 crossbar we note the previously predicted >5x jump in area reported by the area model. The power model's area grows by only 56% over this same transition. As the power model is closely tied to ORION's structural model, we feel this demonstrates the 5x increase is mostly an artifact of the suboptimality in Figure 8's multiplexer tree calculation.

E. Crossbar Summary

Overall, we have highlighted multiple issues with ORION's multiplexer-based crossbar implementation. Many of these issues relate design parameters and spacing constraints. Additionally we have shown how an alternative crossbar layout may avoid certain area overheads which are experienced by the ORION crossbar design. Finally, we pointed out issues relating to the ORION area model that limits its accuracy and usage.

IV. Conclusion

Within this paper we have performed a component-wise analysis of ORION's input buffer and crossbar modeling. Resulting from this analysis we observe that in its current form, ORION overstates the relative area and power of input buffers and crossbars by large factors. These overestimations originate from ORION's generality and abstraction, overly

[4] SIM_router_area.c:120-123

conservative design parameters, and some inconsistencies in modeling methodology.

Overstating the area and power of interconnection networks has the undesirable impact of affecting design tradeoffs. If these impacts are large enough, beneficial design changes may be discarded. As multiple of ORION's datapath components have been affected, numerous publications may have their results impacted.

On the positive side, we find that much of ORION's discrepancy can be rectified by simply reconsidering the parameters that are utilized. First, the estimated SRAM cell and other spacing parameters are not in line with 45nm process technology. By re-evaluating their base values and scaling factors, ORION's model could be brought closer in-line with current technology. Additionally, alternative crossbar layouts like the one presented here could be incorporated into ORION. Finally, ORION's crossbar area modeling issue could be rectified by incorporating many of the calculations already utilized within its power model.

In conclusion, we would like to thank the ORION authors for publicly releasing their tool, as public releases allow for peer review and the improvement of community models. It is our hope that by highlighting these issues, users of ORION can avoid certain pitfalls and ORION's models could incorporate solutions to these issues.

References

[1] G. Moore, "Cramming More Components Onto Integrated Circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82 –85, jan. 1998.

[2] R. Dennard, F. Gaensslen, V. Rideout, E. Bassous, and A. LeBlanc, "Design of ion-implanted MOSFET's with very small physical dimensions," *Solid-State Circuits, IEEE Journal of*, vol. 9, no. 5, pp. 256 – 268, oct 1974.

[3] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik, "Orion: a power-performance simulator for interconnection networks," in *Microarchitecture, 2002. (MICRO-35). Proceedings. 35th Annual IEEE/ACM International Symposium on*, 2002, pp. 294 – 305.

[4] A. Kahng, B. Li, L.-S. Peh, and K. Samadi, "Orion 2.0: A power-area simulator for interconnection networks," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 20, no. 1, pp. 191 –196, jan. 2012.

[5] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu, "Kilo-NOC: a heterogeneous network-on-chip architecture for scalability and service guarantees," in *Proceedings of the 38th annual international symposium on Computer architecture*, ser. ISCA '11. New York, NY, USA: ACM, 2011, pp. 401–412.

[6] T. Krishna, L.-S. Peh, B. M. Beckmann, and S. K. Reinhardt, "Towards the ideal on-chip fabric for 1-to-many and many-to-1 communication," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-44 '11. New York, NY, USA: ACM, 2011, pp. 71–82.

[7] A. K. Mishra, N. Vijaykrishnan, and C. R. Das, "A case for heterogeneous on-chip interconnects for CMPs," in *Proceedings of the 38th annual international symposium on Computer architecture*, ser. ISCA '11. New York, NY, USA: ACM, 2011, pp. 389–400.

[8] C. Fallin, C. Craik, and O. Mutlu, "CHIPPER: A low-complexity bufferless deflection router," in *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, feb. 2011, pp. 144 –155.

[9] M. Ahn and E. J. Kim, "Pseudo-Circuit: Accelerating Communication for On-Chip Interconnection Networks," in *Microarchitecture (MICRO), 2010 43rd Annual IEEE/ACM International Symposium on*, dec. 2010, pp. 399 –408.

