Efficient Layout Generation and Evaluation of Vertical Channel Devices

Wei-Che Wang and Puneet Gupta
weichewang@ucla.edu, puneet@ee.ucla.edu
Department of Electrical Engineering, University of California, Los Angeles

***Abstract*—Vertical gate-all-around (VGAA) structure has been shown to be one of the most promising devices for the scaling beyond 10nm for its reduced area, large driving current, and good gate control. Moreover, emerging devices such as heterojunction tunneling FETs are more amenable to vertical fabrication. However, past studies of vertical channel devices focused more on regular memory architectures and simple standard cells like inverters. Since naïve migration of regular FinFET layouts to vertical FETs yields little benefits, we identify several vertical efficient layout structures and propose novel layout generation heuristics for vertical channel devices. We also compare VGAA with symmetric and asymmetric source/drain architectures and different contact placement strategies. The layout efficiencies of several VGAA structures, vertical double gate (VDG), lateral gate-all-around (LGAA), and FinFET are presented in our experiments. Routing congestion estimation on both cell-level and chip-level after placement and routing are also presented. We observe that even though most vertical channel standard cells have more diffusion gaps than lateral cells do, they still benefit from vertical architectures in area because of the vertically aligned top contacts. For asymmetric architectures, the area is larger than symmetric architectures because of the extra diffusion gaps needed, but our experiments indicate that for both symmetric and asymmetric architectures, vertical channel devices are likely to have a density advantage over lateral channel devices.**

***Index Terms*—CAD, vertical channel device, layout optimization, design rules, technology accessment.**

I. Introduction

Performance and size scaling demands of modern IC chips have become the driving forces to the development of new devices [1]. Vertically fabricated transistors, such as vertical gate-all-around (VGAA) [2], vertical double-gate (VDG) [3], and vertical heterojunction tunneling FET (VHTFET) [4] are being considered to be the alternative structures in the future. The concept of vertical channel FETs was proposed for more than two decades ago [5], but it did not catch much attention due to the complex fabrication process at that time. FinFET [6], instead, has become a more practical solution for scaled semiconductor technologies [7]. However, as conventional channel length scaling hits its barriers in the sub-10nm regime, vertically fabricated transistors are being reconsidered to be one of the replacements of FinFET devices [8]. Recent studies on vertical devices have demonstrated the improved fabrication process control and many appealing properties [9], [10]. Arrays of VGAA devices with 20nm diameter have been successfully fabricated, and good transistor characteristics such as large drive current, high I_{on}/I_{off} ratio, delay improvement [11], and better short channel effect control of VGAA have been observed [2], showing the potential opportunities provided by VGAA for the continued scaling of semiconductor devices.

Vertical heterojunction tunneling FET (VHTFET) is one of the vertical channel FETs with steep subthreshold swing and improved performance due to the decreased source-to-channel tunnel barrier height [4]. Because of the multi-junction nature, heterojunction tunneling FETs are more amenable to be fabricated vertically. The structure of VHTFET is similar to VGAA except that the source/drain terminal of VGAA MOSFET is interchangeable while VHTFET has a fixed source/drain structure [12]. Vertical slit FET (VESFET) is another emerging 3D device with four vertical pillars forming a device [13]. However, VESFET is not a vertical channel FET because the current flow is parallel to the wafer plane. It is similar to planar CMOS because source and drain are on the two sides of gate control, and the layouts of standard cells can be obtained automatically using Euler path-based algorithm [14]. For vertical channel FETs, a direct migration from planar to vertical layout generation will yield little benefits. Therefore, new layout design style and strategies are introduced in this paper to optimize transistor density for vertical channel FETs.

A. Introduction to Vertical FETs

Many vertical structures have been studied and discussed [11], [15], [12]. Unlike planar transistors, the current flow of vertical channel FETs is perpendicular to the wafer plane, which brings new challenges to efficient layout generation.

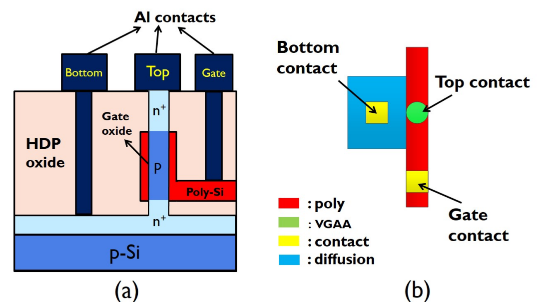


Fig. 1. VGAA device: (a) Cross section view of VGAA. (b) 2D layout view.

W. Wang and P. Gupta are with the Electrical Engineering Department, University of California Los Angeles, Los Angeles, CA 90095 USA. e-mail:(weichewang@ucla.edu; puneet@ee.ucla.edu).

This work is supported by IMPACT+ center (http://impact.ee.ucla.edu).

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

Figure 1 shows the cross section and 2D layout view of a VGAA transistor. Two ends of the vertical pillar are doped, and the middle of the pillar is surrounded by polysilicon gate. Contacts are connected to the top, bottom, and gate of the vertical pillar. Note that the gate extension can be aligned with the bottom and top contact plane [11] as shown in Figure 1(a), or be perpendicular to the bottom and top contact plane [12] as shown in the 2D layout in Figure 1(b). The efficient layout generation proposed in this paper is applicable to both vertical structures. However, we focus on the structure presented in Figure 1(b) because its layout resembles LGAA and FinFET more than the layout of Figure 1(a) does. The top contact serves as either a source or a drain terminal, and so does the bottom contact. Interestingly, even though the source and drain terminals are interchangeable, the device behavior differs significantly between the two architectures. When the top tip of vertical pillar serves as the source, the I_{on} is about 30% larger than the case where substrate side serves as the source, which could be due to low doping on the bottom side caused by the shadowing effect [2]. However, the results in [16] show that a two-stage inverter delay is nearly 50% higher when top tip of vertical pillar serves as the source because of the increased series resistance and load capacitance. Therefore, the electrode asymmetry and parasitics are important considerations for circuit design using VGAA. In our experiments, symmetric and asymmetric VGAA structures are compared (albeit only from a layout efficiency perspective), where symmetric means that source and drain are interchangeable, and asymmetric means that the top contact can only be served as source.

Besides the attractive characteristics of VGAA, another aspect that has impact on device performance is the crystalline orientation. Similar to FinFET, the channel of VGAA stands vertically on the wafer and can easily lie outside of the base crystallographic plane. In fact, on a (100) wafer, the surface orientation of VGAA is a mix of (110) and (100) because of the cylindrical channel shape. From the previous work on surface orientation optimization of FinFET [17], the surface orientation with the highest hole mobility and electron mobility is (110) and (100), respectively. Furthermore, since the PMOS enhancement on (110) is larger than the NMOS degradation due to velocity saturation, the overall delay can be improved by moving away from a standard (100) surface due to the enhancement of hole mobility.

The effective device performance of vertical structures compared to lateral structures is complex and beyond the scope of this work. Interested readers may refer to [11], [18] for some early studies. In this paper, we neglect the overall possible benefits of vertical channel devices by comparing the area using same effective width with lateral channel devices. Our focus is primarily to study layout efficiencies of vertical channel devices.

The VGAA fabrication process flow on 8-in bulk Si wafer has been studied and demonstrated. Figure 2 explains a common process flow [9], [10].

