Optimal Transistor Folding and Placement for Synthesizing Standard Cells of Complementary FET Technology

Suwan Kim
Seoul National University
Seoul, South Korea
suwankim@snucad.snu.ac.kr

Taewhan Kim
Seoul National University
Seoul, South Korea
tkim@snucad.snu.ac.kr

ABSTRACT

As the VLSI technology continues to scale beyond 5nm, a strong demand on the continuing layout reduction of standard cells is required. However, the standard cells with conventional FinFET or nanosheet-FET structure are becoming much hard to meet this requirement due to the lateral P-FET and N-FET separation. It has been widely accepted that Complementary-FET (CFET) is a promising technology, which stacks P-FET on N-FET or vice versa, to achieve this objective. In comparison with synthesizing the conventional FET based standard cells, two prominent optimization tasks in CFET based multi-row cell synthesis that significantly affect the cell quality, in terms of area and routability, are (1) *determining transistor folding shapes* and (2) *determining placement order of transistors with fully secured vertical i.e., z-directional routing space* on the stacked FETs as well as buried power rail (BPR). In this work, we propose an optimal solution to the combined problem of tasks 1 and 2. Precisely, we *develop a search tree-based area-optimal method of transistor folding and placement*, in which we accelerate the cost computation of partial solutions by formulating it into dynamic programming while performing a strict feasibility checking of securing in-cell vertical routing space of partial solutions by formulating and solving it into an instance of network flow problem. In the meantime, through experiment with benchmark circuits, it is shown that the CFET cells produced by our cell synthesizer are 5% smaller in size on average even with 38% shorter total metal length and 70% less use of metal2 for in-cell routing over the cells produced by the recent state-of-the-art CFET cell generator.

ACM Reference Format:
Suwan Kim and Taewhan Kim. 2024. Optimal Transistor Folding and Placement for Synthesizing Standard Cells of Complementary FET Technology. In *61st ACM/IEEE Design Automation Conference (DAC '24), June 23–27, 2024, San Francisco, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649329.3658261

1 INTRODUCTION

FinFET [16] which has better controllability than planar MOSFET faces new scaling issue beyond sub-5nm technology node. For example, using fin depopulation technology [9] in FinFET can shrink the cell height by reducing the number of fins in a transistor while increasing fin height. However, such scaling came to the point at which a single fin per transistor could be used in 3nm node [13]. Nano-sheet FET (NSFET) [5, 10], which vertically stacks thin sheet-shaped channels into multiple layers, reduces the cell height further since no lateral spacing is required between nanosheets in NSFET. However, cell layout scaling is still limited due to the lateral spacing requirement between P-FET and N-FET devices.

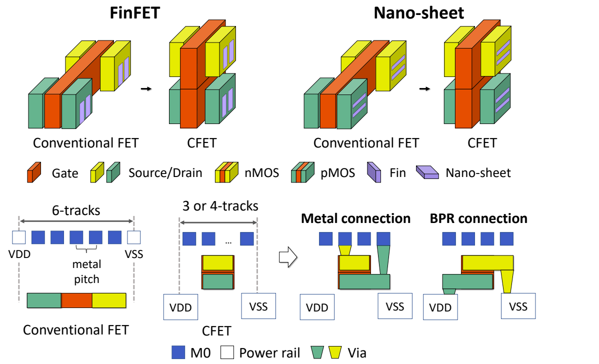


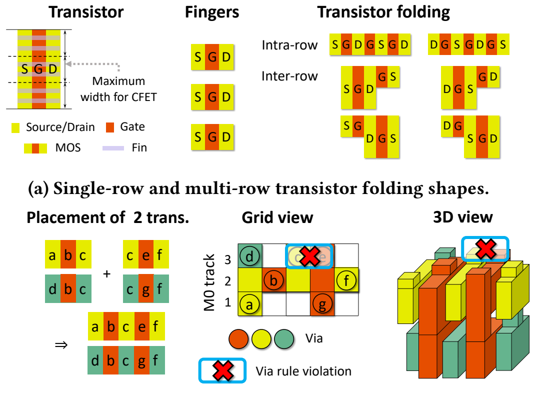
Figure 1: (Top) 3D architecture of FinFET, Nano-sheet and corresponding conventional FET, and CFET with N-FET stacked on P-FET vertically. (Bottom) Cross-sectional view of conventional FET and CFET.

