

Evaluation of BEOL Design Rule Impacts Using An Optimal ILP-based Detailed Router

Kwangsoo Han[‡], Andrew B. Kahng^{†‡} and Hyein Lee[‡]

[†]CSE and [‡]ECE Departments, UC San Diego, La Jolla, CA 92093
{kwhan, abk, hyeinlee}@ucsd.edu

ABSTRACT

Continued technology scaling with more pervasive use of multi-patterning has led to complex design rules and increased difficulty of maintaining high layout densities. Intuitively, emerging constraints such as unidirectional patterning or increased via spacing will decrease achievable density of the final place-and-route solution, worsening die area and product cost. However, no methodology exists for accurate assessment of design rules' impact on physical chip implementation. At the same time, this is a crucial need for early development of BEOL process technologies, particularly with FinFET or future vertical-device architectures where cell footprints can become much smaller than in bulk planar CMOS technologies. In this work, we study impacts of patterning technology choices and associated design rules on physical implementation density, with respect to *cost-optimal* design rule-correct detailed routing. A key contribution is an Integer Linear Programming (ILP) based optimal router (*OptRouter*) which considers complex design rules that arise in sub-20nm process technologies. Using *OptRouter*, we assess wirelength and via count impacts of various design rules (implicitly, patterning technology choices) by analyzing optimal routing solutions of clips (i.e., switchbox instances) extracted from post-detailed route layouts in an advanced technology.

Categories and Subject Descriptors

B.7.2 [Hardware]: INTEGRATED CIRCUITS—*Design Aids*

General Terms

Algorithms, Design, Performance

Keywords

Design Rule Evaluation, Routability, Multiple Patterning, ILP-based Detailed Router

1. INTRODUCTION

To scale semiconductor process nodes below the resolution limits of 193i optical lithography, multi-patterning techniques (e.g., litho-etch-litho-etch (LELE) and self-aligned double and quadruple patterning (SADP, SAQP) [13] have already been widely used in production. Multi-patterning is expected to be the basis of mainstream process offerings through the foundry 10nm and even 7nm nodes, and will persist even with deployment of extreme ultraviolet (EUV) lithography [10]. Although multi-patterning techniques are key enablers for advanced sub-20nm process technologies, they can induce highly complex design rules which challenge both IC physical design tools and the development (and enablement) of IC physical implementation methodology. Tight design rules (e.g., via placement restrictions, unidirectional routing on Mx layers, etc.) lead to design wirelength and density overheads, to the point where benefits from technology scaling reduce or even disappear altogether. Assessing the real value of a prospective future technology is also difficult in FinFET nodes, where

higher drive strengths enable smaller standard-cell footprints that further challenge pin access and routability [1].

Given the above considerations, as well as the enormous cost of technology development and design enablement for a new process node, it is critical for the industry to be able to assess the impact of design rules (implicitly, patterning technology choices) on physical implementation metrics. Such assessments should be made as early as possible, to permit correct choices among various technology options and to enable design-technology co-optimization. Unfortunately, there are two basic reasons why process technology developers cannot easily evaluate impacts of complex design rules on chip implementation metrics. First, EDA vendors often require prolonged, close co-development with customers to correctly support new advanced design rules. While the latest Library Exchange Format standard (LEF5.8) [21] supports advanced design rule descriptions, even for the rapidly approaching foundry 10nm node there is varying (and contradictory) support across the EDA industry today [17, 18]. Thus, it is practically difficult to study new “future” design rules with current EDA tools. Second, EDA tools apply many heuristics to perform efficient large-scale layout optimizations. This clouds evaluations of how new patterning technologies or design rules impact chip implementation metrics. In other words, the “chicken-egg” relationship between current EDA algorithms that are optimized for current design enablements (design rules, cell libraries, etc.) makes it difficult to assess true impacts of future design enablements. Wherever possible, we would like to reduce the “chicken-egg” obstacles to design rule and patterning technology assessment.

In this work, we provide a framework for evaluating how prospective sub-20nm design rules – as well as back-end-of-line (BEOL) stack choices – will affect chip implementation metrics such as density or wirelength. Our framework is based on *optimal* detailed routing that is correct with respect to advanced design rules. We describe *OptRouter*, an ILP-based optimal detailed router which considers various design rules and technology options especially for the coming 10nm/7nm process nodes. *OptRouter* computes optimal routing solutions for small switchboxes (approximately the size of a single gcell [12], similar to the recent work of [11]), and has the ability to consider routing direction (unidirectional or bidirectional), design rules induced by advanced patterning technology (e.g., SADP), via adjacency restrictions, and pin shapes. Our studies combine realistic testcases in multiple technologies (including testcases synthesized with a prototype 7nm cell library from a leading commercial IP provider) with cost-optimal detailed routing. It is this combination that enables new, quantitative assessment of design rule impact on detailed routing metrics. The key contributions of our work are summarized as follows.

- We formulate as an integer linear program (ILP) a minimum-cost switchbox routing problem that arises in advanced technology nodes (corresponding to clips from standard-cell place-and-route instances). In contrast to previous approaches, our formulation captures multi-pin net routing (i.e., Steiner routing), via shapes, via adjacency restrictions, pin shapes, layer uni-/bi-directionality, and SADP constraints that occur with sub-20nm patterning.
- We develop *OptRouter*, which extracts layout clips from place-and-route solutions and uses ILOG CPLEX v12.5.1 [20] to solve the corresponding ILP instances. The correctness and capability of *OptRouter* are validated against commercial router results with foundry 28nm 8- and 12-track and 7nm 9-track libraries.

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

DAC '15, June 07 - 11, 2015, San Francisco, CA, USA.

Copyright 2015 ACM 978-1-4503-3520-1/15/06 ...\$15.00.

<http://dx.doi.org/10.1145/2744769.2744839>.

- We apply OptRouter within a novel methodology to quantify and rank impacts of complex sub-20nm design rules on layout metrics (wirelength, vias, and routability). Our testbed notably include a prototype 7nm PDK from a leading IP provider, as well as the aforementioned 28nm foundry libraries.
- Our comparisons of different design rules' impacts can potentially guide patterning technology choices and other basic design-technology co-optimization decisions.