B. Related Work

Lot of research of VGAA application in memory devices has been done in the past because of the potential shrinking ability

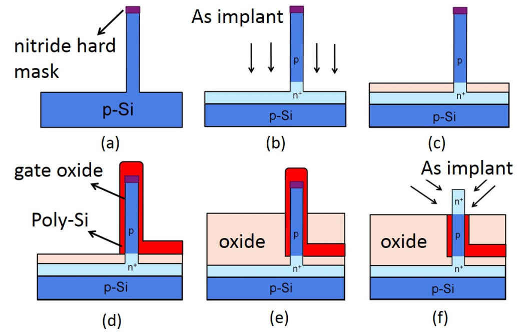


Fig. 2. Fabrication process flow of VGAA: (a) Space nitride hard mask patterning and pillar etching. (b) As implant. (c) Oxide deposition. (d) Gate oxide growth and polysilicon gate deposition. (e) Another oxide deposition. (f) Isotropic etch and pillar top implantation.

on both individual devices and multilevel memory structures [19], [15]. Studies on basic standard cells like inverter have also been done [11]. However, since standard cells use a large variety of layout structures, it would be difficult to evaluate the layout efficiency of a complete vertical channel standard cell library without using a systematic framework. For planar CMOS, lateral gate-all-around (LGAA) [20], [21], [22], and FinFET, the layout generation methodologies have been studied [23]. Frameworks for device optimization [24] and early stage design rule evaluation were also proposed [25], [26]. However, these algorithms cannot be applied to VGAA given that the structure of VGAA is radically different from planar or lateral FETs. Previous study showing evident area reduction of a vertical channel inverter cell is given in [12]. The area reduction comes from the elimination of diffusion contacts between the adjacent polysilicon gates. Replacing these diffusion contacts by top contacts as illustrated in Figure 1(b) helps reduce area significantly.

In our experiments, we extended the concept of contact space saving and performed a fair comparison on a full standard cell library. In this paper, we propose a systematic framework that generates efficient VGAA standard cell layouts and evaluates the impact of design rules as an early technology assessment of the emerging future vertical devices.

C. Our Contributions

Key contributions of this work are summarized as follows:

- We develop the first heuristics for effective layout generation for vertical channel devices.
- Layout efficiencies of several variations of VGAA, VDG, LGAA and FinFET are compared, including area and intra-cell wire length. Impact of design constraints on design benchmarks are also evaluated systematically.
- Cell-level area with intra-cell congestion estimation and chip-level area post placement and routing are both evaluated to compare VGAA and LGAA.

The rest of the paper is organized as follows. In Section II, variations of VGAA devices, efficient/inefficient vertical structures are described, followed by detail layout implementations such as wire length optimizations and bottom contact placement. The cell bipartite graph representation

and minimum chaining algorithm will be demonstrated in Section III. Section IV presents the design rules evaluation and experimental results on the proposed method. Finally conclusions are presented in Section V.

II. VGAA Layout Structures

A. Variations of VGAA Structures

We evaluated four kinds of VGAA structures to have a comprehensive understanding of the area impact of different VGAA cell architectures and patterning technology restrictions. The four architectures of VGAA are given below:

- *Fixed-Pitch VGAA (FVGAA):* FVGAA has regular rectangular polysilicon gate shape with fixed pitch. The polysilicon gate spacing is defined as the sum of contact width and two times of contact-poly spacing. The effective transistor width is the perimeter of the VGAA. Figure 3(a) shows an example of FVGAA 2D layout of an inverter with the bottom contact serves as source and top contacts are drain terminals. The driving strength of the PMOS is equal to six VGAA pillars and three VGAA pillars for NMOS width.
- *Contact Spacing Reduction Fixed-Pitch VGAA (RVGAA):* The polysilicon gate pitch of RVGAA could be one or two times of the minimum polysilicon gate spacing plus a polysilicon gate width, depending on whether or not a bottom contact is formed. Detailed design rules are given in section IV. As shown in the Figure 3(b), every polysilicon gate is still located on grid, but the spacing becomes less than half if no bottom contact is placed between two polysilicon gates. Therefore, RVGAA devices have less area than FVGAA for large drive cells with multiple polysilicon gates.
- *Polygon-Poly VGAA (PVGAA):* The architecture of PVGAA is given in Figure 3(c). Similar to [27], the shape of the polysilicon gate depends on the number of VGAA needed to form the cell. Array of vertical pillars are surrounded by a large polygon polysilicon gate shape, so the area becomes much smaller than FVGAA and RVGAA because contact spacing is smaller than polysilicon gate spacing. Similar VGAA array fabrication was demonstrated in [2], however, lithographic patterning of the surrounding irregular polysilicon gate shape is challenging. We include PVGAA with spacing rules same as FVGAA in our comparison to give an idea of how much benefit it could have compared with FVGAA.

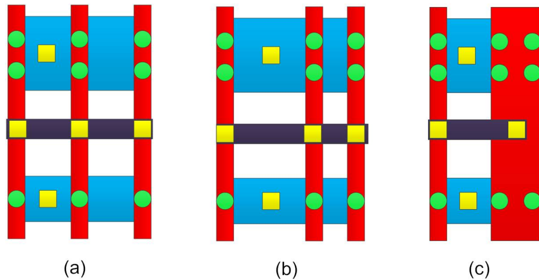


Fig. 3. VGAA structures: (a) FVGAA. (b) RVGAA. The polysilicon gate is still located on grid. (c) PVGAA.

- *Hetero-VGAA (HVGAA):* HVGAA can be considered as a mixed version of RVGAA and PVGAA with different cell heights. The shape of HVGAA polysilicon gate is fixed to one or two VGAA widths plus VGAA pitch, and the cell height depends on the number of VGAAs that can be held in a polysilicon gate shape. The number of VGAAs in each polysilicon gate shape can be viewed as arrays of columns and rows "1x1", "4x2", or other combinations as shown in Figure 4, where each HVGAA variation contains eight VGAAs. The structure of standard cells within one library is the same so the cells can be easily abutted and power rails can be shared. Due to the two extra tracks left for power rails on the upper and lower part of the cell, the best suited HVGAA structure depends on the strengths of cells used in the design. Short HVGAA variations, such as structures shown in Figures 4(a) and 4(d), are more efficient for small driving cells. On the other hand, tall HVGAA variations, such as structures shown in Figures 4(c) and 4(f), are more efficient for large driving cells. There are dummy polysilicon gate shapes in the "4x2" structure because a diffusion space is needed for bottom contact placement, which shows that the "4x2" structure is not an efficient structure for cells with small number of VGAAs.

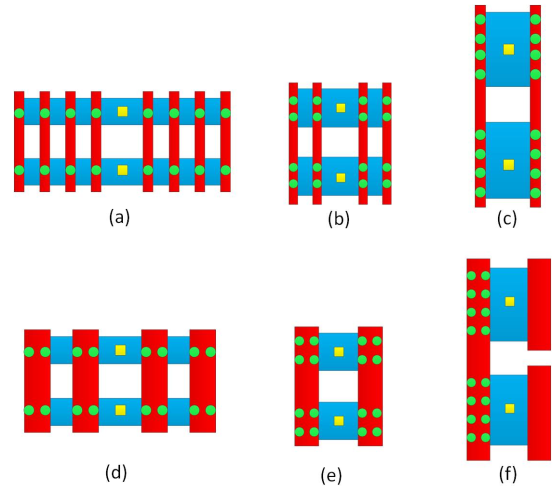


Fig. 4. HVGAA structures: (a) 1x1 (b) 2x1 (c) 4x1 (d) 1x2 (e) 2x2 (f) 4x2

B. Vertical Efficient Structures

In this section, we will introduce vertical efficient structures that provide efficient layouts given the restriction of vertical channel structure, and our VGAA default structure is FVGAA. For CMOS layouts, sharing the same active/diffusion region (what we refer to as a chain in this paper) between different transistors results in fewer diffusion breaks and smaller area. The definition of chain is the same as in [25], which means pairs of P-N transistors that share a same diffusion strip. For instance in conventional lateral channel devices, two parallel connected transistors and any number of series connected transistors can share the same diffusion region or chain. Less number of chains means smaller area. Since the structure of VGAA is radically different from lateral FETs, some CMOS

circuit schematics that are considered efficient in lateral FET may not be the most efficient schematics for VGAA. Figure 5 shows an example of how a 3-Parallel structure is implemented in VGAA on a single chain, while in FinFET, at least two chains are needed to realize the structure as the drain terminals of A, B, and C are not connected. In addition, only one bottom contact is needed for VGAA, which makes the area even smaller.