[10] S. A. R. Jafri, Y.-J. Hong, M. Thottethodi, and T. N. Vijaykumar, "Adaptive Flow Control for Robust Performance and Energy," in *Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '43. Washington, DC, USA: IEEE Computer Society, 2010, pp. 433–444.

[11] A. Carpenter, J. Hu, J. Xu, M. Huang, and H. Wu, "A case for globally shared-medium on-chip interconnect," in *Proceedings of the 38th annual international symposium on Computer architecture*, ser. ISCA '11. New York, NY, USA: ACM, 2011, pp. 271–282.

[12] A. Udipi, N. Muralimanohar, and R. Balasubramonian, "Towards scalable, energy-efficient, bus-based on-chip networks," in *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, jan. 2010, pp. 1 –12.

[13] A. Bakhoda, J. Kim, and T. Aamodt, "Throughput-Effective On-Chip Networks for Manycore Accelerators," in *Microarchitecture (MICRO), 2010 43rd Annual IEEE/ACM International Symposium on*, dec. 2010, pp. 421 –432.

[14] S. Ma, N. Enright Jerger, and Z. Wang, "DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip," in *Proceedings of the 38th annual international symposium on Computer architecture*, ser. ISCA '11. New York, NY, USA: ACM, 2011, pp. 413–424.

[15] J. Li, W. Huang, C. Lefurgy, L. Zhang, W. Denzel, R. Treumann, and K. Wang, "Power shifting in Thrifty Interconnection Network," in *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, feb. 2011, pp. 156 –167.

[16] M. Hayenga and M. Lipasti, "The NoX router," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-44 '11. New York, NY, USA: ACM, 2011, pp. 36–46.

[17] G. Michelogiannakis, N. Jiang, D. Becker, and W. J. Dally, "Packet chaining: efficient single-cycle allocation for on-chip networks," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-44 '11. New York, NY, USA: ACM, 2011, pp. 83–94.

[18] Y. Xu, B. Zhao, Y. Zhang, and J. Yang, "Simple virtual channel allocation for high throughput and high frequency on-chip routers," in *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, jan. 2010, pp. 1 –11.

[19] A. B. Kahng, B. Lin, and K. Samadi, "Improved on-chip router analytical power and area modeling," in *Proceedings of the 2010 Asia and South Pacific Design Automation Conference*, ser. ASPDAC '10. Piscataway, NJ, USA: IEEE Press, 2010, pp. 241–246.

[20] R. Mullins, A. West, and S. Moore, "The design and implementation of a low-latency on-chip network," in *Design Automation, 2006. Asia and South Pacific Conference on*, jan. 2006, p. 6 pp.

[21] T. A. Shah, "FabMem: A Multiported RAM and CAM Compiler for Superscalar Design Space Exploration," Master's thesis, North Carolina State University, 2010.

[22] N. K. Choudhary, S. V. Wadhavkar, T. A. Shah, H. Mayukh, J. Gandhi, B. H. Dwiel, S. Navada, H. H. Najaf-abadi, and E. Rotenberg, "FabScalar: composing synthesizable RTL designs of arbitrary cores within a canonical superscalar template," in *Proceedings of the 38th annual international symposium on Computer architecture*, ser. ISCA '11. New York, NY, USA: ACM, 2011, pp. 11–22.

[23] H. Noguchi, S. Okumura, Y. Iguchi, H. Fujiwara, Y. Morita, K. Nii, H. Kawaguchi, and M. Yoshimoto, "Which is the best dual-port SRAM in 45-nm process technology? – 8T, 10T single end, and 10T differential," in *Integrated Circuit Design and Technology and Tutorial, 2008. ICICDT 2008. IEEE International Conference on*, june 2008, pp. 55 –58.

[24] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik, "A detailed architectural-level power model for router buffers, crossbars and arbiters," Princeton University, Tech. Rep., 2004.

[25] J. Balfour and W. J. Dally, "Design tradeoffs for tiled CMP on-chip networks," in *Proceedings of the 20th annual international conference on Supercomputing*, ser. ICS '06. New York, NY, USA: ACM, 2006, pp. 187–198.

[26] G. Michelogiannakis, J. Balfour, and W. Dally, "Elastic-buffer flow control for on-chip networks," in *High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on*, Feb. 2009, pp. 151 –162.