

# Network Coding for Routability Improvement in VLSI

Nikhil Jayakumar  
nikhil\_at\_ece.tamu.edu  
Sunil P Khatri  
sunilkhatri\_at\_tamu.edu

Kanupriya Gulati  
kgulati\_at\_ece.tamu.edu  
Alexander Sprintson  
spalex\_at\_ece.tamu.edu

Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX

## ABSTRACT

With the standard approach for establishing multicast connections over a network, network nodes are utilized to forward and duplicate the packets received over the incoming links. Recently, there has been a significant interest in a novel paradigm of *network coding*. Network coding generalizes the traditional routing approach by allowing the network nodes to generate new packets by performing algebraic operations on packets received over the incoming links. It has been shown that network coding can increase the throughput of multicast communication. In this paper, we explore the benefits of network coding for improving the routing characteristics of VLSI designs. We demonstrate that when data has to be routed across the IC, it is often beneficial to perform network coding. Initial results demonstrate that network coding can result in a healthy reduction in wire length, wire area, interconnect power as well as the active area associated with the interconnects. This comes at a small delay penalty.

## 1. INTRODUCTION

The standard approach for implementing multicast connections over a network is to forward or duplicate packets at the intermediate network nodes. This approach requires establishing a *Steiner tree* (trees) that connects the source node with the terminal nodes. In a typical scenario where data has to be multicast over a network, we utilize network nodes to simply forward and duplicate incoming messages. Recently, there has been considerable interest in a novel technique of *network coding* [1] that allows the intermediate nodes to generate new packets by performing algebraic operations on packets received over the incoming links. The network coding technique generalizes the traditional routing approach and requires more powerful network nodes, which have the capability of performing encoding. It has been shown [2, 3] that network coding can result in a significant increase in the throughput of multicast connections, compared to the traditional routing approach. In particular, the *coding advantage*, i.e., the gain in the throughput achievable by using network coding has been shown to be as large as  $\Omega((\log(n)/\log \log n)^2)$  and  $\Omega(\sqrt{k})$  [2, 3], where  $n$  is the

number of nodes and  $k$  is the number of terminals in the network.

In this paper, we explore the applicability of network coding for improving routing characteristics of VLSI designs. Many signals in a VLSI chip can be viewed as instances of multicast communication. Any signal with a fanout of more than one is, in the most general sense, an example of multicast communication within an IC. However, not all such on-chip multicast instances may benefit from network coding. Since network coding requires additional coding logic in the network, the on-chip applicability of such a technique is somewhat hampered for situations where wires are short. This is because the overhead due to the coding logic may overshadow the gains obtained by wire-length reductions. Scenarios where network coding may be beneficial on-chip occur in the routing of multicast signals using long buses. For bus routing in current day ICs, buffer insertion [4] is required for reducing the wiring delay. With decreasing feature size, the amount of power consumed by interconnects exceeds that consumed by logic [4, 5]. If network coding was used in such scenarios, the additional logic area required for performing encoding can be effectively negligible compared to the traditional routing case. In addition, the wire-length reduction obtained due to network coding yields a power reduction as well.

## 2. PREVIOUS WORK

There has been significant interest in the area of network coding in recent times [3, 6–11]. These papers have addressed important network coding related topics such as network capacity, algorithms for network coding, network information flow, coding advantage, etc.

Establishing efficient multicast connections is one of the central problems in network coding. In the *multicast network coding problem* a source  $s$  needs to deliver  $h$  symbols to a set  $T$  of  $k$  terminals over the underlying communication graph  $G$ . It was shown in [1] and [12] that the capacity of the network, i.e., the maximum number of packets that can be sent between  $s$  and  $T$  per time unit, is equal to the minimum size of a cut that separates the source  $s$  from a terminal  $t \in T$ . Specifically, a source  $s$  can send  $h$  symbols to all terminals  $T$  if and only if the total capacity of all edges in any cut that separates  $s$  and  $t \in T$  is at least  $h$ . Li et al. [12] proved that *linear network codes* are sufficient for achieving the capacity of the network. In a subsequent work, Koetter and Médard [13] developed an algebraic framework for network coding. This framework was used by Ho et al. [14] to show that linear network codes can be efficiently constructed through a randomized algorithm. Jaggi et al. [11] proposed a deterministic polynomial-time algorithm for finding feasible network codes for multicast networks.

