# Short Papers

## I/O Clustering in Design Cost and Performance Optimization for Flip-Chip Design

Hung-Ming Chen, I-Min Liu, and Martin D. F. Wong

*Abstract*—Input–output (I/O) placement has always been a concern in modern integrated circuit design. Due to flip-chip technology, I/O can be placed throughout the whole chip without long wires from the periphery of the chip. However, because of I/O placement constraints in design cost (DC) and performance, I/O buffer planning becomes a pressing problem. During the early stages of circuits and package co-design, I/O layout should be evaluated to optimize DC and to avoid product failures. The objective of this brief is to improve the existing/initial standard cell placement by I/O clustering, considering DC reduction and signal integrity preservation. The authors formulate it as a minimum cost flow problem that minimizes $\alpha W + \beta D$, where $W$ is the I/O wirelength of the placement and $D$ is the total voltage drop in the power network and, at the same time, reduces the number of I/O buffer blocks. The experimental results on some Microelectronics Center of North Carolina benchmarks show that the author's method averagely achieves better timing performance and over 32% DC reduction when compared with a conventional rule-of-thumb design that is popularly used by circuit designers.

*Index Terms*—Chip-package co-design, flip-chip design, input–output (I/O) planning, signal integrity.

## I. INTRODUCTION

With today's advanced integrated circuit (IC) manufacturing technology in deep-submicrometer (DSM) environment, we can integrate entire electronic systems on a single chip (SoCs). Because more input–output (I/Os) are needed in current designs, I/O placement has been a major concern in designing high-performance ICs. Flip-chip and multichip module (MCM) technologies now allow high-performance ICs and microprocessors to be built with more I/O connections than in the past [1], [2], among which area-array bonded connection (Fig. 1) is considered a better choice [3], [4]. Because area-array style allows I/O buffers to be placed anywhere on the die, we need to be aware of I/O buffer placement constraints to improve the design. Another consideration in modern methodology is the cost for placing I/O buffer blocks, which are sets of I/O buffers adjacently placed (as shown in Fig. 1), in the core.

There were some approaches/methodologies for this problem. In [5]–[8], similar methodologies for I/O cell placement and electrical checking using flip-chip technology have been presented. Some of them have graphic or interactive I/O placement tool to provide some constraint checking, trying to avoid hot-spot problems. Recently, Kozhaya *et al.* [9] further developed a greedy algorithm to place I/O buffers in an integer linear programming (ILP) formulation of
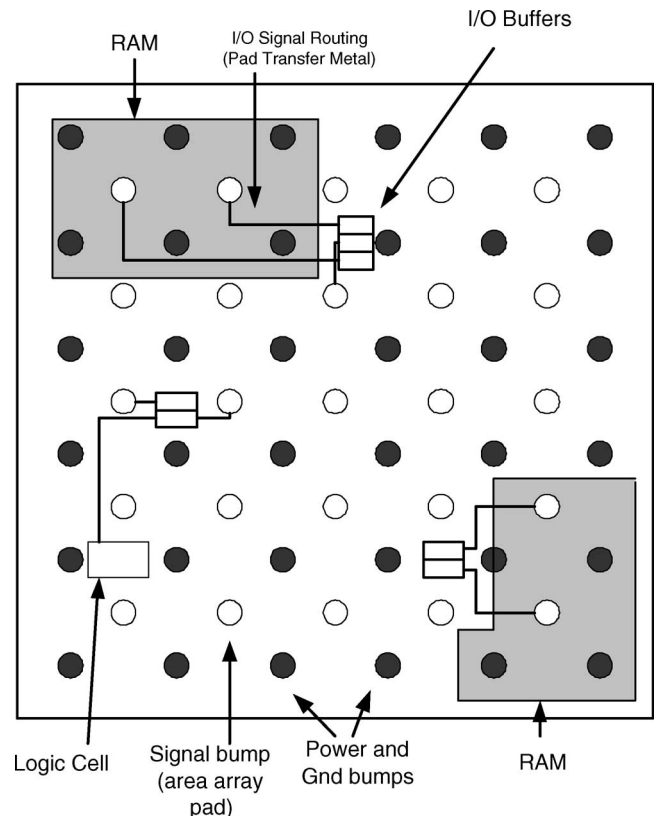
Fig. 1. Area-array footprint ASIC. The Vdd and Gnd bumps are uniformly distributed across the die with signal bumps in fixed interspersed locations. I/O buffers are associated with some specified signal bumps and connected by pad transfer metal.

voltage drop constraint. In [10], Lomax *et al.* utilized area I/O flip-chip packaging for minimizing interconnect length, which is a major metric for cell and I/O placement optimization. However, those approaches failed to consider the building cost of I/O buffer blocks.