In recent times, there has been a considerable interest in the groundbreaking technology known as the complementary field-effect transistor (CFET) [6, 12]. This technology features a unique structure with P-FET stacked on N-FET, or vice versa, leading to a significant reduction in area, as illustrated in Fig. 1.[1] However, the aggressive scaling along the lateral axis comes with a trade-off, diminishing the number of routing tracks available for CFET. For instance, as shown in Fig. 1, FinFET cells in 5nm node are typically designed with 5 metal tracks or more while CFET cells could be designed with 3 or 4 tracks [11]. To address this limitation, the adoption of buried power rails (BPR) [4], which provides the delivery of power signals from backside power delivery network (PDN), has become essential to increase vertical space for improved routability. (See Fig. 1.) In addition, since CFET facilitates the connection of inter-row signals within the same column i.e., y-direction on the same layer in standard cells, given that the signals are identical [15], there is a growing trend towards the implementation of a *multi-row* standard cell architecture in CFET cell layout generation [3, 15] to improve in-cell routability and relieve block-level routing congestion.

In comparison with synthesizing the conventional FET based standard cells, *two prominent optimization tasks in CFET based multi-row cell synthesis* that significantly affect the cell quality, in terms

[1] The term CFET refers to the technology of stacking transistors on top of each other. Either fin or nanosheet technology can be used to implement the individual transistors in CFET. Without loss of generality, CFET in this work is assumed N-FET being on top of P-FET.

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).
DAC '24, June 23–27, 2024, San Francisco, CA, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0601-1/24/06.
https://doi.org/10.1145/3649329.3658261



(a) Single-row and multi-row transistor folding shapes.

(b) Transistor placement with infeasible in-cell routability.

Figure 2: Examples of transistor folding and placement in the synthesis of CFET based standard cells.

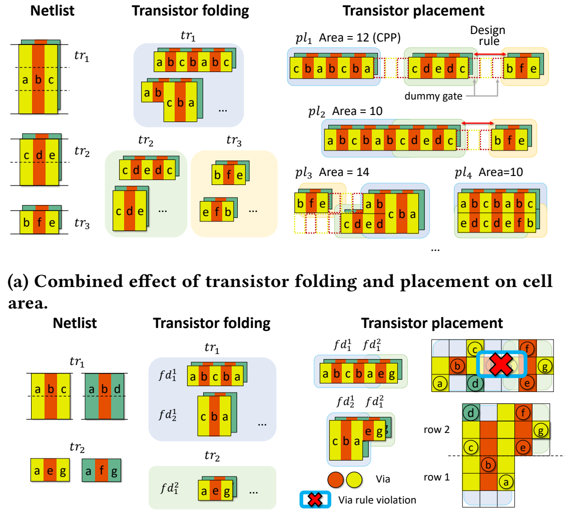
of area and routability with fully secured vertical i.e., z-directional routing space on the stacked FETs as well as BPR, are (1) *transistor folding* and (2) *transistor placement*.

Transistor folding is the process of dividing transistors into smaller units, known as *fingers*, such that the maximum width of each finger should not be longer than a specified width threshold. The increased complexity in determining the shapes of transistor folding arises from the fact that the folded fingers of individual transistors in designing a CFET standard cell should be placed on the *multi-rows* of the cell. In multi-row transistor folding, as illustrated in Fig. 2(a), the combined shapes of inter-row and intra-row fingers must be determined. This composition consideration of inter/intra-row fingers significantly amplifies the complexity of synthesizing CFET based standard cells compared to the single-row transistor folding in the conventional FET based cell synthesis.

Another noteworthy challenge in CFET cell synthesis pertains to a comprehensive *transistor placement*, in which the limited number ($\leq$ 4) of metal tracks poses a risk of inappropriate transistor ordering that makes in-cell net routing infeasible, as illustrated in Fig. 2(b) in which by stacking N-FET (i.e., nMOS) on P-FET (i.e., pMOS), the route of signal c is obstructed by the route of signal e. Such in-cell routability obstruction is mostly exacerbated by via rule (VR), which reserves one gate poly or source/drain column of the track contacting a via, rendering the adjacent left and right two columns inaccessible.