In the remainder of this paper, Section 2 briefly reviews relevant previous works, with an emphasis on the seminal recent work of Jia et al. [11]. In Section 3, we explain our ILP formulation of multi-pin net routing with the consideration of various cell library and routing strategies (e.g., pin shapes and via shapes) and routing rules (e.g., via adjacency restrictions and SADP-specific rules) Section 4 describes our empirical studies: experimental questions, design of experiments, result and discussion. We offer conclusions and ongoing research directions in Section 5.

2. RELATED WORK

Relevant previous works are found in two areas: (1) design rule evaluation frameworks, and (2) ILP-based global and detailed routers.

Design rule evaluation. The work of [8] exemplifies efforts to connect layout ground rules with layout area, electrical variability, and parametric yield implications. Specifically, the authors of [8] study the effect of a line-end extension rule on logic standard cell and SRAM bitcell layout area, and on leakage variability and parametric yield. Ghaida and Gupta [6] propose DRE, a platform that comprehensively connects design rule alternatives to the automated synthesis of standard-cell library cells, and then to the power-performance-area envelope of standard-cell based layouts of small blocks. Subsequently, [7] extend the DRE approach to chip-level analyses. Badr et al. [2] suggest a pattern matching-based design rule evaluation method, which is then applied to checking of routing within standard cells. A fundamental distinction between these previous works and our present work is that we provide a new capability to assess design rule and patterning technology choices with *cost-optimal detailed routing*.

ILP-based routers. ILP has been widely used for optimization problems due to its simplicity and the ability to find optimal solutions up to some limit of tractable instance complexity. A number of works adopt ILP for global routing, often starting from a multi-commodity flow perspective. The early work of Carden and Cheng [3] uses column-generating techniques within a multi-commodity flow based global router. Cho et al. [5] propose a global router based on box expansion and progressive ILP. After decomposing nets into two-pin nets, ILP is used to choose a routing between two L-shaped candidate routings for each two-pin net within a box. The approach iteratively expands the box and solves new nets within the expanded new box, using progressive ILP and maze routing. Similarly, Hu et al. [9] use ILP for global routing; they enumerate two path candidates to connect two-pin nets after initial routing, and an ILP is formulated to select the better path between the two candidates.

An important recent work is that of Jia et al. [11], which proposes a *detailed* router based on multi-commodity flow. The authors of [11] formulate an ILP for detailed routing with all nets being two-pin nets. Pin shapes and basic design rules (side-to-side, tip-to-tip, cut-to-cut) are considered. The proposed methods are demonstrated to reduce the number of Design Rule Check (DRC) violations in a 45nm technology without wirelength or via overheads. A fundamental distinction between the work of Jia et al. [11] and our present work is that [11], while using ILP, does not guarantee optimal routing since multi-pin nets are not handled in the formulation. Further, only basic design rules are considered. In particular, the ability to compute minimum-cost *optimal* routing solutions with SADP-specific rules and via shapes is unique to our present work.

3. OPTIMAL ROUTING FORMULATION

We now describe our ILP-based formulation of the detailed routing problem for a netlist of multi-pin nets, with consideration of via adjacency restrictions, unidirectional routing, SADP-aware line end rules, pin shapes, and via

shapes. Like previous works, our development adopts the well-known paradigm of multi-commodity flow. Table 1 gives the notations that we use.

Table 1: Notations.

| Notation | Meaning |
|------------------------|---|
| N | set of multi-pin nets |
| n_k | k^{th} multi-pin net |
| s_k | source of n_k |
| T_k | set of sinks of n_k |
| $t_{k,i}$ | i^{th} sink of n_k |
| $G(V, A)$ | routing graph |
| V | set of vertices (of the routing graph) |
| v_i | a vertex with the location (x_i, y_i, z_i) |
| A | set of directed arcs |
| $a_{i,j}$ | a directed arc from v_i to v_j |
| $e_{i,j}^k$ | 0-1 indicator whether $a_{i,j}$ is used in the routing of n_k |
| $c_{i,j}^k$ | cost for $a_{i,j}$ in the routing of n_k |
| $f_{i,j}^k$ | flow variable for $a_{i,j}$ in the routing of n_k |
| $p_{r,i}^k(p_{l,i}^k)$ | 0-1 indicator whether there are the flows connected to v_i coming from right (left) side, in the routing of n_k |

3.1 General Routing Problem Formulation

We use a *routing graph* $G = (V, A)$ to represent available routing resources, e.g., metal tracks on multiple layers, and inter-layer vias. Each vertex $v_i \in V$ is associated with variables that represent coordinates in the three-dimensional routing resources: horizontal metal track x_i , vertical metal track y_i and metal layer z_i . A directed arc $a_{i,j}$, where $x_j = x_i$, $y_j = y_i$ and $z_j = z_i \pm 1$, represents a via. We solve the optimization:

$$\text{Minimize: } \sum_{n_k \in N} \sum_{a_{i,j} \in A} c_{i,j}^k e_{i,j}^k$$

Subject to:

$$\sum_{n_k \in N} (e_{i,j}^k + e_{j,i}^k) \leq 1 \quad a_{i,j}, a_{j,i} \in A \quad (1)$$

$$e_{i,j}^k \geq \frac{f_{i,j}^k}{|T_k|} \quad a_{i,j} \in A, n_k \in N \quad (2)$$

$$e_{i,j}^k \leq f_{i,j}^k \quad a_{i,j} \in A, n_k \in N \quad (3)$$

$$\sum_{v_j: a_{i,j} \in A} f_{i,j}^k - \sum_{v_j: a_{j,i} \in A} f_{j,i}^k = \begin{cases} |T_k| & \text{if } v_i = s_k, n_k \in N \\ -1 & \text{else if } v_i \in T_k, n_k \in N \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The objective is to minimize the weighted sum of $e_{i,j}^k$, i.e., weighted total wirelength and the number of vias. Constraint (1) ensures that each arc is used by only one net. Constraints (2) and (3) pertain to the binary variable $e_{i,j}^k$, which indicates whether there is a flow through $e_{i,j}^k$. Constraint (4) ensures source-sink connectivities (flow conservation). The first and second terms respectively represent the sum of the flows exiting v_i (outflows of v_i) and the sum of the flows entering v_i (inflows of v_i). For any internal node that is not a source or a sink, the sum of the node's outflows must equal to the sum of the node's inflows. For a source s_k , the sum of outflows of s_k must be $|T_k|$ (the number of sinks) since there must be $|T_k|$ flows which connect between s_k and $|T_k|$ number of sinks in n_k , and the sum of inflows of s_k must be zero. On the other hand, for a sink $v_i \in T_k$, the sum of inflows must be one since a flow coming from s_k must reach each sink, and the sum of outflows must be zero.