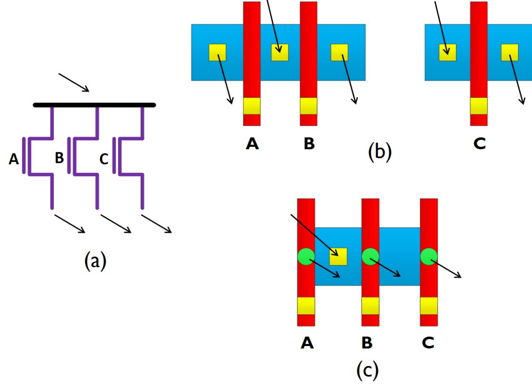


Fig. 5. 3-Parallel structure: (a) Transistor schematic. (b) FinFET. (c) VGAA.

In order to find out the best structure of VGAA layout, we categorize two types of vertical efficient layout structures.

- *n-Parallel:* The n-Parallel structure is composed of n VGAA devices with a connected source or drain terminals. The cross section view and schematic of a 3-Parallel structure is given in Figure 6. Three VGAA devices are shared on a same diffusion strip, where the shared terminal can be either a source or a drain. Three gate contacts are placed perpendicular to the bottom and top contact plane as shown in the 2D layout Figure 1(b).

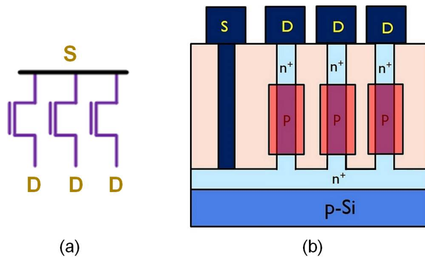


Fig. 6. n-Parallel structure: (a) Transistor schematic. (b) VGAA cross section view.

- *2-Stack n-m-Parallel:* A 2-Stack n-m-Parallel structure consists of two stacked n-Parallel structures, where the number n and m can be equal to or greater than one. Figure 7 illustrates the cross section view of a 2-Stack 3-3-Parallel structure. Note that for asymmetric structures, this is not a valid vertical structure because source and drain are not interchangeable.

Any circuit schematic of these two forms can be realized using only one vertical channel chain, and a single chain is the preferred structure in minimizing the layout area. These schematic patterns for efficient layout will be identified in a systematic way and become the input to the proposed minimum chaining algorithm for vertical channel devices.

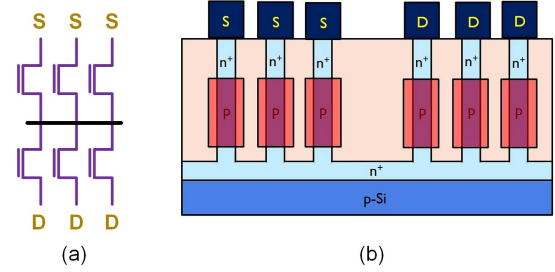


Fig. 7. 2-Stack n-m-Parallel Structure: (a) Transistor schematic. (b) VGAA cross section view.

C. Vertical Inefficient Structures

In contrast to vertical efficient structures, here are two examples of vertical inefficient structures that can be realized by lateral FETs in one chain but will require multiple chains in vertical FETs.

- *3-Stack:* A 3-Stack schematic is given in Figure 8(a). In VGAA, the minimum number of chains to realize this structure is two. The reason is that to form a 2-stack structure in VGAA, the bottom diffusion must be shared by the stacked transistors. To cut off the shared diffusion, instead of by a polysilicon gate control like FinFET, the only way for VGAA is to introduce another chain, which explains why a 3-Stack structure cannot be implemented in VGAA with only one chain. Figures 8(b) and (c) show examples of how a 3-Stack structure can be implemented in FinFET in one chain, while in VGAA the minimum number of chains is two.

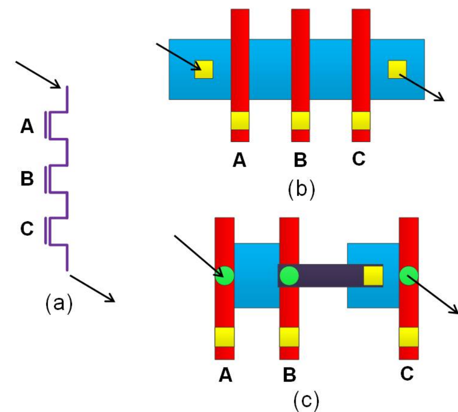


Fig. 8. 3-Stack structure: (a) Transistor schematic. (b) FinFET. Only one chain is needed. (c) VGAA. Two chains and a metal connecting segment are needed.

- *Stack-Parallel:* A Stack-Parallel structure is similar to n-Parallel, but at least one of the paralleled structures is a stack of transistors. Figure 9 gives an example of a Stack-Parallel structure. Unfortunately, this structure appears in many standard cells, making most of the vertical channel structures having more chains than lateral channel structures.

D. Intra-cell Wire Length Optimization

Another layout benefit of vertical channel is that for an n-Parallel chain, the bottom contact can be placed on multiple

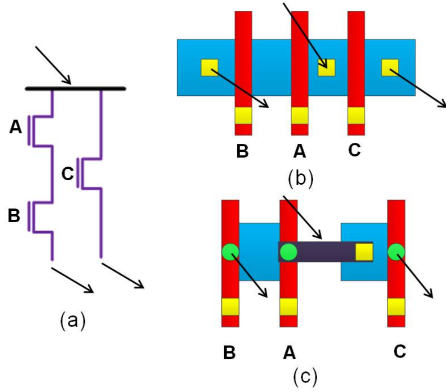


Fig. 9. Stack-Parallel structure: (a) Transistor schematic. (b) FinFET. Only one chain needed. (c) VGAA. Two chains are needed.

locations of the diffusion strip because all parallel transistors share diffusion region. This flexibility helps to reduce the intra-cell wire length of vertical channel FETs. Figure 10 shows an example of how a wire length can be reduced by moving the bottom contacts closer to each other, where the initial locations are on either the rightmost or the leftmost of the chain. A chain has at most one P/N bottom contact, and each bottom contact belongs to only one net. The total wire length estimation is obtained by summing the half parameter wire length (HPWL) of each net. Since the number and constraints of intermediate routing layers may differ between vertical channel device implementations, using HPWL provides a general estimate of the routing track usage, which is used in the congestion estimation. Note that wire length of VDD/GND is not counted in HPWL. Steps of minimizing HPWL are described below:

1) For each net, identify its leftmost and rightmost ends. If the two ends are on the same location horizontally, the HPWL cannot be reduced because the distance between PMOS part and NMOS part diffusions are assumed to be fixed. If the two ends are not on the same location horizontally, proceed to step 2.
2) If any of the two ends is a bottom contact, move the leftmost bottom contact to right, and move the rightmost bottom contact to left.

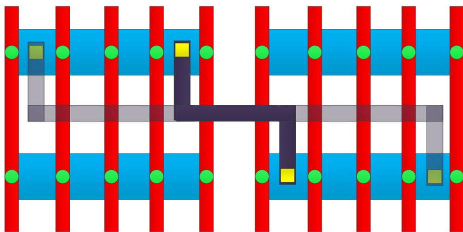


Fig. 10. Net HPWL reduced by moving bottom contacts.