To the best of the authors' knowledge, this is the first paper to explore the applicability of network coding in the VLSI context.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCAD 2006, November 5-9, 2006, San Jose, CA.

Copyright 2006 ACM 1-59593-389-1/06/0011 ...\$5.00.

### 3. PRELIMINARIES AND TERMINOLOGY

The interconnection network is represented by a directed graph  $G = (V, E)$ , where  $V$  is the set of nodes and  $E$  the set of edges in  $G$ . We assume that each edge  $e \in E$  is of unit capacity, (i.e., can transmit one symbol per time unit). A *cut* of the graph  $G = (V, E)$  is a partition  $(V_1, V_2)$  of  $V$  ( $V_1, V_2 \subseteq V, V_2 = V \setminus V_1$ ). We say that a cut  $(V_1, V_2)$  separates node  $s$  from node  $t$  if  $s \in V_1$  and  $t \in V_2$ . The size of a cut  $(V_1, V_2)$  is equal to the number of edges that connect nodes in  $V_1$  to nodes in  $V_2$ .

Our goal is to establish a multicast connection between the source node  $s \in V$  and a set  $T \subset V$  of terminals. The connection must transmit  $h$  symbols per time unit to any terminal  $t_i \in T$ . We assume that each symbol is an element of a finite field  $\Sigma$ .

**DEFINITION 1 (INTEGER NETWORK CODE  $\mathbb{F}(\mathbb{N})$ ).** A network code for multicast network  $(G(V, E), s, T, h)$  is defined by the set of encoding functions  $\mathbb{F}(\mathbb{N}) = \{f_e \mid e \in E\}$ . If  $e(v, u)$  is an outgoing edge of the source node  $s$ , then  $f_e$  is a mapping from  $\Sigma^h$  to  $\Sigma$ . Otherwise,  $f_e$  is a mapping from  $\Sigma^{d_{in}(v)}$  to  $\Sigma$ , where  $d_{in}$  is the input degree of node  $v$ .

The encoding function  $f_e$  of  $e(v, u)$  determines the symbol transmitted on edge  $e$  for any possible combination of the symbols available at the source (if  $v = s$ ) or received over the incoming edges of  $v$  (if  $v \neq s$ ). We focus on linear network codes  $\mathbb{F}(\mathbb{N})$ , i.e., for each  $e \in E$  the encoding function  $f_e$  is a linear function over  $\Sigma$ . With linear network coding, each symbol transmitted over edge  $e \in E$  is a linear combination of the  $h$  symbols available at source  $s$  over finite field  $\Sigma$ . A network code  $\mathbb{F}(\mathbb{N})$  is said to be feasible if for each destination node  $t \in T$ , there exists a decoding function  $g_t : \Sigma^{d_{in}(t)} \mapsto \Sigma^h$  that enables each destination to decode the original packets transmitted by the source node  $s$ , where  $d_{in}(t)$  is the input degree of node  $t$ .

It was shown in [1] and [12] that the maximum rate  $h$  achievable by a multicast connection between source  $s$  and set  $T$  of terminals is equal to the minimum size of a cut that separates source  $s$  and a terminal  $t \in T$ . Menger's theorem [15] implies that a minimum size of a cut that separates  $s$  and  $t$  is equal to the maximum number of *edge-disjoint* paths between  $s$  and  $t$ . Thus, any multicast network that sends  $h$  symbols from  $s$  to  $T$  must include  $h$  edge-disjoint paths between  $s$  and each terminal  $t \in T$ .

### 4. OUR APPROACH

We first present an algorithm due to [11] that receives as input an acyclic coding network  $\mathbb{N}(G, s, T, h)$  and computes a feasible integer network code for  $\mathbb{N}$  over a field of size  $k = |T|$ . This algorithm can be used to find encoding functions for all edges of an arbitrary network.