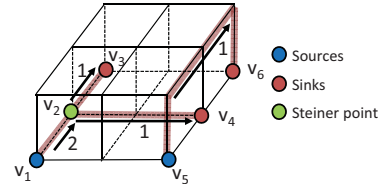


Figure 1: Example showing multi-pin nets and the routing solution.

Figure 1 shows a two-net example consisting of a three-pin net (n_1) and a two-pin net (n_2), along with its solution. Net n_1 has a source v_1 and two sinks, v_3 and v_4 . Net n_2 has a source v_5 and a sink, v_6 . According to Constraint (4), for n_1 , the sum of outflows of the source node $v_1 = 2$ ($|T_1|$) and the sum of inflows of sink nodes v_3 and $v_4 = -1$. Similarly, for n_2 , the sum of outflows of $v_5 = 1$ and the sum of inflows of $v_6 = 1$. For all other vertices, the sum of outflows is equal to the sum of

inflows so that flows are conserved for each net. According to Constraint (1), $e_{1,2}^1, e_{2,3}^1 = 1$, which connects between v_1 and v_3 , since $f_{1,2}^1, f_{2,3}^1$ are non-zero value. Constraint (1) forces $e_{1,2}^2, e_{2,1}^2, e_{2,1}^1$ to be zero so that the edge between v_1 and v_2 can be reserved only for n_1 .

3.2 Routing Rule Formulation

Via restrictions. As noted in [13], placement of vias next to each other is not allowed in advanced nodes. That is, as via pitches are larger (e.g., by a $\sqrt{2}$ factor) than metal pitches, placement of a via at a particular location blocks horizontally and vertically adjacent locations, and sometimes diagonally adjacent locations as well.

We use the following constraint so that any neighbor vertical arcs $a_{i',j'}$ of a vertical arc $a_{i,j}$ can be blocked if there is a via between v_i and $v_{j'}$, where $x_{i'} = x_j \pm 1, y_{i'} = y_{j'} = y_i \pm 1, z_{i'} = z_i$ and $z_{j'} = z_j$.

$$e_{i,j}^k + e_{j,i}^k + e_{i',j'}^k + e_{j',i'}^k \leq 1 \quad \forall a_{i',j'}$$

In our study below, we consider two types of restrictions: (i) blocking of orthogonally adjacent locations (N, E, S, W neighbors), and (ii) blocking of both orthogonally and diagonally (NE, NW, SE, SW) adjacent locations.

Unidirectional routing. Patterning with severe restriction, as with one-pitch/one-orientation metal layers, is used because of better robustness, scalability and manufacturability – as well as fewer masking steps – compared to a standard LELE-patterned bidirectional metal layer. We trivially restrict routing on a given layer to be unidirectional by removing arcs that are not in the preferred direction. (See also discussion of SADP constraints, below.)

Pin shape. In the above example of Figure 1, we assume that each source or sink has a single fixed location. However, in actual routing, a pin has multiple access points, which means that the source or sink locations ultimately used in the routing solution can vary. Multiple access points for source and sink are captured by creating a *supersource* or *supersink* which is connected to all available access points in the corresponding pin. We note that the supersource and supersink are virtual vertices, which are not actually located but which nonetheless have flows. We also observe that each access point for source or sink becomes an internal node.¹

Via shape. To trade off between manufacturability and routability, various via types with square or rectangular shapes may be instantiated. Some vias shapes, e.g., 2×2 size, are too large to be modeled as a single vertex in our routing graph. We model a via's shape by creating a *representative vertex* which is connected to all the vertices that belong to a via, according to that via's footprints on lower and upper layers.²

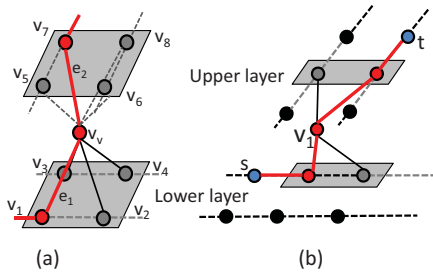


Figure 2: Via shape. (a) 2×2 square via. (b) 2×1 bar via.

Figure 2(a) shows a via and its vertices in a routing graph. v_v is a square via with size 2×2 (with respect to the number of metal tracks). With the flow conservation constraint

¹Pin shape is important in assessment of routing costs, e.g., smaller pin geometries with fewer access points in advanced FinFET nodes are a major challenge to detailed routing. In 7.5T or 7.25T library cells in FinFET nodes, power/ground rails, fin connections and other aspects of standard cell architectures must reconcile with pin shapes (access points). Strict tip-to-tip spacing (more than one contacted poly pitch (pin pitch)), diagonal via placement restriction as discussed above, and wider power rails also decrease the number of access points to a cell and potentially cause unroutability [14]. We study interactions of smaller pin shapes in 7nm (Figure 9(c)) and routing rules in Section 4.1.

²Doubled or redundant vias are also modelable with small modification of via shape formulation.

(Constraint (3)), once a flow (routing) enters v_v , the flow goes through one of four vertices in the upper layer. Note that for each via type, vertices are created for all possible locations where the via can be placed. For example, if a 2×2 size square via type is added to the routing graph with three layers and 15×15 tracks ($15 \times 15 \times 3$), we will create 392 ($14 \times 14 \times 2 = (15 - 1) \times (15 - 1) \times (3 - 1)$) vertices for the squared via at all possible locations. Further, note that we use lower cost values for larger via shapes so that the optimization selects as many larger vias as possible to achieve better manufacturability.