E. Bottom Contact Placement

Though placing a bottom contact on different locations of a chain will not affect the functionality, the high resistivity of the bottom diffusion layer can cause parasitics-induced performance degradation. Therefore, the performance-congestion tradeoffs should also be considered by adding extra constraints to bottom contact placement, such as the maximum number of VGAAs supported per bottom contact. For VGAAs, adding more bottom contacts also means that more area would be required. Figures 11(a) and (b) show two alternative layouts for four parallel-connected devices and the tradeoff between area and wire length. Details and comparisons of the constraints will be presented in Section IV-D. The intra-cell wire length

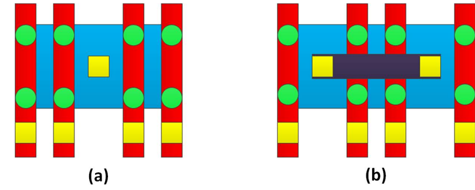


Fig. 11. (a) One contact with higher diffusion layer parasitics. (b) Two contacts with lower parasitics requiring larger area and extra connecting metal segment.

optimization can be adapted straightforwardly as illustrated in Figure 12 for RVGAA where a bottom contact has to be placed for every two polysilicon gates. Instead of moving one bottom contact on a chain, multiple contacts with fixed distances are moved together, and a polysilicon gate is skipped whenever a bottom contact is placed as described in Section II.

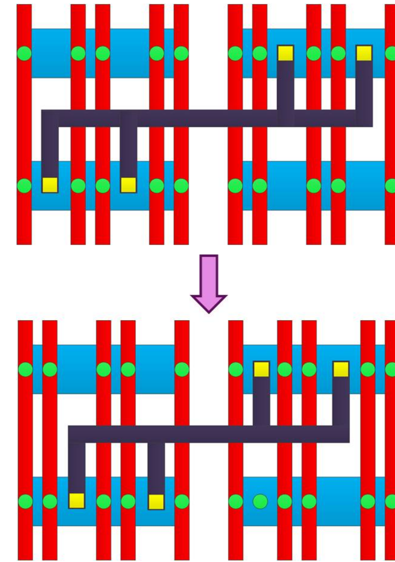


Fig. 12. Net HPWL reduction with bottom contact constraint.

III. Vertical Channel Layout Generation Methodology

The proposed vertical channel layout generation methodology is divided into two steps. The first step is the development of the bipartite graph, from which the vertical efficient layout structures can be easily identified. The second step is to find the minimum number of chains by finding the minimum set of edges that cover all transistors in the bipartite graph.

A. Bipartite Graph Representation

We first define the graph notation in a similar fashion as [23]. The triple (T, D, S) represents the three attributes of

a transistor t, where $T(t)$(= P or N) indicates whether t is a PMOS or NMOS, $D(t)$ and $S(t)$ represent the connecting net of the drain and the source terminal respectively. A P-N transistor pair $P(t_i, t_j)$, contains two transistors where $T(t_i) = P$ and $T(t_j) = N$. In this paper, we consider perfect pair implementation only [25], so a pair $P(t_i, t_j)$ means that t_i and t_j have the same gate input signal. To identify the vertical efficient layout structures, we represent the transistor schematic using a bipartite graph $G = (V_p \cup V_n, E)$. Each vertex in V_p or V_n corresponds to a set of PMOS or NMOS transistors that form one of the two vertical efficient layout structures. Once vertices are constructed, an edge is built between two vertices if the two vertices contain at least one P-N transistor pair. That is, each edge *covers* all the common transistor pairs between a V_p and V_n. This is different from the previous work [23] because for vertical channel devices, more than two transistors in a vertex can be formed on a chain, and each edge corresponds to a chain. Once the bipartite graph is built, we apply the proposed minimum edge covering algorithm to find out the chaining solution to the cell implementation. The formal description of E, V_p, and V_n are given:

$$E = \{E_{ij} : t_1, t_2, ..., t_k | \{t_1, t_2, ..., t_k\} = V_p i \cap V_n j\}$$
$$V_p = \{V_p i | V_p i = \{t_p | T(t_p) = P \cap (D(t_p) = i \cup S(t_p) = i)\}\}$$
$$V_n = \{V_n j | V_n j = \{t_n | T(t_n) = N \cap (D(t_n) = j \cup S(t_n) = j)\}\}$$

The description of the bipartite graph given is for symmetric structures. The representation for asymmetric structures can be easily obtained by first splitting each vertex representing a 2-Stack n-m-Parallel into two n-Parallel vertices, and then building the edges in the same fashion. It is obvious that asymmetric structure is very likely to have more chains than symmetric structures because one of the two vertical efficient structures does not exist anymore. In our benchmark experiments, both symmetric and asymmetric VGAA structures are compared.

Figure 13 gives an example of the bipartite graph representation and schematic of a symmetric VGAA AOI21 standard cell. In the bipartite graph, the node V_p1, for example, contains three PMOS transistors that either their sources or drains are on net V_p1 as illustrated in the schematic. V_p1 itself represents a 2-Stack 1-2-Parallel vertical efficient structure, meaning that PMOS transistors A, B1, and B2 can be realized on a single chain. However, we need to consider the pairing with NMOS transistors by selecting edges in the graph. Edge E_{11} connects V_p1 and V_n1 and represents transistor pairs B1 and B2, so selecting edge E_{11} means that a chain is needed to realize pairs B1 and B2.

B. Minimum Edge Covering Algorithm

The goal of the algorithm is to select all P-N transistor pairs with the minimum number of edges. In the bipartite graph, each edge represents a set of P-N transistor pairs that can be realized on a chain, so a minimum edge covering algorithm is proposed to minimize the number of chains needed to implement a cell. The notation $E(P(t_i, t_j))$ is defined as the set of edges that cover the P-N transistor pair $P(t_i, t_j)$. A P-N transistor pair $P(t_i, t_j)$ is *covered* if at least one edge belonging to $E(P(t_i, t_j))$ is selected.

The algorithm is described as follows:

- For a bipartite graph, we generate a connection table for all edges and pairs. Each edge is represented by a column, and each pair $P(t_i, t_j)$ is represented by a row. If an edge covers a pair, the value of the corresponding location is set to one, otherwise it is set to zero. The $E(P(t_i, t_j))$ of each pair is obtained after the connection table construction by finding the edges corresponding to ones in the row of $P(t_i, t_j)$.
- Identify *essential edges* by observing the connection table. If a row contains only a single 1 or a column contains only 1, the corresponding edge of the column is an essential edge. Every essential edge is selected and removed from the connection table, along with all transistor pairs corresponding to the edge.
- For the remaining connection table, apply Petrickś method [28] by formulating the edges as Boolean variables and P-N transistor pairs as minterms. To cover a minterm $P(t_i, t_j)$, the sum of variables in $E(P(t_i, t_j))$ must be true. The logical function $F(G)$ is defined as the product of all minterms, and it must be true because all minterms must be covered.
- $F(G)$ is further simplified by using simple Boolean simplification technique $X + XY = X$. The product term with the least number of edges along with essential edges are returned by the algorithm as the minimum set of edges to cover all P-N transistor pairs.

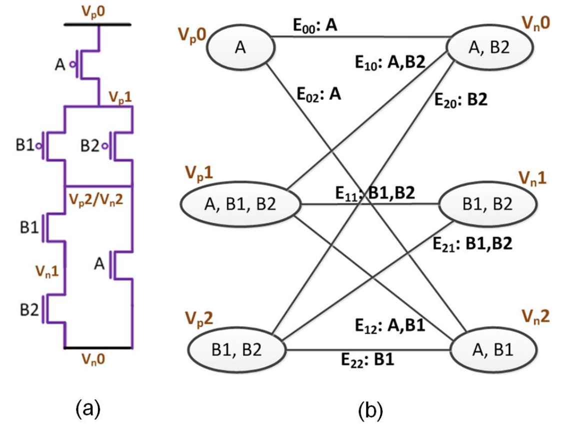


Fig. 13. AOI21 schematic and bipartite graph representation: (a) Schematic. (b) Bipartite representation.

TABLE I
CONNECTION TABLE OF AOI21

	E00	E02	E10	E11	E12	E20	E21	E22
A	1	1	1	0	1	0	0	0
B1	0	0	0	1	1	0	1	1
B2	0	0	1	1	0	1	1	0

Following the above procedure, the chaining solution of a vertical channel cell is then obtained. Table I gives an example of the connection table of AOI21. From the table we know that to cover P-N transistor pair of A, for example, the expression $(E_{00} + E_{02} + E_{10} + E_{12})$ must be true.