I/O buffers usually come with a peripheral circuitry such as testing logic and electrostatic discharge (ESD). Thus, there is a clearance region for standard cells outside of I/O buffers. With I/O buffers clustered in one slot (buffer block), the clearance region is shared. In addition, power-routing design is usually done with special care. With I/O buffers clustered, the design cost (DC) for power routing is reduced. If we just place I/O buffers in greedy ways [7], [11], more I/O buffer blocks will be generated, thus, the DC will be increased. Therefore, during the early stages of co-design of IC and package [12], [13], the quality of I/O layout should be emphasized in the design flow [14].

In this paper, we study the problem of I/O clustering for flip-chip design and propose an algorithm to solve the problem with respect to DC and performance optimization while preserving signal integrity. Our approach is good for cell-based and block-based designs. Our objective is to reduce the number of I/O buffer blocks and to estimate their positions in an existing standard cell placement. We formulate it as a min-cost maximum flow problem that minimizes $\alpha W + \beta D$, where $W$ is the I/O wirelength of the placement and $D$ is the total voltage drop in the power network. This can be used in postplacement optimization or interim/evaluation step in a performance-driven placement methodology.

The rest of this paper is organized as follows. Section II describes the I/O placement considerations and problem formulation. Section III presents the algorithm for I/O clustering in DC and performance optimization. Section IV presents the experimental results and discussion, and Section V concludes this paper.

## II. AREA-ARRAY I/O BUFFER PLACEMENT IN DESIGN COST-CONSTRAINED AND PERFORMANCE-DRIVEN PLACEMENT METHODOLOGY

To keep up the performance in technology advancement, concurrent design of chip packaging and very large scale integration (VLSI) systems is applied to satisfy system specification and to optimize the DC [12]–[14]. Flip-chip technology allows high-performance ICs and microprocessors to be built with more power and I/O connections than in the past. To completely take advantage of this technology, we need to focus on the placement of highly power-hungry buffers, namely I/O buffers.

The design will suffer mainly from hot-spot problem [9] and long interconnect length [10] if I/O buffers are not carefully planned. From the footprint of application-specified IC (ASIC) in area-array design [5, Fig. 1], I/O buffers are placed near signal bumps, one I/O buffer is connected to one signal bump. Those buffers also need to be placed near the power bump to consume power to avoid large current–resistance $(IR)$ drop[1] and long interconnections. Furthermore, some areas cannot be used for placing I/O buffers such as RAMs. Kozhaya *et al.* [9] list some of the primary I/O placement constraints that mainly keep voltage drop below the threshold in power sources when placing I/O buffers.

On the other hand, generating a minimal number of I/O buffer blocks is another major objective during cell placement for flip-chip design. If we can cluster I/O buffers, the clearance region for testing logic and ESD purposes can be shared, and power-routing cost can be reduced. Otherwise, we will face solutions with more I/O buffer blocks by using greedy and intuitive approach, and the DC is inevitably increased. Therefore, we need to find a way to handle the tradeoff between power distribution constraint violation, wirelength estimation, and DC. In the succeeding subsections, we discuss I/O buffer placement for flip-chip design and problem formulation. Note that we can add this approach to an existing design flow in [11] to present a more complete methodology in DC and performance optimization (Fig. 2).

### A. I/O Buffer Placement for Flip-Chip Design

The analysis of the effect of I/O placement on the performance of power grids requires the modeling of the grids as well as that of the power sources and drains [9], [15]. For an efficient analysis of power supply network, power grids are modeled as linear resistance–capacitance $(RC)$ networks, power sources are modeled as simple constant voltage sources, and power drains are modeled as independent time-varying currents (Fig. 3).