In addition to the basic formulation with Constraints (1)–(3), vertices used by a via must be blocked and not be used by other nets. For example, in Figure 2(a), as v_7 is selected by e_2 , all the gray edges connected to other vertices used by the via (v_5, v_6, v_8) must be disabled for other nets. A generalized formulation is given in Constraint (5) where i' are the vertices that are not used for the routing and within the used via shape, and j' are the neighbor vertices of i' :

$$\sum_{n_k \in N} (e_{v,i}^k + e_{i,v}^k) + \sum_{n_{k'} \in N, v_{i'}, v_{j'}: a_{i',v}, a_{j',i'} \in A} (e_{i',j'}^{k'} + e_{j',i'}^{k'}) \leq 1$$

where $a_{i,v} \in A, k' \neq k, i' \neq i$ (5)

Thus, Constraint (5) prevents any other nets from using the vertices i' or the edges connected to i' . Figure 2(b) shows an example of 2×1 size bar via shape. Vertices s and t are source and sink, respectively. The red lines are selected as routing from s to t . The gray dots in Figure 2(b) are disabled for other nets by Constraint (5), so that there is no overlap between the bar via and other nets.

SADP-aware rules. Xu et al. [16] propose SADP-specific design rules. As shown in the Figure 3(a), the end of line (EOL) of a wire segment is the key parameter to check the rules.

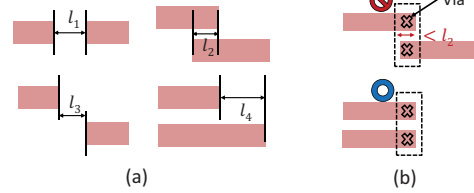


Figure 3: (a) SADP-specific design rules; (b) example showing that via location does not provide enough information to distinguish the upper and lower cases, i.e., to check SADP-aware rules.

In our ILP formulation, we use via locations to check the locations of EOL, but this is not enough to differentiate the two cases in Figure 3(b), where the upper case is an illegal routing while the lower case is legal with the same via placements. Therefore, binary variables $p_{r,i}^k, p_{l,i}^k$ that indicate whether there are flows connected to a vertex v_i come from right or left direction, respectively, are defined for a net n_k to represent the directions of EOL. Note that there are only two directions, since we assume unidirectional routing.

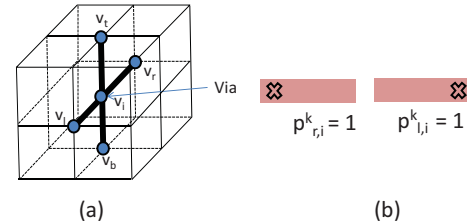


Figure 4: (a) An example of a routing graph. (a) The p variable of a vertex v_i is determined by flow variables of edges with vertex v_i 's neighbor vertices v_t, v_b, v_l, v_r ; (b) wire segment geometries that respectively result when $p_{r,i}^k = 1$ and $p_{l,i}^k = 1$.

Figure 4(a) shows a vertex v_i and its top, bottom, left, right neighbor vertices (v_t, v_b, v_l, v_r) in a routing graph, and Figure 4(b) enumerates the cases when each p variable is one. For the left EOL at (x_i, y_i, z_i) in Figure 4(a), where $p_{r,i}^k = 1$, the right edge ($e_{r,i}^k$) connected to v_i must be used in routing and the left edge ($e_{l,i}^k$) connected to v_i must not be used. By Constraint (4), the expression $e_{r,i}^k \&\& \neg e_{l,i}^k$ is equal to the

right-hand side of Constraint (7). The right EOL ($p_{l,i}^k = 1$) is formulated in the same manner as Constraint (6).

$$p_{l,i}^k = (e_{l,i}^k * e_{l,t}^k) \vee (e_{l,i}^k * e_{i,b}^k) \vee (e_{l,i}^k * e_{i,l}^k) \vee (e_{l,i}^k * e_{b,i}^k) \quad (6)$$

$$p_{r,i}^k = (e_{r,i}^k * e_{i,t}^k) \vee (e_{r,i}^k * e_{i,b}^k) \vee (e_{i,r}^k * e_{l,i}^k) \vee (e_{i,r}^k * e_{b,i}^k) \quad (7)$$

As all the variables are binary, we can convert the quadratic constraints in Constraint (6) and (7) to linear constraints by using a simple technique as shown in (8). ($a \leq b$) && ($a \leq c$) condition ensures that a is zero when either b or c is zero. The condition ($a \geq b + c - 1$) makes $a = 1$ when both b and c are one.

$$a = b * c \iff (a \leq b) \&\& (a \leq c) \&\& (a \geq b + c - 1) \quad (8)$$

We then convert the Constraint (6) to a set of linear constraints as shown in Constraint (9).

$$\begin{aligned} p_{l,i}^k &\geq p_{l,i,1}^k ; p_{l,i}^k \geq p_{l,i,2}^k ; p_{l,i}^k \geq p_{l,i,3}^k ; p_{l,i}^k \geq p_{l,i,4}^k \\ p_{l,i}^k &\leq p_{l,i,1}^k + p_{l,i,2}^k + p_{l,i,3}^k + p_{l,i,4}^k \\ (p_{l,i,1}^k \leq e_{l,i}^k) \&\& (p_{l,i,1}^k \leq e_{i,t}^k) \&\& (p_{l,i,1}^k \geq e_{l,i}^k + e_{i,t}^k - 1) \\ (p_{l,i,2}^k \leq e_{l,i}^k) \&\& (p_{l,i,2}^k \leq e_{i,b}^k) \&\& (p_{l,i,2}^k \geq e_{l,i}^k + e_{i,b}^k - 1) \\ (p_{l,i,3}^k \leq e_{i,l}^k) \&\& (p_{l,i,3}^k \leq e_{t,i}^k) \&\& (p_{l,i,3}^k \geq e_{i,l}^k + e_{t,i}^k - 1) \\ (p_{l,i,4}^k \leq e_{i,l}^k) \&\& (p_{l,i,4}^k \leq e_{b,i}^k) \&\& (p_{l,i,4}^k \geq e_{i,l}^k + e_{b,i}^k - 1) \end{aligned} \quad (9)$$

Here, $p_{*,*}^k$ is a net-specific variable. As SADP rules must be checked over all nets, we define global p variables as follows:

$$p_{l,i} = \sum_{n_k \in K} p_{l,i}^{n_k}, \quad p_{r,i} = \sum_{n_k \in K} p_{r,i}^{n_k} \quad (10)$$

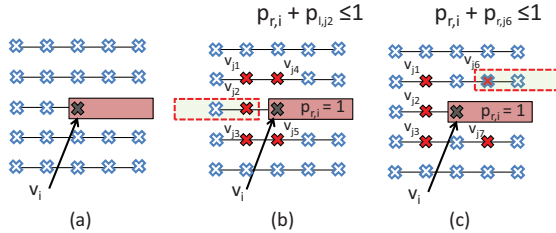


Figure 5: (a) A wire segment, of which the EOL is located at vertex v_i with the wire coming from the right side. (b) Forbidden via locations for other wire segments with $p_{l,i} = 1$. (c) Forbidden via locations for other wire segments with $p_{r,i} = 1$.