The algorithm has two stages. In the first stage, we find, for each terminal  $t_i \in T$ , a set  $P_i$  of  $h$  disjoint paths between  $s$  and  $t_i$ . In the second stage, we visit each node  $v$  of the graph  $G$  and determine the encoding function of all outgoing edges of  $v$ . We visit the edges in topological order, which ensures that when edge  $e$  is visited, the encoding functions are already determined for all edges leading to  $e$ . The encoding function of each edge  $e \in E$  is chosen in such a way that all terminals that lie downstream of  $e$  receive  $h$  linearly independent symbols.<sup>1</sup>

The algorithm maintains the following data structures for each terminal  $t \in T$ :

<sup>1</sup>It was shown in [11] that such an encoding function exists if the finite field is sufficiently large (greater than the number of terminals).

ALGORITHM NETCOD ( $\mathbb{N}(G, s, T, h)$ ).

**Input:**  $\mathbb{N}$  - a multicast network;

```

1  Insert a new source  $s'$  into  $V$ 
2  Insert  $h$  parallel edges  $\{e_1, \dots, e_h\}$  in  $E$ 
   between  $s'$  and  $s$ ;
3  for each terminal  $t_i \in T$  do
4    Let  $P_i$  be a set of edge-disjoint paths between  $s$  and  $t_i$ 
5  for each edge  $e_j \in \{e_1, \dots, e_h\}$  do
6     $b_{e_j} = [0^{j-1}, 1, 0^{h-j}]$ 
7  for each terminal  $t_i \in T$  do
8     $C_{t_i} = \{e_1, \dots, e_h\}$ 
9     $B_{t_i} = \{b_{e_1}, \dots, b_{e_h}\}$ 
10 for vertex  $v \in V \setminus \{s\}$  in topological order
    towards  $T$  do
11   for each outgoing edge  $e$  of  $v$  do
12     for each terminal  $t_i \in T$  do
13       Let  $e'_i$  be the edge that precedes  $e$  on a path
          $p \in P_i$  between  $s$  and  $t_i$ 
14       Choose a linear encoding function  $f_e$  such that
         for each terminal  $t_i \in T$  it holds that
          $(B_{t_i} \setminus \{b_{e'_i}\}) \cup \{b_e\}$  is a set of linearly independent
         vectors, where  $b_e = f_e(b_{e'_1}, \dots, b_{e'_{d_{in}(v)}})$ 
15   for each terminal  $t_i \in T$  do
16      $B_{t_i} \leftarrow (B_{t_i} \setminus \{b_{e'_i}\}) \cup \{b_e\}$ 
17      $C_{t_i} \leftarrow (C_{t_i} \setminus \{e'_i\}) \cup \{b_e\}$ 

```

**Figure 1: Algorithm NETCOD**

1. A set  $C_t \in E$  that includes the most recently processed edge for each path  $p \in P_i$ .
2. A set  $B_t$  that includes the linear combinations (of the original packets) transmitted by edges in  $C_t$ .

The algorithm maintains the following invariant.

**INVARIANT 1.** At any step of the algorithm, each set  $B_t, t \in T$  contains  $h$  linearly independent vectors.

This invariant ensures that each terminal  $t \in T$  receives a set of linearly independent vectors, which, in turn, ensures that each terminal is able to decode the symbols sent by the source node  $s$ . The detailed description of the algorithm appears in Figure 1. The computational complexity of the algorithm is  $O(|E|kh + |V|k^2h^2(k+h))$  [7, 11].

In a typical network coding application, an underlying communication graph  $G$  represents the connectivity of the sources, sinks, and intermediate nodes. Whenever different packets are received at an intermediate node, a network coding opportunity presents itself. As such, in an IC, routes with fanout greater than or equal to two present network coding opportunities, when they run alongside other similar routes. The most common scenario in an IC where network coding may be applied is one in which  $h$  wires (or buses) are to be multicast to a set  $T$  of  $k$  terminals. Consider a modern microprocessor IC. The instruction and data buses represent two multicast sources ( $h = 2$ ). These buses are routed to a set  $T$  of  $k$  functional units, all of which require the information of both buses. The detailed routing of the instruction and data buses can be considered to induce a graph  $G$ , with nodes at the following locations:

- The positions of the source and sink nodes.
- The endpoints of routing segments of one or more multicast trees which overlap (i.e., run in parallel).