The behavior of the system can be expressed in the modified nodal analysis (MNA) [16] formulation as the following ordinary differential equation (ODE):

$$Gx + C\dot{x} = u(t) \tag{1}$$

where $x$ is a vector of node voltages and source currents, $G$ is the conductance matrix, $C$ includes the capacitance terms, and $u(t)$ includes the contributions from the sources and the drains. Applying
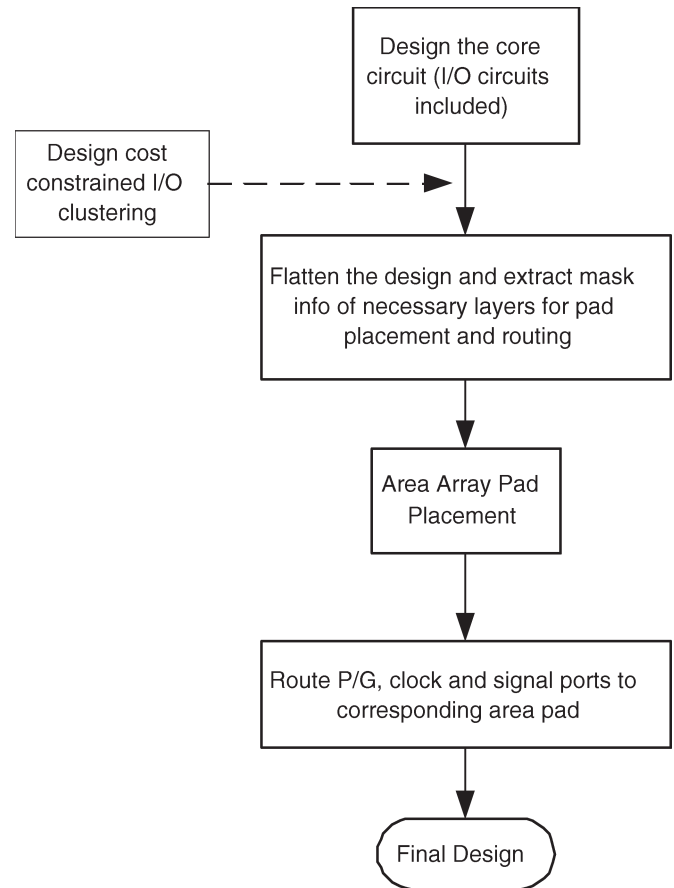


Fig. 2. Intrinsic area-array pad placement and routing flow from [11] and proposed clustering step.
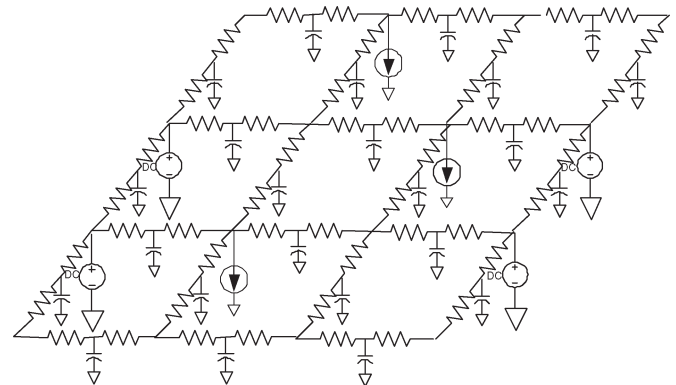


Fig. 3. Power supply network in area-array design for efficient analysis. Power grids are modeled as linear $RC$ networks, power sources are modeled as simple constant voltage sources, and power drains are modeled as independent time-varying currents.

a backward Euler (BE) numerical integration, we can express the resultant linear equations as

$$Ax(t + h) = u(t + h) + x(t)C/h \tag{2}$$

where $A = G + C/h$. The system matrix $A$ can be shown to be symmetric and further reformulated to be a nonsingular $\mathcal{M}$ matrix [17]. Because the direct current solution is a prudent, conservative, and practical approach to the problem, the system equation becomes $Ax = u$, where $A = G$. Considering the voltage drop in power grids, we reformulate the equation as $A\delta = b$, where $\delta = V - x$ is the vector