To our knowledge, a couple of approaches [2, 3] been proposed to automatically synthesize CFET based standard cells. Their approaches leverage Satisfiability Modulo Theory (SMT) to systematically design CFET based standard cell layouts, taking into account both transistor placement and in-cell routing in a holistic manner. To guarantee in-cell routability, they introduce specific constraints i.e., minimum pin length (MPL) and minimum I/O pin opening (MPO). However, the holistic approaches face scalability issues for larger cells. On the other hand, the work in [15] manually synthesizes CFET standard cells in order to conduct a comprehensive analysis of the parasitic effects. Note that the prior works [2, 3, 15] have not exploited the diverse structures of CFET multi-row folding shape in their CFET cell synthesis.

In this work, we propose an area-optimal solution to the combined problem of (*sub-problem 1*) determining multi-row CFET folding shapes and (*sub-problem 2*) establishing placement order of transistors ensuring in-cell routability. The main contributions of our work are as follows.



(a) Combined effect of transistor folding and placement on cell area.

(b) Combined effect of transistor folding and placement on in-cell routability.

Figure 3: Examples illustrating how the transistor placement combined with transistor folding affects cell area and in-cell routability.

- To address sub-problem 1, we devise *a scheme of generating diverse shapes of multi-row transistor folding*. We leverage signal sharing among multi-row transistors to fully explore valid folding shapes.
- To tackle sub-problem 2, we *evaluate the in-cell vertical (z-direction) routing space of partial solutions* by formulating it into a graph search problem. Partial solutions with infeasible in-cell routability are pruned off to enhance the overall efficiency of the solution exploration process.
- We develop *a search-tree based framework to explore the solution candidates of the combined problem* of sub-problems 1 and 2, in which we formulate the incremental cost computation of partial solutions into an instance of dynamic programming to accelerate the search process.

2 MOTIVATIONS

We explain how the transistor folding and placement influence the cell quality through illustrative examples.

- **Effect of folding and placement on cell size**: Suppose that a CFET cell should be implemented with three transistor pairs tr_1, tr_2, and tr_3. The size and signal information of the transistors are shown on the left side in Fig. 3(a) and various CFET folding shapes of the transistors are shown on the middle in Fig. 3(a). Then, four transistor placement results pl_1, pl_2, pl_3, and pl_4 using the folding shapes are shown on the right side in Fig. 3(a) where pl_1 and pl_3 lead to cell area of 12 and 14, respectively, while pl_2 (single-row cell) and pl_4 (double-row cell) each leads to the minimum cell area of 10. This example clearly shows that a careful selection of folding shapes combined with placement order is essential to synthesize CFET cells of minimal area.
- **Effect of folding and placement on in-cell routability**: Suppose that two transistor pairs tr_1 and tr_2 are given in a netlist. The

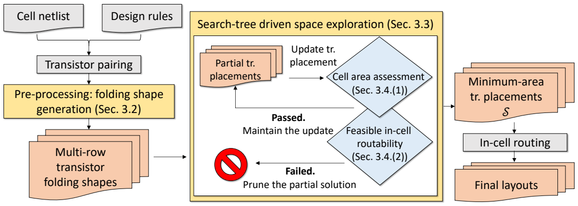

Cell netlist
Design rules
Transistor pairing
Pre-processing: folding shape generation (Sec. 3.2)
Multi-row transistor folding shapes
Search-tree driven space exploration (Sec. 3.3)
Partial tr. placements
Update tr. placement
Cell area assessment (Sec. 3.4.(1))
Passed.
Maintain the update
Feasible in-cell routability (Sec. 3.4.(2))
Failed.
Prune the partial solution
Minimum-area tr. placements $\mathcal{S}$
In-cell routing
Final layouts

Figure 4: Flow of our CFET cell synthesis framework CFET-fp **for simultaneous transistor folding and placement.**

transistor information is shown on the left side in Fig. 3(b). Then, a number of CFET folding shapes are shown on the middle in Fig. 3(b). Finally, two possible transistor placement results using the folding shapes are shown on the right side in Fig. 3(b), in which the lower CFET cell with placement order (tr_1, tr_2) and folding shapes fd_2^1 and fd_1^2 has no conflict on the via allocation for routing all signal nets while the upper CFET cell with placement order (tr_1, tr_2) and folding shapes fd_1^1 and fd_1^2 cannot avoid a conflict on via allocation.