Since there are no essential edges for the AOI21 cell, we directly apply Petrickś method to form the Boolean functions

given below:

$$A = E_{00} + E_{02} + E_{10} + E_{12}$$
$$B1 = E_{11} + E_{12} + E_{21} + E_{22}$$
$$B2 = E_{10} + E_{11} + E_{20} + E_{21}$$
$$F(G) = A \cdot B1 \cdot B2$$

$F(G)$ must be true to cover all the transistor pairs A, $B1$, and $B2$. After Boolean expansion and simplification of $F(G)$, there are eleven product terms with only two edges, which is the minimum set of edges needed. To make $F(G)$ true, one of these product terms is selected and set to be true, for example, $E_{00}E_{11}$. Now we know that the vertical channel cell AOI21 needs two chains to be realized, and the transistor pair on the first chain E_{00} is A, and the transistor pairs on the second chain E_{11} are $B1$ and $B2$. The ordering of transistors with minimum wire length is then decided by the procedure described in [25], which includes solving a min-cut placement problem and possible chain flipping. The final chaining result of AOI21 is shown in Figure 14.

The PMOS (upper) part of E_{11} is a 2-Parallel structure, so a bottom contact is placed as shown in Figure 5(c), while no bottom contact is placed in the NMOS (lower) part because it is a 2-Stack 1-1-Parallel structure similar to the stacked transistors A and B shown in Figure 8(c). Note that for asymmetric architecture, chain E_{11} will have to be split because the 2-Stack 1-1-Parallel structure of the NMOS is not valid.

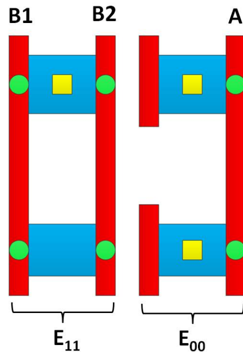


Fig. 14. AOI21 chaining result. Two chains E_{11} and E_{00} are needed.

C. Chaining Efficiency Improvement

For RVGAA, to further utilize the benefits of vertical efficient structures, the cell schematic can be slightly tweaked to create efficient 2-Stack structures without changing the functionality of the cell. As an example cell AOI21 shown in Figure 15. A Stack-Parallel vertical inefficient structure on the NMOS part can be transformed into an efficient 2-Stack 2-2-Parallel structure by duplicating transistor A and creating a net connecting to all transistors. After the transformation, the cell can be implemented on a single chain with smaller area than the one shown in Figure 14. However, this transformation only benefits symmetric structures because it creates a 2-Stack efficient structure, which is not allowed in asymmetric structures. When the transistor is duplicated and stacked, the sizes of new transistors (A_1 and A_2) are enlarged according to its adjacent stacked transistors (B_1 and B_2) to match the delay, and the corresponding PMOS transistor is also duplicated in parallel as illustrated.

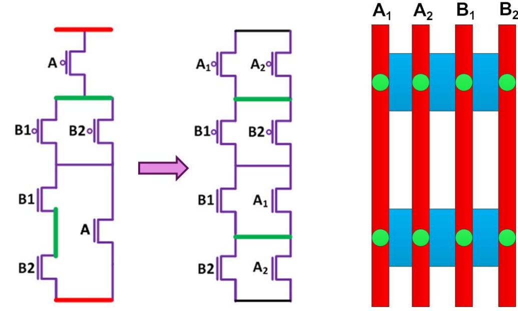


Fig. 15. AOI21 improved chaining result. Transistor A is duplicated, and the new connection forms a 2-Stack 2-2-Parallel vertical efficient structure. Only one chain is needed after the transformation.

Nevertheless, not all transformations are beneficial since the number of transistor pairs becomes more than the original structure. As demonstrated in Figure 16, instead of making the area smaller, the transformation increases the number of polysilicon gates and thus the area becomes larger. Therefore, the decision should be made carefully.

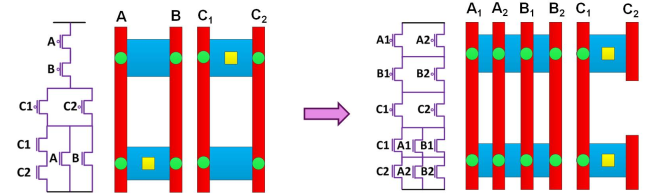


Fig. 16. Transformation worsens the area.

Consider general structures before the transformation as given in Figure 17(a), where X is the number of 2-Stacks, and Y is the number of single transistors to be duplicated. To determine if a candidate structure of this form will benefit from the transformation, a quick estimation is proposed to help making the decision. A key observation here is that each

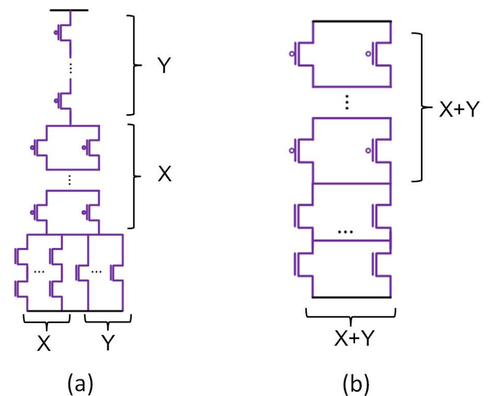


Fig. 17. (a) Before transformation. (b) After transformation.

of the vertical efficient structures implementing this structure

will need a bottom contact because at least one side of NMOS or PMOS part is a parallel structure. Also, since no vertical efficient structures can be formed by more than two pairs of transistors in this structure, the number of chains can be calculated as $\lceil \frac{Y}{2} \rceil + X$. Therefore, the width of final RVGAA can also be calculated as given below:

$$w_before = \lceil \frac{Y}{2} + X \rceil \times (2 \times Poly_width + PS_R + PS_R2) \quad (1)$$

where PS_R and PS_R2 are polysilicon gate spacings of RVGAA with and without bottom contact, respectively as given in Figure 18 .

Now consider the case after the transformation as given in Figure 17(b), where the number of transistors becomes $2(X+Y)$. Since all vertical structures are 2-Stack structures except for the remaining pair, if any (when $X+Y$ is an odd number), the width can be calculated from the number of transistors as given below:

$$\begin{aligned} w_after = & 2(X+Y) \times (Poly_width + PS_R) + \\ & ((X+Y) \mod 2) \times (PS_R2 - PS_R)) \end{aligned} \quad (2)$$

Equations 1 and 2 give a quick indication of the choice that only when w_after is smaller than w_before will the transformation be beneficial. However, there are still other limitations. If a cell contains other transistors besides the ones that form the general case as shown in Figure 17(a), and the other transistors do not form a vertical efficient structure by themselves, then the transformation may not be helpful. In such case, we directly generate and compare the results between before and after the transformation and choose the one with smaller area.

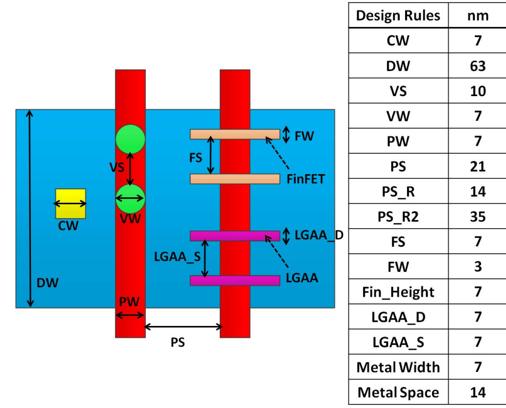
IV. Experimental Results

The proposed algorithm is implemented in C++ with the use of OpenAccess (OA) API [29] for GDSII file generation. We have worked with a 7nm standard cell library, which is a scaled version of commercial 28nm standard cell library, and the design rules are given in Figure 18. For RVGAA, PS_R is the polysilicon gate spacing when no bottom contact is placed, and PS_R2 is the polysilicon gate spacing when a bottom contact is placed. Two metal layers are used in the intra-cell routing congestion estimation, where the width and spacing of the metal layers are aligned with polysilicon gate width and PS_R, respectively. Note that the results of the comparison will strongly depend on the design rules chosen. The framework itself is, of course, applicable if different design rules are used.