---

[1]In this paper, we only discuss the $IR$ drop constraint for I/O placement; however, other devices in VLSI design also have this constraint.
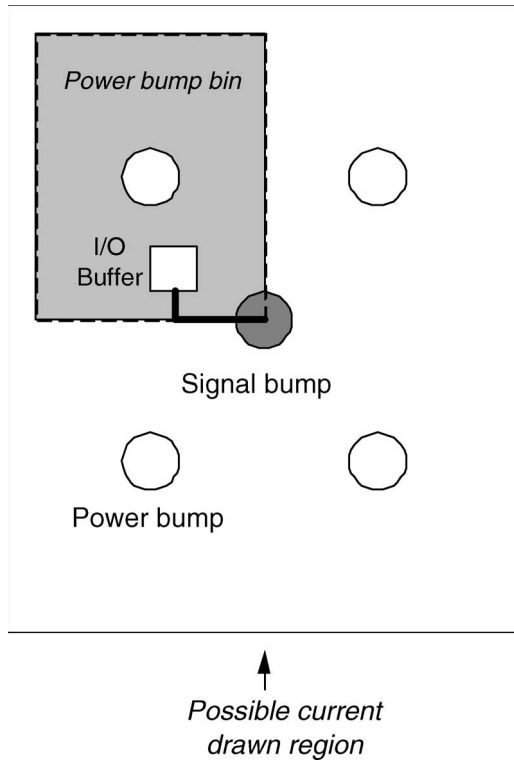
Fig. 4. Relationship between signal bump, power bump, power bump bin, I/O buffer possible positions, and possible current drawn region.

of the voltage drops, and $b$ is the vector of the current sources. In other words, $b$ can be expressed as

$$b_i = \sum_{k=1}^{n} d_{ik} I_k \qquad \forall i \qquad (3)$$

where $I_k$ is the current associated with buffer $io_k$, $d_{ik} = 1$ if $io_k$ consumes the power from node $p_i$; otherwise, $d_{ik} = 0$, and $n$ is the number of I/O buffers. Therefore, the relationship between the voltage drop at node $p_j$ and all the entries of the vector $b$ is expressed as

$$\delta_j = \sum_{i=1}^{m} a_{ji}^{-1} b_i \qquad \forall j \qquad (4)$$

where $a_{ji}^{-1}$ is the element on row $j$ and column $i$ of the inverse of the system matrix $A^{-1}$, and $m$ is the number of nodes and the dimension of matrix $A$. The problem can be formulated by planning a given set of I/O buffers as clusters (I/O buffer blocks) while suppressing the voltage drop to be under the user-specified voltage drop thresholds, which are denoted by $\delta_{\max}$.[2]

### B. Problem Formulation

*1) Problem —I/O Clustering in DC and Performance Optimization (ICDCPO):* Given an existing/initial standard cell placement, a set of I/O buffers that has a corresponding set of signal bumps IO = $\{io_1, \ldots, io_n\}$ and the current $I_i$ associated with I/O buffer $io_i$, a set of power bumps $P = \{p_1, p_2, \ldots, p_m\}$, a user-specified voltage drop threshold vector $\delta_{\max}$, the system matrix $A$ for power network, a certain building cost for I/O buffer blocks, and a set of nets $N = N_1 \cup N_2 \cup \cdots \cup N_k$, find a solution to simultaneously suppress the DC,

---

[2]We ignore the internal currents from functional blocks in the circuit due to the analysis of severe voltage drops in I/O buffers. Therefore, the drop thresholds are set up based on the drops for the I/O buffer part.
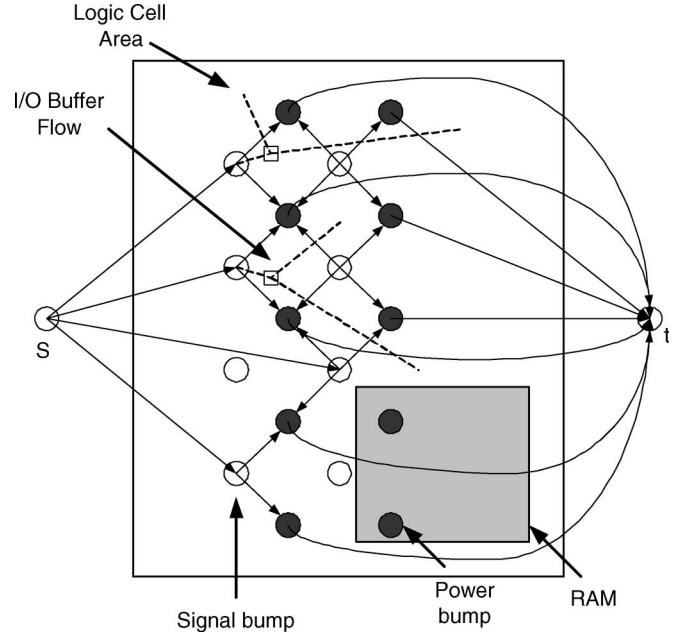