An example is shown in Figure 2 (i), which shows the decoded instruction ( $I$ ) and data ( $D$ ) buses routed to two terminal locations  $t_1$  and  $t_2$ . The induced graph  $G$  is shown in Figure 2 (ii). Note that the instruction and data buses have an overlapping segment (i.e., parallel segment) along the middle of the chip. This induces two intermediate nodes  $A$  and  $B$  in  $G$ . In particular, the node  $A$  receives different signals along its two incoming edges. Hence this presents a network coding opportunity. Figure 2 (iii) illustrates how this network coding opportunity is utilized, to yield a design with smaller wire-length and routing area. In this figure, node  $A$  receives  $i$  and  $d$ , but computes and forwards  $i \oplus d$ . As a consequence:

- Sink  $t_1$  receives  $i$  and  $i \oplus d$ , from which both  $i$  and  $d$  can be retrieved, as required.
- Sink  $t_2$  receives  $d$  and  $i \oplus d$ , from which both  $i$  and  $d$  can be retrieved, as required.
- The total wire-length is reduced, since the two separate  $i$  and  $d$  wires in Figure 2 (ii) are replaced by one wire in Figure 2 (iii). This reduces chip area and routing congestion as well.

In general, several important nets (and buses) in an IC form such multicast trees. When their routing is considered, we can extract the induced connectivity graph  $G$ . We then apply the Algorithm NETCOD on the graph  $G$ , to derive the logic at a coding node.

## 5. EXPERIMENTAL RESULTS

We demonstrate the efficacy of network coding in the IC routing context by applying the technique to two representative test case designs. For each test case, we assumed a  $0.1\mu\text{m}$  BPTM [16] process, with  $VDD = 1.2\text{V}$ . Each test case was a chip, with dimensions  $1\text{cm} \times 1\text{cm}$ . We considered two buses (the decoded instruction bus which is driven by the instruction decode unit, and the data bus which is driven from the data cache). Buses were routed on METAL3 and METAL4, and bus wires were  $0.5\mu\text{m}$  wide in all cases. The topology of each test case is shown in Figures 3 and 4. Each test case consists of the dark wires representing the instruction decode bus, and the light wires representing the data bus.

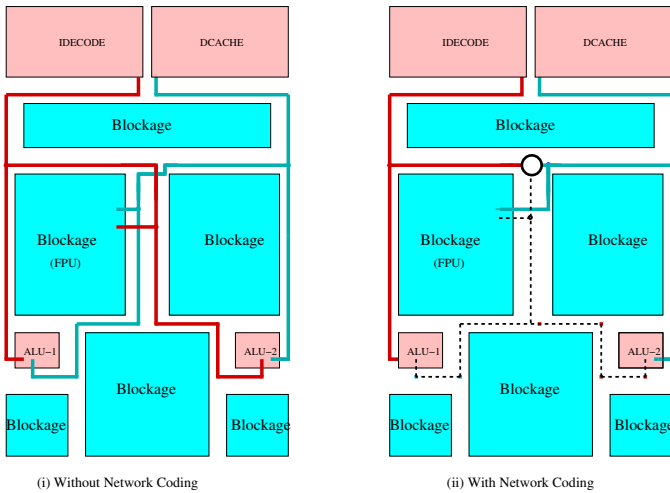


Figure 3: Test Case 1

Figures 3 (i) and 4 (i) represent the traditional routing scenario for the I and D buses. In Figure 3, both buses are required at three