The comparisons are evaluated on four design benchmarks [30]. Details of these benchmarks synthesized using the complete commercial standard cell library are given in Table II.

A. Cell Area Comparison

Table III shows the cell area comparison between LGAA, the three VGAA symmetric structures, and VDG on simple cells with different driving strengths and a complex flip-flop. The cell heights in this comparison are equivalent to the "4x1" version of HVGAA as shown in Figure 4(c), where four VGAAs can be placed on each side. The values shown are the



Design Rules	nm
CW	7
DW	63
VS	10
VW	7
PW	7
PS	21
PS_R	14
PS_R2	35
FS	7
FW	3
Fin_Height	7
LGAA_D	7
LGAA_S	7
Metal Width	7
Metal Space	14

Fig. 18. Design Rules. PS_R and PS_R2 are for RVGAA polysilicon gate spacings.

TABLE II
Synthesis Results of Four Benchmarks

Benchmark	Sequential		Combinational	
	Instances	Area %	Instances	Area %
MIPS	1947	60.2	8307	39.8
FPU	656	12.4	22969	87.6
USB	1726	59.7	6278	40.3
AES	530	14.0	19827	86.0

percentages of change in cell area in comparison to FinFET.[1] For the layouts of LGAA and FinFETs, the generation is based on the framework presented in [25]. Positive percentages indicate that the device is larger than FinFET and negatives mean that the device is smaller than FinFET.

For LGAA, the area is smaller than FinFET for some cells because they are both lateral channels, and LGAA has a larger effective transistor width per polysilicon gate than FinFET. For FinFET, the effective width of each fin is $FW + Fin_Height \times 2 = 17$, while the effective width of each LGAA is $LGAA_D \times \pi \approx 22.0$. For both FinFET and LGAA, each polysilicon gate can hold up to seven fins or LGAAs, so the effective transistor width per polysilicon gate for FinFET is $7 \times 17 = 119$, and for LGAA it is $7 \times 22.0 \approx 154.0$. Larger effective transistor width per polysilicon gate means less cell width and cell area.

For FVGAA, the area benefit of large driving strength

[1]More area efficient isolation techniques for FinFET (e.g., [31]) may improve FinFET results over what is presented here.

TABLE III
Percentages of Change in Area in Comparison to FinFET

Cell	LGAA	FVGAA	RVGAA	PVGAA	VDG
INV_X1	0.0%	-50.0%	-30.2%	-50.0%	-50.0%
INV_X2	0.0%	-50.0%	-30.2%	-50.0%	-50.0%
INV_X4	0.0%	-50.0%	-30.2%	-50.0%	-50.0%
INV_X8	0.0%	-66.7%	-53.5%	-66.7%	-66.7%
INV_X16	-20.0%	-40.0%	-44.2%	-54.4%	-60.0%
INV_X32	-12.5%	-12.5%	-30.2%	-42.4%	-50.0%
NAND2_X1	0.0%	-66.7%	-53.5%	-66.7%	-66.7%
NAND2_X2	0.0%	-66.7%	-53.5%	-66.7%	-66.7%
NAND2_X4	0.0%	-66.7%	-53.5%	-66.7%	-66.7%
NAND2_X8	0.0%	-40.0%	-44.2%	-52.1%	-40.0%
DFF_X1	0.0%	-51.9%	-61.2%	-51.9%	-51.9%

inverters and NAND gates is not as significant as smaller driving cells because the number of VGAAs that can be held per polysilicon gate is only five according to the design rules. For small driving cells, both lateral and vertical cells need only one active polysilicon gate pitch, thus the contribution of removing diffusion contact becomes evident [12]. For flip-flops, the number of polysilicon gates is much larger than an inverter or a NAND gate because of its complex cell structure, but the area of the flip-flop is still much smaller than FinFET for all vertical devices. The reason is that many transistors connected to VDD and GND in flip-flops form a huge n-Parallel structure, which is a vertical efficient structure. FinFET needs more chains than VGAA to form the structure as given in Figure 4.

Compared to FVGAA, RVGAA shows more area reduction on cells with multiple shared polysilicon gates because of the small PS_R when no bottom contact is placed. PVGAA also shows higher layout efficiency than FVGAA in large driving cells because VS is smaller than PS, and large driving cells require multiple polysilicon gates. VDG shows high area efficiency because of the larger effective width per polysilicon gate than FinFET in addition to the diffusion contact removal.[2]

B. Symmetric and Asymmetric Architectures

The results of area and intra-cell HPWL comparison on both symmetric and asymmetric RVGAA to LGAA are given in Table IV. For asymmetric structure, the area and HPWL are both worse than symmetric structure because for asymmetric structures, each chain realizing a 2-Stack n-m-Parallel must be split and it increases the number of chains. In symmetric architecture, 54% of all the chains generated are 2-stack structures. However, this ratio does not directly translate to the area increased from symmetric to asymmetric architectures because it mainly depends on how long the chain is (how many transistors on the chain), also there could be other solutions for the minimum edge selection that contain only n-Parallel structures after the split, which will not increase the area at all. In addition, many widely used basic cells, such as INV and BUF, and large cells like DFF and SDF, do not contain 2-stack structures. AOI21 and XOR2 are two cells benefit from the transformation illustrated in Figure 15. Results show that their symmetric implementations after the transformation are much better than asymmetric implementations and before the transformation for both area and HPWL. The asymmetric results remain the same after the transformation because 2-Stack structures are not allowed.

Figure 19 gives the total design area ratio (in percentage) of "4x1" HVGAA (RVGAA) to LGAA. For all the benchmarks, the area of asymmetric architecture is much larger than symmetric architecture as expected, but even with the asymmetric assumption, RVGAA area is still about 20% smaller than LGAA. The ratio of symmetric architecture after the transformation is about 2% to 5% smaller than symmetric architecture before the transformation, and both of them are smaller asymmetric architecture.

[2]VGAA pillar diameter and pitch rules can alter this conclusion. In our current experiment, we assume pillar design rules to be same as contact rules.

TABLE IV
SYMMETRIC/ASYMMETRIC RVGAA COMPARISON TO LGAA

Cell	Symmetric		Asymmetric	
	Area	HPWL	Area	HPWL
AOI21_X1	-30.2%	-30.2%	-16.3%	82.9%
*AOI21_X1	-19.6%	16.4%	-16.3%	82.9%
BUFFER_X1	-47.7%	-7.0%	-47.7%	-7.0%
DFF_X1	-57.6%	-15.1%	-57.6%	-15.1%
SDF_X1	-58.9%	-8.8%	-58.9%	-8.8%
XOR2_X1	-20.3%	-9.0%	19.6%	150.6%
*XOR2_X1	11.8%	88.2%	19.6%	150.6%

* Indicates the cell before transformation

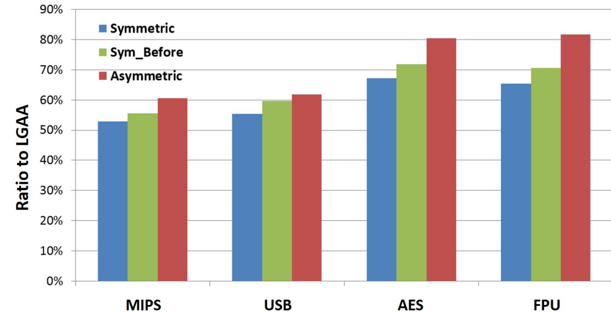


Fig. 19. Benchmark results of symmetric, symmetric before transformation, and asymmetric RVGAA to LGAA.