3 SYNTHESIS OF CFET BASED MULTI-ROW STANDARD CELLS

3.1 The Overall Flow

The flow of our proposed CFET cell synthesis framework, called CFET-fp, for simultaneous transistor folding and placement is depicted in Fig. 4. CFET-fp takes, as inputs, the transistor-level netlist of target CMOS cell and design rules.[2]

Given the set of paired transistors[3], CFET-fp generates, as a pre-processing step in Sec. 3.2, all feasible multi-row folding shapes for every transistor pair by enumerating all combinations of n/pMOS folding shapes while utilizing inter-row signal sharing when stacking nMOS on top of pMOS. Then, in the main step in Sec. 3.3, CFET-fp performs a search-tree based transistor placement by exploring the feasible folding shapes of paired transistors in the course of expanding the search-tree branches. We incrementally and efficiently compute the folding shape cost whenever a new branch is expanded (i.e., the partial placement appending the folding shape) by formulating it into dynamic programming, based on which the resulting partial placement is assessed in terms of cell area and in-cell routability. The specific criteria for each assessment are detailed in Sec. 3.4. If the partial placement corresponding to the branch of search-tree expansion meets the criteria, the partial solution is maintained. Otherwise, the branch of partial placement is pruned off in the search-tree.

The outcome of the search-tree driven space exploration in our CFET-fp is the set $\mathcal{S}$ of multi-row transistor folding and placements of minimum area. Then, we apply an in-cell CFET router to each of placements in $\mathcal{S}$ and collect the cells with in-cell route completion.[4]

3.2 Generation of Feasible CFET Folding Shapes

Let $\#R_{CFET}$ and W_{CFET} be the CFET cell height in terms of the number of rows and the diffusion width on CFET cell, respectively. Our folding shape generation is performed in three steps.

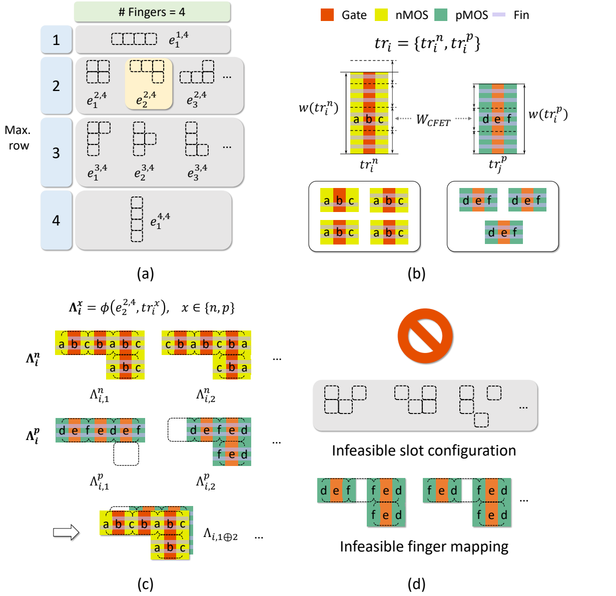


Figure 5: The steps of generating folding shapes. (a) Generating feasible slot configurations. (b) Generating CFET channel-width constrained fingers. (c) Generating feasible CFET folding shapes. (d) Infeasible slot and folding shapes.