Fig. 5. Network construction for ICDCPO. Some signal bump (corresponding I/O buffer) vertices $io_i$ only connect to power bump bin vertices inside possible current drawn region for $io_i$. Note that for each power bump bin vertex, there is specified capacity indicating number of I/O that it can accommodate. Dashed lines in figure represent connection between I/O buffers, signal bumps, and other logic cells. We use corresponding wirelength (I/O wirelength) as part of cost function.

TABLE I
NUMBER OF CELLS, NETS, AND I/O TERMINALS IN SOME MCNC
STANDARD CELL PLACEMENT BENCHMARKS

| Benchmark | Cells | Nets | IOs |
|-----------|-------|------|-----|
| struct | 1952 | 1920 | 64 |
| biomed | 6514 | 7052 | 97 |
| industry1 | 3085 | 2594 | 814 |
| industry2 | 12637 | 13419 | 495 |
| industry3 | 15433 | 21940 | 374 |

the I/O wirelength for the placement, and the voltage drop threshold violation (VDTV) for the power network.

We divide the whole die into bins based on power bumps. Each bin has a certain amount of area for accommodating I/O buffers, which are obtained from the dead space or other preplanned free space in an existing placement and the building cost of I/O buffer blocks. For some bins that are occupied fully or partially by memory blocks, the area of corresponding bins will be zero or much less than a certain amount. Thus, we use $P$ to represent the set of power bump bins as well. We define $H = \{h_1, \ldots, h_n\}$ to be the set of regions that the buffer $io_i$ can possibly draw current from (shown in Fig. 4), which is similar to [18]. Each region contains a set of power bumps that the corresponding I/O buffer can consume power from.

In the next section, we introduce a cost function to minimize the I/O wirelength and total voltage drop in power network and present an algorithm to solve the proposed problem. Note that the I/O wirelength that we mention in this brief is the wirelength estimation of connecting I/O buffer, signal bump, and I/O port of corresponding logic cells. It is not the total wirelength estimation for the whole placement.

### III. I/O CLUSTERING ALGORITHM IN DESIGN COST AND PERFORMANCE OPTIMIZATION

We first construct a network with an embedded cost function and run a min-cost flow algorithm [19] to obtain the solution. The network

TABLE II
EXPERIMENTAL RESULTS OF OUR APPROACH ON MCNC BENCHMARKS SUMMARIZED IN TABLE I, COMPARED WITH INTUITIVE/GREEDY APPROACH.
WITH SLIGHT INCREASE PERCENTAGE IN VDTV, DC REDUCTION CAN BE OBTAINED WITH AVERAGELY SHORTER I/O WIRELENGTH

| Benchmark | # power bump | # signal bump | Greedy | | | I/O Clustering | | | Comparison | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Time (sec) | VDTV (%) | # i/o buf block | Time (sec) | VDTV (%) | # i/o buf block | dc imp (%) | I/O WL $\frac{cluster}{greedy}$ |
| struct | 306 | 272 | <1 | 1.96 | 60 | <1 | 1.96 | 52 | 15.4 | 0.62 |
| biomed | 342 | 306 | <1 | 1.1 | 86 | 1 | 1.1 | 78 | 10.3 | 0.99 |
| industry1 | 1116 | 1050 | 1 | 0.1 | 669 | 10 | 0.98 | 455 | 47.0 | 0.002 |
| industry2 | 676 | 625 | <1 | 0.1 | 405 | 4 | 1.1 | 264 | 53.4 | 0.98 |
| industry3 | 576 | 529 | <1 | 0.15 | 311 | 2 | 0.88 | 229 | 35.8 | 1.2 |
| Average | | | | 0.68 | | | 1.2 | | 32.4 | 0.76 |

graph $G = (V, E)$ is constructed as follows (see also Fig. 5 for illustration).