Figure 5 shows how the p variables can be used to formulate SADP-aware rules for ILP. Figure 5(a) shows a wire segment, of which the EOL is located at vertex v_i with the wire coming from the right side; (b) and (c) show forbidden via locations for the other wire segments. The constraints shown in Figures 5(b) and (c) are formulated as Constraint (11) and (12), respectively.

$$(p_{r,i} + p_{l,j_1} \leq 1) \&\& (p_{r,i} + p_{l,j_2} \leq 1) \&\& (p_{r,i} + p_{l,j_3} \leq 1) \&\& (p_{r,i} + p_{l,j_4} \leq 1) \&\& (p_{r,i} + p_{l,j_5} \leq 1) \quad (11)$$

$$(p_{r,i} + p_{r,j_1} \leq 1) \&\& (p_{r,i} + p_{r,j_2} \leq 1) \&\& (p_{r,i} + p_{r,j_3} \leq 1) \&\& (p_{r,i} + p_{r,j_6} \leq 1) \&\& (p_{r,i} + p_{r,j_7} \leq 1) \quad (12)$$

4. EMPIRICAL STUDIES

Our empirical studies seek to answer two basic questions:

- What are the design costs of various BEOL rules with respect to wirelength, # vias, and routability metrics?
- How much do impacts of design rules vary across different technologies and different-track cell architectures?

Overall flow of BEOL rule evaluation. We implement our experimental testbed as C++ code, with interface to support LEF/DEF [21] implemented via the *OpenAccess 2.6* [23] API. We use *CPLEX 12.5.1* [20] as our ILP solver. Figure 6 shows our overall BEOL rule evaluation flow. From a routed design, all possible routing clips are extracted, and evaluated according to our pin cost metric. The clips with highest pin cost are selected, and each clip (= switchbox instance) is converted to a routing graph based on available metal tracks, and then to a corresponding ILP instance, for each routing rule configuration

that we study. Our *OptRouter* then obtains optimal routing solutions by solving the ILP. From the solution, we report out wirelength, number of vias, and feasibility (routability) for each given clip with each given rule configuration. For the experiments that we report here, routing cost in the ILP is defined as *wirelength* + $4 \times$ *number of vias*. We have separately observed that the ILP sensibly handles alternative routing cost definitions with different weighting of via count.

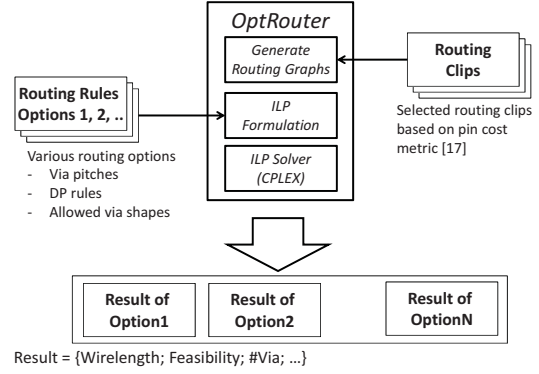


Figure 6: Overall flow of BEOL rule evaluation.

Physical implementation with advanced technology.

We verify our methods using the open-source AES design [22] and an ARM Cortex M0, implemented with three different technologies and standard-cell libraries: 8-track in 28nm FDSOI (N28-8T), 12-track in 28nm FDSOI (N28-12T), and 9-track in 7nm (N7-9T). We use *Synopsys Design Compiler H-2013.03-SP3* [24] for synthesis and *Cadence Encounter Digital Implementation System XL 13.1* [19] for P&R. We implement each design multiple times, with a range of final utilizations. Table 2 summarizes benchmark design information.

Table 2: Benchmark designs.

| Tech. | Design | Period (ns) | #inst. | Util. (%) |
|---------|--------|-------------|-----------|-----------|
| N28-12T | AES | 1.2 | 13.5–14K | 89–94 |
| | M0 | 2.2 | 9.2K | 90–96 |
| N28-8T | AES | 2 | 12–12.7K | 89–95 |
| | M0 | 2.5 | 9.3–9.5K | 90–95 |
| N7-9T | AES | 0.6 | 13–15K | 93–97 |
| | M0 | 1.2 | 9.7–11.4K | 92–95 |

For 7nm technology, we use 7nm standard-cell libraries (P&R, layout and timing views) from a leading IP provider; metal pitches on layers M1 to M6 and layers M7 to M8 are 40nm and 80nm, respectively. In this technology, our design enablement is missing detailed BEOL technology information such as RC values and BEOL stack options. Thus, to obtain timing-closed P&R results we scale up the geometries of the 7nm 9-track cells by $2.5 \times$ in the vertical dimension (i.e., by the ratio of $1 \times$ metal pitch in 28nm horizontal layers (100nm) to $1 \times$ metal pitch in 7nm horizontal layers (40nm)). Then, the scaled 7nm cells fit into the 28nm BEOL stack with the same number of horizontal metal tracks, for which we use 100nm metal pitch in horizontal layers. To scale the widths of the 7nm standard cells, we scale by the ratio of the 28nm placement grid (vertical metal layer pitch of 136nm) to that of the 7nm placement grid (vertical metal layer pitch of 54nm), which is ~ 2.5 . We further adjust pin locations so that pins are on-grid, since simple scaling results in off-grid pins which affect routability.³ To derive the missing 7nm wire RC information from 28nm RC values, we scale up R by $15 \times$ for 7nm wire R, and use the same wire C value. This follows methodology of, e.g., [4] to account for the rapid increase of resistivity in advanced nodes. Then, since we are using the scaled geometries to mimic a 7nm P&R flow, R and C per unit length are scaled

³In gorier detail: the 28nm and 7nm placement grids are 136nm and 54nm, respectively, with ratio between the two being ~ 2.519 . It is not possible to obtain integer cell widths by simply scaling with this number. Thus, we scale up the 7nm cells by 2.5 so that all cell widths are a multiple of 135nm. We then increase each cell width by *scaled cell width* / 135 in order to make it a multiple of 136nm, which is the foundry 28nm placement grid. Since scaling by $2.5 \times$ results in a pin pitch of 135nm, which is off-grid with respect to a 136nm grid, we perform a scripted movement of pin locations so that all pins are again on-grid (the pin x locations should be multiples of 136nm).

down (in the P&R tool) by $2.5\times$. The end results is that 7nm R and C per unit length (R_{N7} , C_{N7}) are obtained from 28nm R and C per unit length (R_{N28} , C_{N28}) as $R_{N7} = 6 \times R_{N28}$ and $C_{N7} = C_{N28}/2.5$.