C. HVGAA Intra-cell Routing Congestion Estimation

As demonstrated in Figure 4, short cells are much more area efficient than tall cells for designs with many small driving cells. However, as areas become smaller, the routing becomes more congested, thus extra area accommodating routing congestion for each cell has to be estimated and taken into consideration in the comparisons. Unless otherwise specified, all results of our work are based on the congestion estimation in [25], where the cell area is increased to accommodate routing congestion if necessary. In this section, we specifically compare the area before and after applying congestion estimation to all symmetric HVGAA variations. As shown in Table V, "1x1" and "1x2" HVGAA variations require up to 20% and 13% extra area due to routing congestion, respectively. For tall and wide cells such as "2x2", "4x1", and "4x2", no extra area is needed because of their sufficient number of routing tracks.

Figure 20 shows normalized areas of the four design benchmarks after congestion accommodation. We see that "2x1" becomes the most efficient HVGAA because its required additional area due to routing congestion is not as much as "1x2", which was originally the smallest HVGAA before

TABLE V
AREA INCREASE DUE TO INTRA-CELL ROUTING CONGESTION

HVGAA	MIPS	USB	AES	FPU
1x1	13.4%	16.9%	20.8%	18.9%
2x1	1.1%	1.6%	2.3%	2.5%
4x1	0.0%	0.0%	0.0%	0.0%
1x2	13.2%	9.6%	2.9%	4.3%
2x2	0.0%	0.0%	0.0%	0.0%
4x2	0.0%	0.0%	0.0%	0.0%

congestion estimation. Compared with the baseline symmetric "4x1" HVGAA shown in Figure 19, area reduction of "2x1" to LGAA can be expected to be larger even with routing congestion considered.

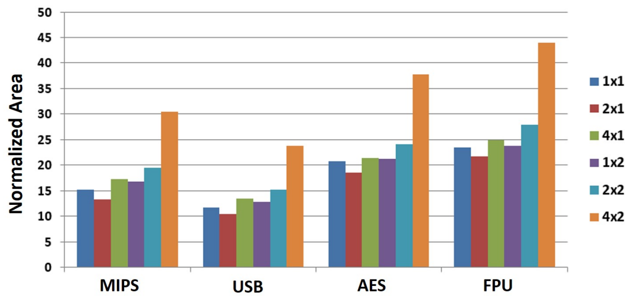


Fig. 20. Normalized area after intra-cell routing congestion estimation.

D. Comparison of Bottom Contact Placement

As demonstrated in Figure 11, different contact placements can lead to different area and routing congestions. In this section, results of three additional bottom contact placement constraints with routing congestion estimation are compared. As illustrated in Figure 21, constraint $SPACE1$ means that the number of polysilicon gates between any two contacts is at most one, where for constraint $SPACE2$ and $SPACE3$, the number of polysilicon gates between any two contacts is two and three, respectively. Structures with $SPACE3$ constraint have smaller area than other constraints because less bottom contacts are needed.

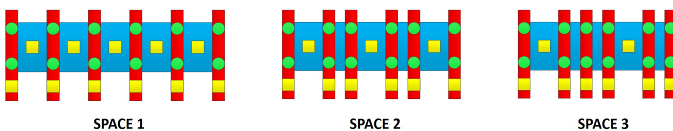


Fig. 21. Different bottom contact placement constraints.

Figure 22 shows the normalized MIPS area comparisons with bottom contact placement constraints and the structures without constraints. For "4x1" HVGAA, the number of VGAAs on a polysilicon gate is four, while the number of VGAAs on a polysilicon gate of "1x1" HVGAA is only one. Therefore the total number of polysilicon gate needed for "4x1" HVGAA is much smaller than "1x1" HVGAA for the same cell. The impact of contact constraints on area becomes more evident as the number of polysilicon gate becomes larger, thus "1x1" HVGAA shows a significant area overhead when the constraints are imposed.

E. Chip-Level Placement and Routing

Cell-level evaluation can be sometimes misleading due to chip-level congestion [32]. We use symmetric RVGAA with no bottom contact constraints as our baseline to demonstrate the final area comparison to LGAA after chip-level placement and routing using a commercial tool [33], and totally four metal layers are used for chip-level routing. Table VI and Table VII show detailed reports of final DRC-clean results of LGAA

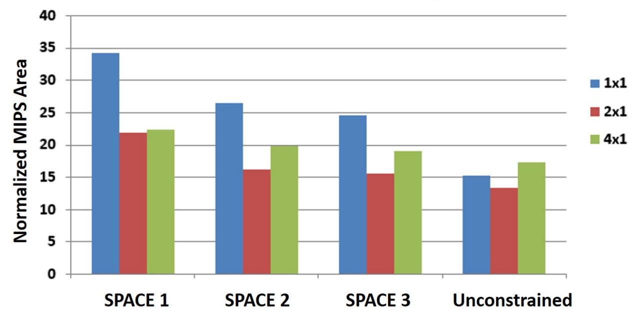


Fig. 22. Normalized MIPS areas of different bottom contact placements.

TABLE VI
LGAA PLACEMENT AND ROUTING REPORTS

	CPU Time (s)	Wire Length (mm)	Util.	Init. DRC	Congestion
FPU	1856	224.49	0.83	768	28.0%
USB	1861	159.04	0.96	1876	26.6%
AES	1868	290.52	0.81	7457	39.8%
MIPS	2022	196.08	0.86	1837	22.3%

and RVGAA, respectively. The CPU time is similar between LGAA and RVGAA, and the wire length shown in the third column shows that RVGAA has shorter wire length due to smaller area. The forth column shows that the utilization of RVGAA is much smaller than LGAA. The numbers shown in the fifth column indicate the number of DRC violations when detailed routing starts, and the percentages of over-congested gcells are shown in the last column.

Figure 23 shows RVGAA area reduction compared to LGAA before placement and routing (only the total cell area is compared) and after placement and routing. We can see that area reduction ratios after placement and routing become 5% to 16% worse than before placement and routing. This may partly be due to higher pin densities in VGAA. However, even though the effective utilization of RVGAA is not as high as LGAA, our results show that RVGAA can still achieve 20% to 40% area reduction in comparison to LGAA after chip-level placement and routing.

V. CONCLUSION

In this paper, we develop the first framework and heuristics for efficient layout generation for standard cells using vertical channel devices (available for download at http://nanocad.ee.ucla.edu/Main/DownloadForm). Several vertical efficient and inefficient layout structures are identified to explain the difference in layout generation strategies between vertical and lateral devices. Symmetric and asymmetric vertical channel architectures and bottom contact placement

TABLE VII
RVGAA PLACEMENT AND ROUTING REPORTS

	CPU Time (s)	Wire Length (mm)	Util.	Init. DRC	Congestion
FPU	2183	188.69	0.76	2361	44.6%
USB	2309	106.78	0.82	3148	49.7%
AES	1843	274.32	0.65	8045	53.6%
MIPS	1909	143.74	0.68	2656	41.1%

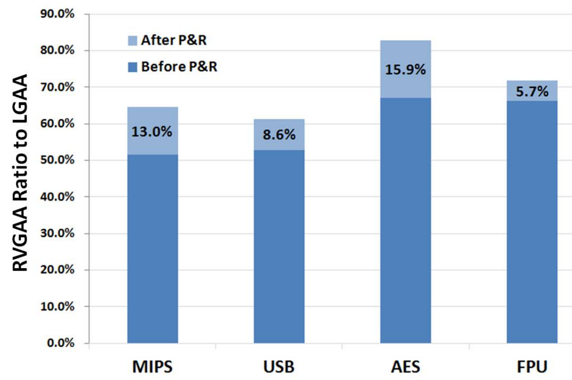


Fig. 23. Area benchmark results before and after chip-level placement and routing.

strategies are also discussed and compared. The layout efficiencies of several VGAA structures, VDG, LGAA, and FinFET are compared in our experiments, along with congestion estimation analysis on both cell-level and chip-level. Complete placement and routing are performed on four benchmark designs using a commercial standard cell library. Our results show that standard cells and designs implemented by vertical channel devices are likely to have smaller area even with cell-level and chip-level routing congestion estimation. Even though several simple standard cells are composed of vertical inefficient structures, vertical structures provide the ability of placing a top contact aligned with a vertical channel and thus still reduce the area significantly. Our future work will study the layout generation for different vertical structures, such as inbound power rails with bottom contact escaping from upper/lower part of the cell [34], and the impact of parasitics on performance of different vertical cells.