- **Step 1** (*Generating finger slot configurations*): Due to the short diffusion width in CFET, each of initial n/pMOS transistor pairs is folded into multiple fingers, so that the fingers should be arranged in one or multiple rows to form a CFET (stacked) folding shape. In this step, we generate all feasible finger slot configurations such that the fingers should be placed on the slots without overlap. For example, Fig. 5(a) lists the finger slot configurations that can accommodate *at most* 4 fingers in r rows where r = 1, 2, 3, and 4 ($= \#R_{CFET}$), in which $e_j^{r,k}$ indicates the j^{th} slot configuration that can place at most k fingers in r rows. Note that infeasible slot configurations are those in which some slots are isolated, as shown in Fig. 5(d).
- **Step 2** (*Generating CFET channel-width constrained fingers*): For a transistor pair tr_i composed of nMOS transistor tr_i^n and pMOS transistor tr_i^p, let $w(tr_i^n)$ and $w(tr_i^p)$ respectively be channel width of tr_i^n and tr_i^p. Then, we generate $max(\lceil w(tr_i^x)/W_{CFET} \rceil)$ number of fingers for tr_i^x, $x = n$ and p, where W_{CFET} refers to a maximum width for CFET. For example, Fig. 5(b) shows the set of fingers for each of n/pMOS transistors.
- **Step 3** (*Generating feasible CFET folding shapes*): We introduce a mapping function $\phi(e_j^{r,k}, tr_i^x)$, $x = p$ or n, which produces the set, $\boldsymbol{\Lambda}_i^x$, of all feasible binding results for the fingers of tr_i^x obtained in Step 2 to the slot configuration $e_j^{r,k}$ prepared in Step 1. For example, the top two rows in Fig. 5(c) show a number of folding shapes in $\boldsymbol{\Lambda}_i^n = \{\Lambda_{i,1}^n, \Lambda_{i,2}^n, \cdots\}$ and $\boldsymbol{\Lambda}_i^p = \{\Lambda_{i,1}^p, \Lambda_{i,2}^p, \cdots\}$ for the slot configuration $e_2^{2,4}$. Note that infeasible mapping results are those in which some inner slots are empty, as shown in Fig. 5(d). Then, we produce all vertical stacking (i.e., CFET) folding shapes for transistor pair tr_i, which correspond to the evaluation of $\boldsymbol{\Lambda}_i^n \times \boldsymbol{\Lambda}_i^p$. For example, the shape labeled $\Lambda_{i,1\oplus 2}$ shown at the bottom in Fig. 5(c) indicates the CFET folding shape produced by stacking $\Lambda_{i,1}^n$ on $\Lambda_{i,2}^p$.

[2] Design rules include diffusion breaking/width/jog constraints, MAR, VR, EOL, PRL, and SHR outlined in [1–3].

[3] Transistor pairing involves either pairing pMOS and nMOS transistors sharing a poly gate or pairing pMOS and nMOS with cross-coupled relationship in the input transistor-level netlist.

[4] We implemented an SMT-based in-cell router and used it in our experiments.

3.3 Search-tree Driven Space Exploration

CFET-fp uses a search-tree based exploration engine to find optimal solutions of transistor folding and placement. Fig. 6(a) shows a conceptual view of space exploration. It explores all permutations of transistor pairs $tr_1, tr_2, tr_3, \ldots$, in which optimal partial solutions are determined through dynamic programming as follows.

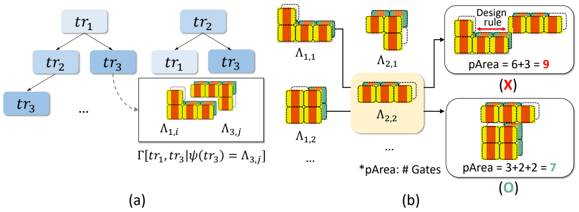


Figure 6: Search-tree based transistor folding and placement. (a) A conceptual view of space exploration by CFET-fp. (b) Expanding transistor placement considering folding shapes.