Extraction of routing clips. We use $1\mu\text{m} \times 1\mu\text{m}$ routing clips extracted from the routed designs as input instances; these correspond to 7 vertical routing tracks \times 10 horizontal routing tracks, with eight metal layers, for *OptRouter*. Figure 7 shows example routing clips extracted from (a) N28-12T cells, (b) N28-8T cells and (c) N7-9T cells.⁴

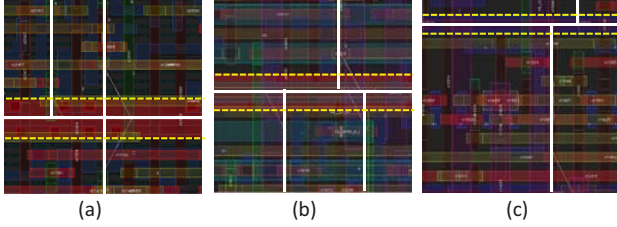


Figure 7: Routing clips from (a) N28-12T, (b) N28-8T and (c) N7-9T. Standard cell boundaries and power/ground rail are highlighted with white lines and yellow dashed lines, respectively.

We select “difficult-to-route” clips based on pin cost metrics of Taghavi et al. [15], specifically, a pin existence cost (PEC), a pin-area cost ($PAC = \sum_{i=1}^{PEC} 2^{2-\frac{area(p_i)}{\theta}}$) and a pin-spacing cost ($PRC = \sum_{i=1}^{PEC-1} \sum_{j=i+1}^{PEC} 2^{2-\frac{spacing(p_i, p_j)}{3\theta}}$). We use $PEC + PAC + PRC$ as the *pin cost* for a routing clip, with $\theta = 500$ to obtain a reasonable range of costs.

We calculate the pin cost for every routing clip in the routed testcases listed in Table 2 ($\sim 10K$ clips per testcase). Figure 8 shows the top-100 pin cost ranges for several versions of AES and M0 design implementations in N7-9T with different utilizations. The utilizations of AES_v1, AES_v2 and AES_v3 are 93%, 95% and 97% respectively, and the utilizations of M0_v1, M0_v2 and M0_v3 are 92%, 94% and 95%⁵. We observe that pin cost distributions do not change significantly with different utilizations, and that pin cost distributions are not design-specific: ranges of top-100 pin costs of both designs are similar (AES: 33~42, M0: 30~41). Thus, in each technology we select top-100 clips from across all design implementation, according to the pin cost metric.

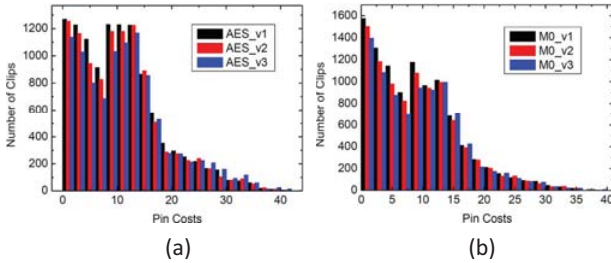


Figure 8: Pin cost distributions (per the $PEC + PAC + PRC$ metrics in [15]) of (a) AES and (b) M0 with different utilizations.

4.1 Design of Experiments

We evaluate various BEOL design rule configurations, each of which is a combination of via restrictions and mix of LELE/SADP BEOL layers. (All routing layers are unidirectional in our study.) Table 3 shows BEOL design rule configurations used in the experiments. We test three via restriction cases (*0 neighbors blocked*; *4 neighbors blocked*; and *8 neighbors blocked*) and five LELE/SADP layer combinations (M2-M8 LELE layers (*No SADP*); M2-M8 SADP layers ($SADP \geq M2$); M2 LELE + M3-M8 SADP layers ($SADP \geq M3$); M2-M3 LELE + M4-M8 SADP layers ($SADP \geq M4$); and M2-M4 LELE + M5-M8 SADP ($SADP \geq M5$)). Via restrictions are applied to the V12 through V78 layers.

⁴By comparison, the recent work of [11] uses $1.26\mu\text{m} \times 1.26\mu\text{m}$ clips in a 45nm technology; these correspond to 9 vertical routing tracks \times 9 horizontal routing tracks.

⁵We use high utilizations to obtain designs that are “difficult-to-route” and sensitive to design rules due to routing congestion.

We select the top 100 routing clips according to pin costs across all designs in Table 2, for each of the three combinations of technology node and cell height, as discussed above. We then run *OptRouter* on each of the 100 routing clips to evaluate the impact of each given routing rule configuration. We obtain the Δcost of each rule configuration, relative to the routing cost of RULE1 (no constraints).⁶ In the present study, we do not use M1 as a routing resource.

Table 3: BEOL design rule configurations.

| Name | SADP rules | Blocked via sites |
|----------------|--------------------------------|---------------------|
| RULE1 | No SADP | 0 neighbors blocked |
| RULE2, 3, 4, 5 | $SADP \geq \{M2, M3, M4, M5\}$ | |
| RULE6 | No SADP | 4 neighbors blocked |
| RULE7, 8 | $SADP \geq \{M2, M3\}$ | |
| RULE9 | No SADP | 8 neighbors blocked |
| RULE10, 11 | $SADP \geq \{M2, M3\}$ | |

We have evaluated all of RULE1 to RULE11 for the N28-12T and N28-8T technologies. However, we do not test RULE2, 7, 9, 10 and 11 for N7-9T since the smaller pin shapes in the 7nm standard cells do not permit the diagonal (adjacency in) via placement which is required for these rules. Figures 9(a), (b) and (c) show pin shapes in NAND2X1 in N28-12T, N28-8T and N7-9T, respectively. As shown in Figure 9(c), input pin shapes have only two access points and the two pins are close to each other. With eight via sites blocked, there is no way to connect two input pins without violations.