1) $V = \{s, t\} \cup \text{IO} \cup P$, where $s$ is the source vertex, and $t$ is the sink vertex. IO and $P$ are defined in the problem.
2) $E = \{(s, \text{io}_i) | \text{io}_i \in \text{IO}\} \cup \{(\text{io}_i, p_j) | \text{io}_i \in \text{IO}, \ p_j \in P \cap h_i\} \cup \{(p_j, t) | p_j \in P\}$, where $h_i$ is the corresponding possible current drawn region for $\text{io}_i$. That is, there are edges from the source $s$ to every signal bump vertex with an I/O buffer attached, and there are edges from every valid power supply bump (bumps not in RAMs) vertex to the sink $t$. There are edges from every signal bump vertex with an I/O buffer attached to every valid power supply bump vertex as well.
3) Edge capacity: $U(s, \text{io}_i) = 1, U(\text{io}_i, p_j) = 1, U(p_j, t) = $ upper bound of the number of I/O buffers that bin $p_j$ can accommodate, which is computed from the dead space or other preplanned free space in the placement.
4) Cost function: $C(\text{io}_i, p_j) = \lceil \alpha W_{ij} + \beta I_i \sum_{k=1}^{m} a_{kj}^{-1} \rceil$, where $W_{ij}$ is the I/O wirelength estimation for I/O buffer $\text{io}_i$ placed at bin $p_j$ (along with the computation with other internal logic modules or cells[3]), and $a_{kj}^{-1}$ is the element on row $k$ and column $j$ of the inverse of the system matrix $A^{-1}$. For other edges, $e \in E$ and $C(e) = 0$.

Any flow in the network can be mapped into an I/O clustering solution for a subset of given I/O buffers. If a flow $f$ exists and $|f| = n$, we can assign all I/O buffers to buffer blocks in given power bump bins. In addition, because the cost of the flow is the cost for the solution of I/O buffer placement, a minimum cost flow guarantees a solution with a minimum total cost $\alpha W + \beta D$, where $W$ is the I/O wirelength, and $D$ is the total voltage drop in the power network. The total capacities of edges going from source vertex $s$ is $n$, hence, the maximum flow $|f_{\max}| = n$. We have the following theorem to show the effectiveness and integrality property [19] of min-cost maximum flow and present the proposed algorithm for solving ICDCPO.

*Theorem:* A min-cost flow $f$ in $G$ corresponds to an I/O clustering solution to the ICDCPO problem with a minimum total cost $\alpha W + \beta D$. A min-cost maximum flow assigns all I/O buffers in IO with a minimum total cost.

Algorithm for solving ICDCPO.

1) Construct the network graph $G$.
2) Assign capacities $U$ and cost $C$.
3) Apply min-cost maximum flow algorithm on $G$.
4) Derive the corresponding I/O clustering solution.

Finding a min-cost maximum flow in a network is a classical problem for which several polynomial-time optimal algorithms are available [19], [20]. We use a capacity-scaling algorithm to solve

the network in $O((m \lg U)(m + n \lg n))$ time [19], where $n = |V|$, $m = |E|$, and $U$ is the upper bound of the edge capacity.

We have presented an approach to clustering I/O buffers. Because we estimate utilizable space for I/O buffer blocks in power bump bins, we need to move part of the existing cells around to accommodate the blocks. We can use either overlap removal in [21], which applies the bisection technique, or mixed mode placement like [22] by treating I/O buffer blocks as small macros.

## IV. EXPERIMENTAL RESULTS AND DISCUSSION

We have implemented our algorithm and run on a 650-MHz Pentium III computer. The existing cell placements based on some Microelectronics Center of North Carolina (MCNC) benchmarks (in Table I) are obtained from the placer FENG SHUI [23], with an aspect ratio of 1.0.

We have adopted the following abstract model of I/O regimes from [24] for our experiments.

1) I/O buffers must be placed exactly at pad locations, and any I/O buffer can be placed at any pad location.
2) No two I/O buffers can occupy the same location.
3) For a design with I/O buffers and a rectangular core layout region, we fix pad locations with an array of locations spaced uniformly within the core layout region.

The number of power bumps and signal bumps is scaled from IBM SA-27E area-array copper technology [5].

Our approach has been compared with a conventional rule-of-thumb design popularly used by circuit designers [11]. This approach greedily minimizes I/O wirelength and induced $IR$ drop when placing I/O buffers. To be more specific, the area-array pads are placed at fixed sites at the top layer, and each of the I/O ports is routed to the closest/nearby pad. Then, all I/O buffers can intuitively have the least signal integrity constraint violations, and the I/O wirelength should be minimized. We believe that this intuitive/greedy approach can only achieve local solutions; even the work of I/O placement is still done manually by this approach in some current designs.