Acknowledgement

The authors would like to acknowledge the generous support of IMPACT+ center (http://impact.ee.ucla.edu). The authors would also like to thank Prof. Tsu-Jae King Liu (UC Berkeley), her student Peng Zheng (UC Berkeley), Dr. Arindam Mallik (IMEC), Dr. Praveen Raghavan (IMEC), Dr. Julien Ryckaert (IMEC), Trong Huynh Bao (IMEC), Dr. Rani Ghaida (GF), and Yasmine Badr (UCLA) for their valuable inputs and discussions.

References

[1] J. Hutchby, G. Bourianoff, V. Zhirnov, and J. Brewer, "Extending the Road beyond CMOS," *IEEE Circuits and Devices Magazine*, Mar 2002.
[2] B. Yang, K. Buddharaju, S.-G. Teo, N. Singh, G. Q. Lo, and D. L. Kwong, "Vertical Silicon-Nanowire Formation and Gate-All-Around MOSFET," *IEEE Electron Device Letters*, July 2008.
[3] H. Cho, P. Kapur, P. Kalavade, and K. Saraswat, "A Low-Power, Highly Scalable, Vertical Double-Gate MOSFET Using Novel Processes," *IEEE TED*, Feb 2008.
[4] G. Dewey *et al.*, "Fabrication, Characterization, and Physics of III-V Heterojunction Tunneling Field Effect Transistors (H-TFET) for Steep Sub-Threshold Swing," in *IEEE IEDM*, Dec 2011.
[5] S. Maeda *et al.*, "Impact of A Vertical Φ-shape transistor (VΦT) Cell for 1 Gbit DRAM and Beyond," *IEEE TED*, Dec 1995.
[6] D. Hisamoto *et al.*, "FinFET-A Self-Aligned Double-Gate MOSFET Scalable to 20 nm," *IEEE TED*, Dec 2000.
[7] L. Chang *et al.*, "Extremely Scaled Silicon Nano-CMOS Devices," *Proc. of the IEEE*, Nov 2003.
[8] K.-M. Persson *et al.*, "Extrinsic and Intrinsic Performance of Vertical InAs Nanowire MOSFETs on Si Substrates," *IEEE TED*, Sept 2013.
[9] Z. Chen *et al.*, "Demonstration of Tunneling FETs Based on Highly Scalable Vertical Silicon Nanowires," *IEEE Electron Device Letters*, July 2009.
[10] Y. Sun *et al.*, "Vertical-Si-Nanowire-Based Nonvolatile Memory Devices With Improved Performance and Reduced Process Complexity," *IEEE TED*, May 2011.
[11] S. Maheshwaram, S. K. Manhas, G. Kaushal, B. Anand, and N. Singh, "Device Circuit Co-Design Issues in Vertical Nanowire CMOS Platform," *IEEE Electron Device Letters*, July 2012.
[12] H. Liu, S. Datta, and V. Narayanan, "Steep Switching Tunnel FET: A Promise to Extend the Energy Efficient Roadmap for Post-CMOS Digital and Analog/RF Applications," in *ISLPED*, Sept 2013.
[13] W. Maly *et al.*, "Twin Gate, Vertical Slit FET (VeSFET) for Highly Periodic Layout and 3D Integration," in *MIXDES*, June 2011.
[14] X. Qiu, M. Marek-Sadowska, and W. Maly, "Characterizing VeSFET-Based ICs With CMOS-Oriented EDA Infrastructure," *IEEE TCAD*, April 2014.
[15] X. Wang *et al.*, "Highly Compact 1T-1R Architecture (4F2 footprint) Involving Fully CMOS Compatible Vertical GAA Nano-Pillar Transistors and Oxide-Based RRAM Cells Exhibiting Excellent NVM Properties and Ultra-Low Power Operation," in *IEEE IEDM*, Dec 2012.
[16] S. Maheshwaram, S. Manhas, G. Kaushal, B. Anand, and N. Singh, "Vertical Nanowire CMOS Parasitic Modeling and its Performance Analysis," *IEEE TED*, Sept 2013.
[17] L. Chang, M. Ieong, and M. Yang, "CMOS Circuit Performance Enhancement by Surface Orientation Optimization," *IEEE TED*, Oct 2004.
[18] M. M. A. Hakim *et al.*, "A Self-Aligned Silicidation Technology for Surround-Gate Vertical MOSFETS," in *ESSDERC*, Sept 2009.
[19] T. Agarwal, O. Badami, S. Ganguly, S. Mahapatra, and D. Saha, "Design Optimization of Gate-All-Around Vertical Nanowire Transistors for Future Memory Applications," in *EDSSC*, June 2013.
[20] S.-D. Suk *et al.*, "High Performance 5nm Radius Twin Silicon Nanowire MOSFET (TSNWFET) : Fabrication on Bulk Si Wafer, Characteristics, and Reliability," in *IEEE IEDM*, Dec 2005.
[21] S. Wang *et al.*, "Analytical Subthreshold Channel Potential Model of Asymmetric Gate Underlap Gate-All-Around MOSFET," in *EDSSC*, 2010.
[22] L. Zhang *et al.*, "Gate Underlap Design for Short Channel Effects Control in Cylindrical Gate-All-Around MOSFETs Based on An Analytical Model," *IETE Technical Review*, 2012.
[23] C.-Y. Hwang, Y.-C. Hsieh, Y.-L. Lin, and Y.-C. Hsu, "A Fast Transistor-Chaining Algorithm for CMOS Cell Layout," *IEEE TCAD*, Jul 1990.
[24] S. Wang *et al.*, "PROCEED: A Pareto Optimization-Based Circuit-Level Evaluator for Emerging Devices," *IEEE TVLSI*, 2015.
[25] R. Ghaida and P. Gupta, "DRE: A Framework for Early Co-Evaluation of Design Rules, Technology Choices, and Layout Methodologies," *IEEE TCAD*, Sept 2012.
[26] A. Mallik *et al.*, "TEASE: A Systematic Analysis Framework for Early Evaluation of FinFET-Based Advanced Technology Nodes," in *Proc. DAC*, May 2013.
[27] D. Yakimets *et al.*, "Lateral Versus Vertical Gate-All-Around FETs for Beyond 7nm Technologies," in *DRC*, June 2014.
[28] J. Roth, Charles H., *Fundamentals of Logic Design*. 5 ed., 2004.
[29] "Openaccess API." http://www.si2.org/.
[30] http://www.opencores.org/.
[31] Y. Du and M. Wong, "Optimization of Standard Cell Based Detailed Placement for 16 nm FinFET Process," in *Proc. DATE*, March 2014.
[32] R. Ghaida, Y. Badr, M. Gupta, N. Jin, and P. Gupta, "Comprehensive Die-Level Assessment of Design Rules and Layouts," in *Proc. ASP-DAC*, Jan 2014.
[33] Cadence SoC Encounter.
[34] T. Bao *et al.*, "Circuit and process co-design with vertical gate-all-around nanowire fet technology to extend cmos scaling for 5nm and beyond technologies," in *Solid State Device Research Conference (ESSDERC), 2014 44th European*, Sept 2014.