Let $\psi(tr_i)$ be a function that maps transistor pair tr_i to a CFET folding shape in set $\mathbf{\Lambda}_i$ $(= \mathbf{\Lambda}_i^n \times \mathbf{\Lambda}_i^p)$ and $\Gamma[tr_{i_1}, tr_{i_2}, \ldots, tr_{i_k}|\psi(tr_{i_k}) = \Lambda_{i_k,j}]$ be the area-minimal folding and placement such that (1) the placement order is $tr_{i_1}, \ldots, tr_{i_k}$ and (2) $\psi(tr_{i_k})$ is bound to folding shape $\Lambda_{i_k,j}$. Then, the recurrence formulation in our dynamic programming can be described as:

$$\begin{aligned} &pArea(\Gamma[tr_{i_1}, \ldots, tr_{i_k}|\psi(tr_{i_k}) = \Lambda_{i_k,j}]) \\ &= \min_{l}\{pArea(\Gamma[tr_{i_1}, \ldots, tr_{i_{k-1}}|\psi(tr_{i_{k-1}}) = \Lambda_{i_{k-1},l}] + \Lambda_{i_k,j})\} \end{aligned} \quad (1)$$

where $pArea(\Gamma[\cdot])$ and $pArea(\Gamma[\cdot]+\Lambda_{i_k,j})$ denote the intermediate area of partial placement $\Gamma[\cdot]$ and partial placement $\Gamma[\cdot]$ with CFET folding shape $\Lambda_{i_k,j}$ appended, respectively. As illustrated in Fig. 6(b), the intermediate area is computed by the sum of the number of columns of each row in the partial placement.

Note that our search-tree based space exploration is performed in the context of *multi-row* transistor folding and placement. For a partial solution $\Gamma[\cdot]$, we explore all possible (right-side) abutments of $\Lambda_{i_k,-}$ on $\Gamma[\cdot]$. For example, as shown in Fig. 6(b), when abutting $\Lambda_{2,2}$ on $\Lambda_{1,1}$, we systematically evaluate various placement combinations such as placing $\Lambda_{2,2}$ on top of $\Lambda_{1,1}$, on the first row of $\Lambda_{1,1}$, on the second row, and so forth while maintaining the placement order. Throughout this exploration, we adhere to transistor placement rules including the constraints of diffusion break and diffusion jog to ensure the valid solutions in transistor folding and placement.

3.4 Pruning Techniques

• **Pruning partial solutions based on the area lower bound**: We describe our pruning technique using the example in Fig. 7, in which it is assumed the input CMOS netlist consists of four transistor pairs tr_1, tr_2, tr_3, and tr_4. Through the search-tree traversal shown in Fig. 7(a), transistor placement (tr_1, tr_3, tr_2, tr_4) with the folding shapes in Fig. 7(a) produces the minimum cell area, which is $A_{min}^{so\_far}$ = 12.

Then, we expand the search-tree node, labeled tr_4 in Fig. 7(b), by attaching a folding shape, $\Lambda_{1,1}$ or $\Lambda_{1,2}$ in $\mathbf{\Lambda}_1$, of tr_1 as shown in Fig. 7(b). We compute the cell area lower bound, A_{lbound}, of a current partial solution, $\Gamma[\cdot]$, of folding and placement by:

$$A_{lbound} = A(\Gamma[\cdot]) + \sum_{tr_j \in \mathcal{T}_{rest}} A_{min}(tr_j) \quad (2)$$

where $\mathcal{T}_{rest}$ is the set of remaining transistor pairs that have not been traversed yet and $A_{min}(tr_j)$ is the smallest size of folding shapes in $\mathbf{\Lambda}_\mathbf{j}$.

For example, in Fig. 7(b), we keep the partial placement on the top since its area lower bound $A_{lbound} \le A_{min}^{so\_far}$, while the partial placement on the bottom is pruned since its $A_{lbound} = 14 > A_{min}^{so\_far}$.[5]

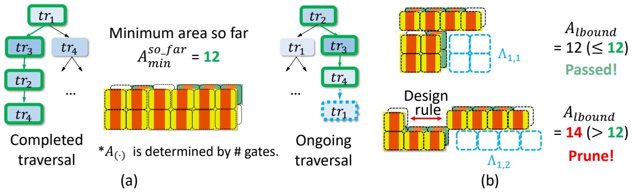


Figure 7: Pruning partial solutions by comparing their lower bound on area with the minimum area of solutions found.