Table II shows the experimental results on MCNC benchmarks summarized in Table I. The percentage of VDTV shown in the table is obtained by the number of nodes whose voltage drop exceeds the threshold normalized by the total number of nodes. From the DC comparison, we obtain much fewer I/O buffer blocks (average 31.5% reduction) with a slight increase in the percentage of VDTV in power nodes. The gain in DC reduction is due to the introduction of I/O buffer block building cost when solving this problem. The reason behind the slight increase in VDTV is that we use worst case $IR$-drop estimation, assuming all buffers draw maximum current at the same time, but the situation virtually never happens in reality. In practice, designers will waive such small violations. Note that we do not count the matrix inversion time into the runtime for both approaches because they need the same time to do a VDTV analysis.

Table II also shows the I/O wirelength comparison results, and the I/O wirelength estimation has been described in Section III (network

---

[3]We apply half-perimeter I/O net wirelength (HPWL) estimation from the center of the bin to other logic cells, not including the effect of further change in the total wirelength after I/O buffer block placement.

construction). The tradeoff coefficients $\alpha$ and $\beta$ are used based on the importance of the two objectives. Here, we adjust the coefficients so that these two terms are approximately of equal weights. In fact, these two terms are tradeoff terms and are different in different test cases. In order not to bias one side, we choose a pair of $\alpha$ and $\beta$ to balance the effects. Although we have observed a slight increase in I/O wirelength for "industry3" case containing many I/O-involved nets, we have obtained a better I/O timing performance by an averagely smaller I/O wirelength.

## V. CONCLUSION

In this paper, we have presented an I/O clustering step, considering DC and performance optimization for high-end flip-chip design. We formulate the problem as a min-cost maximum flow problem, and the experimental results are encouraging. With a slight increase in the percentage of VDTV, we can automate the I/O buffer block generation, which, in turn, will yield an averagely better timing performance and a much less DC.

## REFERENCES

[1] *Design of High-Performance Microprocessor Circuits*, A. Chandrakasan, W. Bowhill, and F. Fox, Eds. Piscataway, NJ: IEEE Press, 2001.

[2] L. Cao and J. Krusius, "A new power distribution strategy for area array bonded ICS and packages of future deep sub-micron ULSI," in *Proc. IEEE Electron. Compon. Technol. Conf.*, 1997, pp. 1138–1145.

[3] P. Sandborn, M. Abadir, and C. Murphy, "The tradeoff between peripheral and area array bonding of components in multichip modules," *IEEE Trans. Compon., Packag., Manuf. Technol. A*, vol. 17, no. 2, pp. 249–256, Jun. 1994.

[4] V. Maheshwari, J. Darnauer, J. Ramirez, and W.-M. Dai, "Design of FPGAS with area I/O for field programmable MCM," in *Proc. ACM Symp. Field Programm. Gate Arrays*, 1995, pp. 17–23.

[5] P. Buffet, J. Natonio, R. Proctor, Y. Sun, and G. Yasar, "Methodology for I/O cell placement and checking in ASIC designs using area-array power grid," in *Proc. IEEE Custom Integr. Circuits Conf.*, 2000, pp. 125–128.

[6] G. Yasar, C. Chiu, R. Proctor, and J. Libous, "I/O cell placement and electrical checking methodology for ASICs with peripheral I/Os," in *Proc. IEEE Int. Symp. Quality Electron. Des.*, 2001, pp. 71–75.

[7] R. Farbarik, X. Liu, M. Rossman, P. Parakh, T. Basso, and R. Brown, "CAD tools for area-distributed I/O pad packaging," in *Proc. IEEE Multi-Chip Module Conf.*, 1997, pp. 125–129.

[8] P. Zuchowski, J. Panner, D. Stout, J. Adams, F. Chan, P. Dunn, A. Huber, and J. Oler, "I/O impedance matching algorithm for high-performance ASICs," in *Proc. IEEE Int. ASIC Conf. Exhib.*, 1997, pp. 270–273.

[9] J. Kozhaya, S. Nassif, and F. Najm, "I/O buffer placement methodology for ASICs," in *Proc. IEEE Int. Conf. Electron. Circuits Syst.*, 2001, pp. 245–248.