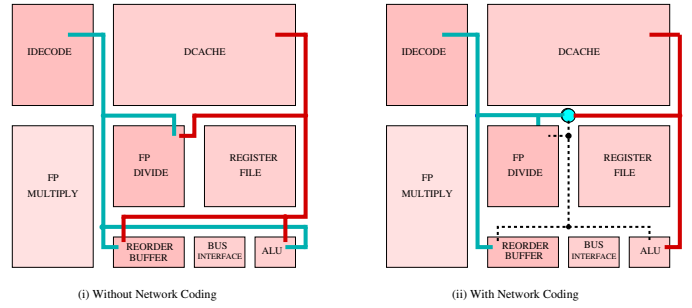


Figure 4: Test Case 2

sinks – the two ALUs and the FPU. In Figure 4, both the buses are required at the ALU, the FP divider and the Reorder buffer. Figures 3 (ii) and 4 (ii) represent the same design, after network coding. Note that the I and D buses have run in parallel through the center of the die, allowing a network coding opportunity. The coding logic is represented as a circle, and in each case, this logic computes the bit-wise XOR of the I and D buses. Note that in Figures 3 (ii) and 4 (ii), only a single bus is transmitted through the center of the chip. Hence there is a saving in the total wire-length of the buses, as well as in the total chip area (even though there is a slight increase in active area due to the presence of coding nodes in Figures 3 (ii) and 4 (ii)). Also, routing congestion is eased as well. After network coding, only one bus (rather than the two buses required without network coding) is driven through the center of the chip. The removal of the one bus helps achieve a power reduction due to the reduced capacitance and due to the removal of the buffers on the bus. Finally, the delay of the test cases shown in Figures 3 (ii) and 4 (ii) is only minimally higher than that of their traditionally routed counterparts. This is because long interconnect in present-day VLSI ICs typically dominate delays, and a small increase in logic delay in the interconnect path therefore has a negligible delay consequence.

The results for both test cases are presented in Tables 1 and 2. In these tables, the first column represents the routing style. The quantities of the remaining columns, and their units are self-explanatory. Table 1 represents results for test case 1, while Table 2 represents the results for test case 2. We note that for the two test cases, network coding results in a reduced power of between 7% and 8.5%. There is also a reduction in total wire length and area (which are in the range of 6% to 10%). Active area of the logic associated with the bus drivers drops by between 2% and 2.5% after network coding. This is because the reduction in wire-length results in a reduction in the number of buffers required, yielding a lower active area (in spite of the fact that the network coding requires additional logic). Finally, network coding results in a small delay penalty in the bus (of about 2% to 3%). This is because of the additional delay resulting from the coding operation. However, since the delay of buses is dominated by interconnect, the impact of this increase is minimal, as column 3 illustrates. Overall, network coding provides an improvement in power, wire length and area, at a small delay penalty.

## 6. CONCLUSIONS

In networking and communication, when data is to be multicast over a network, the nodes are typically utilized to perform forwarding and duplication of their incoming messages. In recent times, there has been considerable interest in utilizing the network nodes to perform coding (i.e., to perform more operations than simple forwarding and duplication). This is broadly referred to as network coding. In this paper, we explored the utility of network coding to

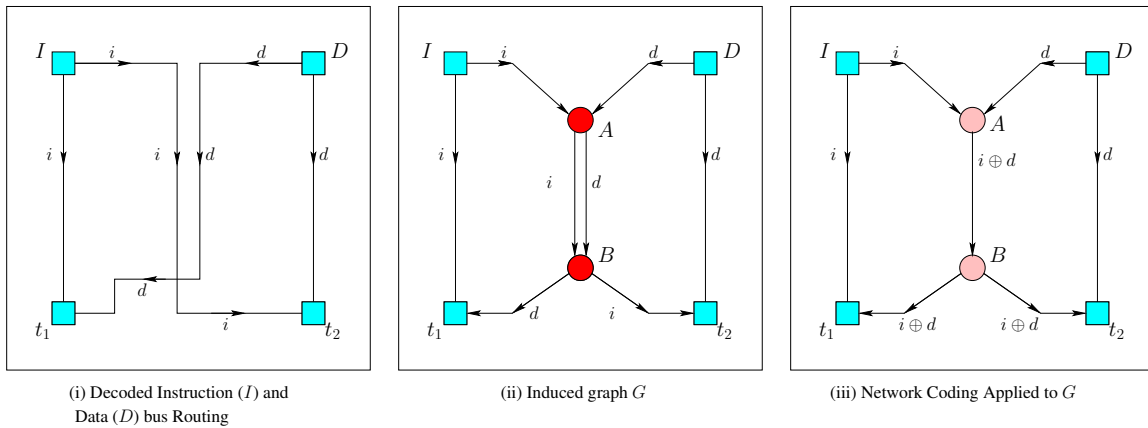


Figure 2: Generating the graph  $G$  for an IC

Style	Power (mW)	Delay (ps)	Wire length ( $\mu\text{m}$ )	Wire Area ( $\mu\text{m}^2$ )	Active Area ( $\mu\text{m}^2$ )
No coding	1.376	3016.73	50000	25000	35.0
Ntk coding	1.258	3116.37	45000	22500	34.14
Gain	8.58%	-3.3%	10%	10%	2.46%