• **Pruning partial solutions based on the feasibility of in-cell routing**: Fig. 8(a) shows the metal track availability configurations, in the left-to-right side view by column-wise cutting a CFET cell, for various cases of connecting signals and VDD/VSS source. For example, for *Case 1* in Fig. 8(a), the nMOS and pMOS in the column both require to be connected to a common signal *sigA*. This can be accomplished by connecting the nMOS and pMOS together with via, by which any of the four tracks is available for connecting to *sigA*. On the other hand, for *Case 3*, nMOS and pMOS in the column respectively require to be connected to *sigA* and *sigB*. Thus, either track 1 or 4 can be used to connect *sigB* to pMOS. Then, with the via-enclosure constraint (e.g., Fig. 8(b)), as indicated by the *available track pair* in Fig. 8(a), either any of tracks 1 and 2, if track 4 were used for pMOS to connect to *sigB*, or any of tracks 3 and 4, if track 1 were used for pMOS, can be used for nMOS to connect to *sigA*.

We then describe how our CFET-fp checks the connection feasibility by using the example in Figs. 8(b) and (c). Fig. 8(b) shows the bird-eye view (in 2D matrix form) of a CFET folding shape of transistor in which the rows indicate metal track while the red and yellow/green columns indicate poly gate and source/drain diffusion, respectively.

Suppose that the five columns, from the leftmost to right, in Fig. 8(b) exhibit *Cases* 4, 3, 3, 1, 5 of metal track availability configuration. Then, the feasible connectivity can be checked by constructing a column-to-column connecting network G with source s and sink t, as illustrated in Fig. 8(b). For example, the matrix labeled p_1^2 indicates the signal connection corresponding to (2,4) in *Case 3* of the second column in Fig. 8(b), which can be abutted to the signal connection corresponding to (3,1) in *Case 3* of the third column, but cannot be abutted to the signal connection of (-,4) in *Case 4* in the first column since via rule violation occurs if they were abutted each other. It is said that in-cell routing on a folding shape is *feasible* if there is a path in G from s to t.

[5] In experiment, we used *area relaxation paramter* γ in $A_{lbound} \le A_{min}^{so\_far}(1+\gamma)$. In case in-cell routing is incomplete for every cell with γ=0.0, we increase the γ value.

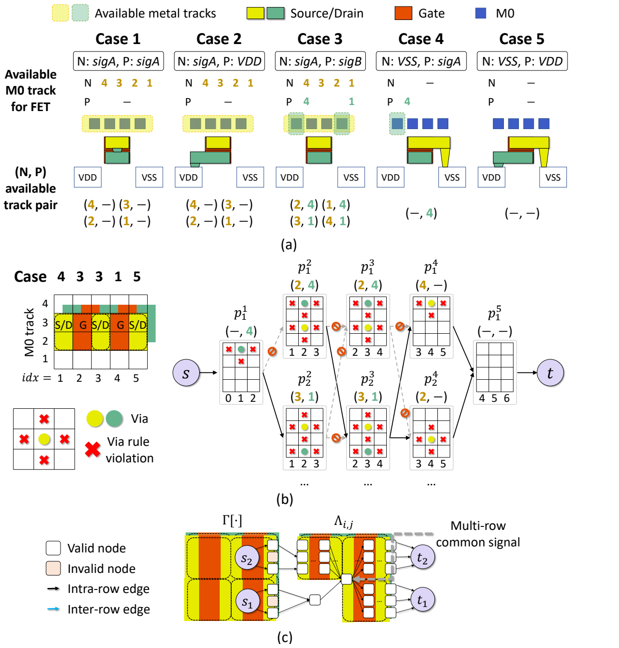


Figure 8: A conceptual view of checking the feasibility of in-cell routing. (a) Metal track availability configurations. (b) Matrix representation of metal tracks and CFET columns for a folding shape and network generation for feasibility checking. (c) Feasibility checking for appending a folding shape to a partial solution.

The checking of the feasibility of appending a folding shape of new transistor at the end of a partial solution of placement and folding can be done similarly by extending the notion described the connection feasibility checking on folding shape. That is, as shown in Fig. 8(c), if there are paths from s_1 to t_1 and from s_2 to t_2, the folding shape $\Lambda_{i,j}$ can be appended to the partial solution $\Gamma[\cdot]$ with no connection conflict.