[10] R. Lomax, R. Brown, M. Nanua, and T. Strong, "Area I/O flip-chip packaging to minimize interconnect length," in *Proc. IEEE Multi-Chip Module Conf.*, 1997, pp. 2–7.

[11] C. Tan, D. Bouldin, and P. Dehkordi, "Design implementation of intrinsic area array ICS," in *Proc. 17th Conf. Adv. Res. VLSI*, 1997, pp. 82–93.

[12] J. McGrath, "Chip/package co-design: The bridge between chips and systems," in *Advanced Packaging Mag.*, Jun. 2001.

[13] J. Parker, R. Sergi, D. Hawk, and M. Diberardino, (2003, Nov.). "IC-package co-design supports flip-chips," *EE Times*. [Online]. Available: http://www.eedesign.com/story/OEG20031113S0055

[14] K.-Y. Chao and D. Wong, "Signal integrity optimization on the pad assignment for high-speed VLSI design," in *Proc. IEEE Int. Conf. Comput.-Aided Des.*, 1995, pp. 720–725.

[15] S. Nassif and J. Kozhaya, "Fast power grid simulation," in *Proc. ACM/IEEE Des. Autom. Conf.*, 2000, pp. 156–161.

[16] L. Pillage, R. Rohrer, and C. Visweswariah, *Electronic and System Simulation Methods*. New York: McGraw-Hill, 1995.

[17] A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. Philadelphia, PA: SIAM, 1994.

[18] I.-M. Liu, H.-M. Chen, T.-L. Chou, A. Aziz, and D. Wong, "Integrated power supply planning and floorplanning," in *Proc. IEEE Asia South Pacific Des. Autom. Conf.*, 2001, pp. 589–594.

[19] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows—Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[20] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.

[21] W. Choi and K. Bazargan, "Incremental placement for timing optimization," in *Proc. IEEE Int. Conf. Comput.-Aided Des.*, 2003, pp. 463–466.

[22] H. Yu, X. Hong, and Y. Cai, "MMP: A novel placement algorithm for combined macro block and standard cell layout design," in *Proc. IEEE Asia South Pacific Des. Automation Conf.*, 2000, pp. 271–276.

[23] P. Madden, "Reporting of standard cell placement results," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 2, pp. 240–247, Feb. 2002.

[24] A. Caldwell, A. Kahng, S. Mantik, and I. Markov, "Implications of area-array I/O for row-based placement methodology," in *Proc. IEEE Symp. IC/Package Des. Integr.*, 1998, pp. 93–98.

# Efficient Static Compaction Techniques for Sequential Circuits Based on Reverse-Order Restoration and Test Relaxation

Aiman H. El-Maleh, S. Saqib Khursheed, and Sadiq M. Sait

*Abstract*—The authors present efficient reverse-order-restoration (ROR)-based static test compaction techniques for synchronous sequential circuits. Unlike previous ROR techniques that rely on vector-by-vector fault-simulation-based restoration of test subsequences, the authors' technique restores test sequences based on efficient test relaxation. The restored test subsequence can be either concatenated to the compacted test sequence, as in previous approaches, or merged with it. Furthermore, it allows the removal of redundant vectors from the restored subsequences using a state traversal technique and incorporates schemes for increasing the fault coverage of restored test subsequences to achieve an overall higher level of compaction. In addition, test relaxation is used to take ROR out of saturation. Experimental results demonstrate the effectiveness of the proposed techniques.

*Index Terms*—Fault coverage, linear reverse-order restoration (LROR), state traversal (ST), static compaction, test relaxation.

## I. INTRODUCTION

The complexity of sequential automatic test pattern recognition (ATPG) is significantly higher than combinational ATPG [1]. For this reason, to maximize fault coverage, sequential ATPG uses heuristics that could result in large test sequences. For example, when genetic algorithms are employed, a high fault coverage is achieved, but at the expense of long test sequences [2].

The length of a test set for testing system-on-chip (SoC) crucially affects the test application time (TAT) and memory requirements of the tester. Therefore, test compaction focuses on reducing the length of a test set while maintaining its fault coverage. Test compaction

The authors are with the Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia (e-mail: aimane@ccse.kfupm.edu.sa; saqib@ccse.kfupm.edu.sa; sadiq@ccse.kfupm.edu.sa).