Table 1: Results for Test Case 1

Style	Power (mW)	Delay (ps)	Wire length ( $\mu\text{m}$ )	Wire Area ( $\mu\text{m}^2$ )	Active Area ( $\mu\text{m}^2$ )
No coding	1.378	3774.75	65000	32500	45.5
Ntk coding	1.284	3854.14	60000	30500	44.64
Gain	6.82%	-2.1%	6.15%	6.15%	1.89%

Table 2: Results for Test Case 2

improve routing characteristics for VLSI designs. From our experiments, we observe that network coding, when applied to buses in a modern IC, can result in a decrease in interconnect power, wiring length and area, as well as active area associated with the bus logic. There is a slight increase in delay associated with network coding.

## 7. REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [2] C. Chekuri, C. Fragouli, and E. Soljanin, "On Average Throughput Benefits and Alphabet Size in Network Coding," *IEEE Transactions on Information Theory*, vol. 52, pp. 2410–2424, June 2006.
- [3] M. Charikar and A. Agarwal, "On the Advantage of Network Coding for Improving Network Throughput," in *Proceedings of IEEE Information Theory Workshop, San Antonio, 2004*.
- [4] "The International Technology Roadmap for Semiconductors." <http://public.itrs.net/>, 2003.
- [5] H. Wang, A. Papanikolaou, M. Miranda, and F. Catthoor, "A Global Bus Power Optimization Methodology for Physical Design of Memory Dominated Systems by Coupling Bus Segmentation and Activity Driven Block Placement," in *Proceedings, Asia and South Pacific Design Automation Conference (ASPDAC)*, pp. 759–761, Jan 2004.
- [6] M. Langberg, A. Sprintson, and J. Bruck, "The Encoding Complexity of Network Coding," *IEEE Transactions on Information Theory*, vol. 52(6), pp. 2386–2397, June 2006. (The joint special issue of the IEEE Transactions on Information Theory and the IEEE/ACM Transactions on Networking and Information Theory).
- [7] M. Langberg, A. Sprintson, and J. Bruck, "Network Coding: A Computational Perspective," in *The 40th Annual Conference on Information Sciences and Systems (CISS)*, March 2006. (Invited paper).
- [8] K. Bhattad, N. Ratnakar, R. Koetter, and K. Narayanan, "Minimal Network Coding for Multicast," in *Proceedings of the IEEE International Symposium on Information Theory*, pp. 1730–1734, September 2005.
- [9] D. Lun, M. Médard, T. Ho., and R. Koetter, "Network Coding with a Cost Criterion," in *Proceedings of International Symposium on Information Theory and its Applications (ISITA 2004)*, October 2004.
- [10] C. Fragouli and E. Soljanin, "Information Flow Decomposition for Network Coding," *IEEE Transactions on Information Theory*, vol. 52, pp. 829–848, March 2006.
- [11] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction," *IEEE Transactions on Information Theory*, vol. 51, pp. 1973–1982, June 2005.
- [12] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear Network Coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371 – 381, 2003.
- [13] R. Koetter and M. Medard, "An Algebraic Approach to Network Coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782 – 795, 2003.
- [14] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The Benefits of Coding over Routing in a Randomized Setting," in *Proceedings of the IEEE International Symposium on Information Theory*, 2003.
- [15] K. Menger, "Zur allgemeinen Kurventheorie," *Fund. Math*, vol. 10, pp. 95–115, 1927.
- [16] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu, "New Paradigm of Predictive MOSFET and Interconnect Modeling for Early Circuit Design," in *Proc. of IEEE Custom Integrated Circuit Conference*, pp. 201–204, Jun 2000. <http://www-device.eecs.berkeley.edu/~ptm>.