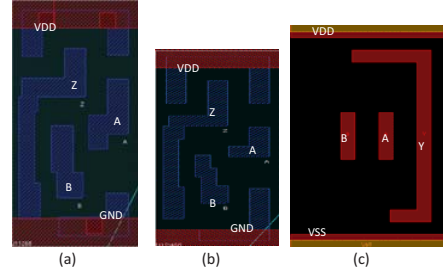


Figure 9: Pin shapes in NAND2X1: (a) N28-12T, (b) N28-8T, and (c) scaled N7-9T.

4.2 Experimental Results and Discussion

Figures 10(a), (b) and (c) respectively show sorted Δcost per clip of each RULE, in N28-12T, N28-8T and N7-9T cell-based designs. The Δ is relative to costs with RULE1 (i.e., the minimum achievable routing cost with eight unidirectional LELE layers and no via restrictions). For unroutable clips, we arbitrarily set $\Delta\text{cost}=500$ for convenience of plot generation.

In N28-12T (Figure 10(a)), we observe that SADP rules for upper metal layers above M3 do not significantly affect routing costs. Two kinds of via restrictions (4 or 8 neighbors blocked) show similar routing costs, suggesting that the orthogonal via restriction (4 neighbors blocked) is dominant. When SADP rules are applied (RULE4, 7, 2), routing costs vary across routing clips. By comparing the “crossing” traces for RULE2 and RULE6, we see that routing costs are higher with SADP layers than with via restriction rules, but that in terms of absolute feasibility, the via restriction appears to result in fewer feasible routings.

With N28-8T (Figure 10(b)), in contrast to the N28-12T case, there is higher sensitivity of Δcost to the number of SADP layers: we see a clear increasing cost trend across RULE2, 3, 4, 5. With respect to via restriction, for the 8 LELE layer (no SADP) cases, having 4 and 8 neighbors blocked yields different pin cost distributions (RULE6 vs. 9), i.e., the orthogonal via restriction is less dominant in this particular context. However, when via restriction is combined with SADP layers, the two forms of via restriction again show similar results (RULE7 vs. 10, RULE8 vs. 11).

With N7-9T (Figure 10(c)), SADP routing rules have less cost impact on layers above M4, as RULE4, RULE5 and

⁶Separate studies support the claimed optimality of *OptRouter*: we have compared the results of *OptRouter* and those of the commercial routing tool, and have found that *OptRouter* always achieves non-positive Δcost with respect to the commercial tool’s solution. Indeed, the average Δcost of $-10\sim-15$, relative to an average routing cost of ~ 380 , suggests the potential for using *OptRouter* for detailed routing improvement.

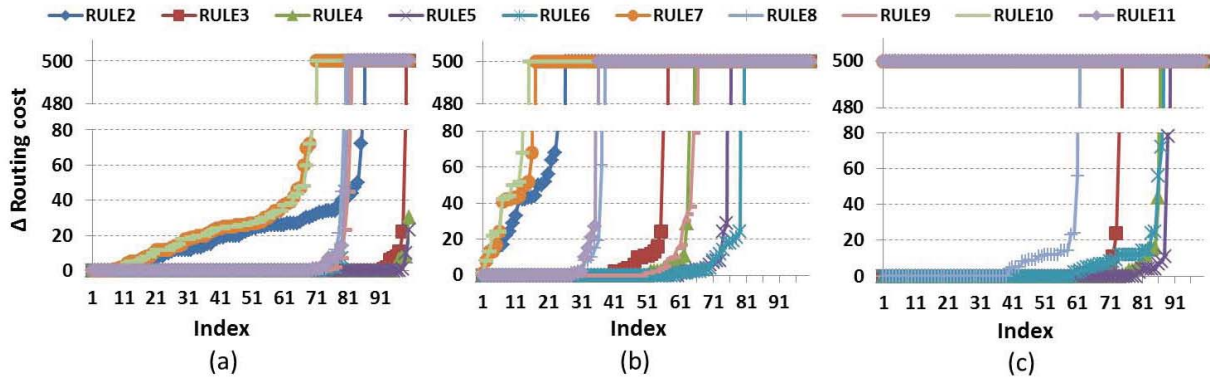


Figure 10: Δcost with different RULE* in (a) N28-12T, (b) N28-8T and (c) N7-9T.

RULE6 show similar Δcost distributions. When M3 is made into an SADP layer (RULE3), the vertical line (i.e., the clip index at which sorted Δcost goes to infinity (infeasible solution)) shifts left significantly. When the 4-neighbors via restriction is added to RULE3 (i.e., in RULE8), the vertical line shifts again. (RULE3, 4, 5, 6 and 8 respectively have 26, 14, 11, 13 and 39 infeasible clips out of 100.)

From the preceding discussions, we may tentatively form two general observations. (1) First, the via restriction and SADP routing rules show different trends, i.e., effects on the Δcost profile. Moreover, the sensitivities of Δcost to design rules and routing options vary with technology. For example, when SADP rules are applied to upper metal layers in N28-12T or N7-9T, the routing costs do not change significantly, which we interpret to mean that SADP rules do not affect routability significantly for these clips. This is different from what we observe in N28-8T. (2) Second, for design rules that are applied to upper metal layers ($>M3$), almost half of routing clips show zero Δcost . This could imply that the pin cost metric of [15] cannot, by itself, accurately quantify the difficulty. In other words, there is a gap between pin accessibility metrics such as [15] and our switchbox-centric evaluation of routability.

Analysis of the number of variables and constraints. The number of directed arcs ($|A|$), the number of vertices ($|V|$) and the number of nets ($|N|$) determine the number of variables and constraints in the ILP. Without via restrictions and SADP rules (no restriction), the number of variables is $O(|A| \cdot |N|)$. With Constraints (1)–(4), the number of constraints is $O((|V| + 3 \cdot |A|) \cdot |N|)$. Regarding via restriction, when α neighbor sites are blocked, the number of variable is the same as the basic case (no restriction), and the number of constraints is $O(\alpha \cdot |V| + (|V| + 3 \cdot |A|) \cdot |N|)$. With the SADP routing rules, the number of variables is $O((10 \cdot |V| + |A|) \cdot |N|)$ because of the additional binary indicator (p); the number of constraints is $O((34 \cdot |V| + 3 \cdot |A|) \cdot |N| + 10 \cdot |V|)$. Regarding via shapes, when a β size of via shape is considered, the number of variables is $O((\beta \cdot |V| + |A|) \cdot |N|)$ due to the creation of additional via edges, and the number of constraints is $O(\beta^2 \cdot |V| \cdot |N| + (\beta \cdot |V| + 3 \cdot |A|) \cdot |N|)$.

5. CONCLUSIONS AND FUTURE WORK

In this work, we have studied impacts of patterning technology choices and design rules on physical implementation metrics, with respect to *cost-optimal* design rule-correct detailed routing. We describe *OptRouter*, an ILP-based optimal detailed router that correctly handles multi-pin nets and various sub-20nm routing challenges including via restrictions, via shapes, and SADP patterning rules. *OptRouter* enables design rule evaluation using “difficult” routing clips (switchboxes) selected according to a pin cost metric. We study Δcost distributions for different design rules, relative to a RULE1 where all layers are LELE and there are no via restrictions. From the results, we observe that the sensitivities of Δcost to design rules and routing options vary with technology. Also, we observe that there is a gap between pin accessibility metrics such as [15] and our switchbox-centric evaluation of routability.

Future work includes speedup of *OptRouter* to gain insights into physical implementation impacts at larger granularity (switchbox size). Currently, *OptRouter* average runtime for a 7 track \times 10 track switchbox ($1.0 \times 1.0 \mu\text{m}^2$ layout area

in 28nm) is 1047 seconds (single-threaded) with SADP and via restriction rules. Without such rules (as in [11]), average runtime is 842 seconds.⁷ As noted above, our results give insight into the degree of suboptimality in current routing tools, and open up the possibility of (massively distributed) local improvement of detailed routing solutions. Also, for better quantification of “difficult-to-route” clips, development of a metric beyond [15] to estimate routability in sub-20nm nodes will be an important aspect of our future work.

6. REFERENCES

- [1] R. Aitken, G. Yeric, B. Cline et al., “Physical Design and FinFETs”, *Proc. ISPD*, 2014, pp. 65–68.
- [2] Y. Badr, K.-W. Ma and P. Gupta, “Layout Pattern-Driven Design Rule Evaluation”, *Proc. SPIE, Design-Process-Technology Co-optimization for Manufacturability VIII*, 2014.
- [3] R. Carden and C.K. Cheng, “A Global Router with a Theoretical Bound on the Optimal Solution”, *IEEE Trans. on CAD* 15(2) (1996), pp. 208–216.
- [4] T. Chan, A. Kahng, J. Li, “Toward Quantifying the IC Design Value of Interconnect Technology Improvements”, *Proc. SLIP*, 2013.
- [5] M. Cho and D. Z. Pan, “BoxRouter: A New Global Router Based on Box Expansion and Progressive ILP”, *IEEE Trans. on CAD* 26(12) (2007), pp. 2130–2143.
- [6] R. S. Ghaida and P. Gupta, “DRE: A Framework for Early Co-Evaluation of Design Rules, Technology Choices, and Layout Methodologies”, *IEEE Trans. on CAD* 31(9) (2012), pp. 1379–1392.
- [7] R. S. Ghaida, Y. Badr, M. Gupta, N. Jin and P. Gupta, “Comprehensive Die-Level Assessment of Design Rules and Layouts”, *Proc. ASP-DAC*, 2014, pp. 61–66.
- [8] P. Gupta, K. Jeong, A. B. Kahng and C.-H. Park, “Electrical Assessment of Lithographic Gate Line-End Patterning”, *SPIE J. Microlithography, Microfabrication and Microsystems* 9(2) (2010), pp. 023014-1–023014-19.
- [9] J. Hu, J. A. Roy and I. L. Markov, “Sidewinder: A Scalable ILP-based Router”, *Proc. SLIP*, 2008, pp. 73–80.
- [10] *International Technology Roadmap for Semiconductors*, Lithography Chapter, 2013. <http://www.itrs.net/>
- [11] X. Jia, Y. Cai, Q. Zhou, G. Chen, Z. Li and Z. Li, “MCFRoute: A Detailed Router Based on Multi-Commodity Flow Method”, *Proc. ICCAD*, 2014, pp. 397–404.
- [12] A. B. Kahng, J. Lienig, I. L. Markov and J. Hu, *VLSI Physical Design: From Graph Partitioning to Timing Closure*, Springer, 2011.
- [13] L. Liebmann, V. Gerousis, P. Gutwin et al., “Demonstrating Production Quality Multiple Exposure Patterning Aware Routing for the 10NM Node”, *Proc. SPIE DPTCOM VIII*, 2014.
- [14] L. Liebmann and D. Pietromonaco, “The Increasing Pain of Scaling with 193i: Where Does it Hurt? How Much More Can We Endure?”, *tutorial, SPIE Microlithography*, 2013.
- [15] T. Taghavi, C. Alpert, A. Huber, Z. Li, G.-J. Nam, S. Ramji, “New Placement Prediction and Mitigation Techniques for Local Routing Congestion”, *Proc. ICCAD*, 2010, pp.621–624.
- [16] X. Xu, B. Cline, G. Yeric, B. Yu and D. Z. Pan, “Self-Aligned Double Patterning Aware Pin Access and Standard Cell Layout Co-Optimization”, *Proc. ISPD*, 2014, pp. 101–108.
- [17] Samsung Electronics Corp. CAE principal engineer, *personal communication*, September 2014.
- [18] Qualcomm, Inc. VLSI technology principal engineer, *personal communication*, July 2014.
- [19] Cadence SOC Encounter User Guide, <http://www.cadence.com>
- [20] IBM ILOG CPLEX. www.ibm.com/products/cplex/
- [21] LEF DEF reference. <http://www.si2.org/openeda.si2.org/projects/lefdef>
- [22] OpenCores: Open Source IP-Cores, <http://www.opencores.org>
- [23] Si2 OpenAccess. <http://www.si2.org/?page=69>
- [24] Synopsys Design Compiler User Guide, <http://www.synopsys.com>

⁷OptRouter runtime for a 10 track \times 10 track switchbox, with (resp. without) SADP and via restriction rules, is 1340 (resp. 925) seconds.