4 EXPERIMENTAL RESULTS

Our CFET cell generator CFET-fp was implemented in C++ on a linux machine combined with Intel i7-8700k 4.7GHz CPU and 64GB memory. Note that CFET-fp ensures the generation of transistor folding and placement of minimum area. The standard cells produced by our CFET-fp are available for access on GitHub.[6]

To seamlessly link in-cell routing to CFET-fp, we implemented the SMT-based router introduced in [2, 3] and used the single-threaded Z3 SMT solver (Version 4.8.11) [7]. The SMT-based in-cell router, which has been configured with the multi-objective of lexicographically reducing the usage of metals M2, M1, and M0, was applied to each solution of transistor folding and placement of minimal area.

As input netlists for synthesizing CFET cells, we used the SPICE netlists of the ASAP 7nm conventional standard cells in [8]. To be a fair comparison, in experiments we varied the design rules (DRs) and minimum pin opening (MPO) settings in accordance with that used in each of the prior CFET generators.

- **Comparison of the CFET cells**: Table 1 shows a comparison of quality metrics of the cells produced by the state-of-the-art prior CFET cell generators [2] (SMT based single-row CFET cell generator), [3] (SMT based multi-row CFET cell generator), and by our CFET-fp. The DRs are set to conform to the specifications outlined in the prior works.[7]

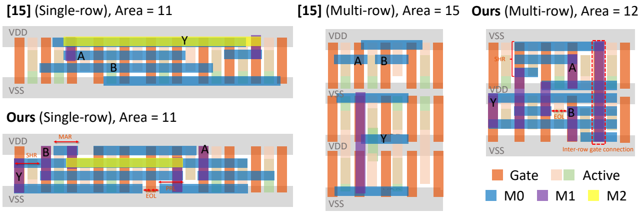

[15] (Single-row), Area = 11
[15] (Multi-row), Area = 15
Ours (Multi-row), Area = 12
Ours (Single-row), Area = 11
VDD
VSS
Y
A
B
MAR
SHR
EOL
Inter-row gate connection
Gate
Active
M0
M1
M2

Figure 9: Layout comparison of the handcrafted CFET cells in [15] and the CFET cells synthesized by our CFET-fp for netlist XNORx1.

In summary, CFET-fp produces CFET cells with 5% smaller cell size over that produced by [3] and cells of the same size as those generated by [2]. Our proposed transistor folding and placement methods demonstrate their effectiveness through a metal length reduction of 38%, as compared to the multi-row CFET cell layout generator [3]. In addition, in the context of single-row configurations, our suggested approach exhibits a 7% improvement in performance compared to the method presented in [2]. Noticeably, CFET-fp demonstrates a considerable runtime improvement compared to the prior CFET generators, validating the effectiveness of our acceleration and pruning techniques in CFET-fp. One unique and very useful merit of CFET-fp is that it is able to generate *multiple CFET cells of diverse layouts with minimal area*, as shown in the last column of the table. Though we perform in-cell routing for each of the multiple cell candidates, the total runtime, as shown in column Ours (all), is merely 2.0% of the runtime spent by [3]. Moreover, with the multiple CFET cell candidates, the designers can select the cells that are best suited for their design objectives and constraints.

CFET-fp also effectively generates large cells, as summarized in Table. 2.[8] On average, the production time for each large CFET cell is 1876 seconds in a multi-row setting and 323 seconds in a single-row setting, both achieving comparable quality.

Finally, Fig. 9 shows the layout comparison of the CFET cells in [15] which was produced manually, and the cells produced by our CFET-fp. [9] It is shown that the cells synthesized by CFET-fp consistently exhibit comparable quality with the handcraft cells in [15], demonstrating even smaller area for multi-row setting.

- **Chip design quality exploration through DTCO**: We conduct the chip implementation process through design and technology co-optimization (DTCO) by varying the number of in-cell routing tracks (2.5T and 3.5T) by which we iteratively implement circuit WB_DMA [14], to assess the DTCO effectiveness using CFET-fp on the chip quality exploration. The utilization beyond 0.78 and 0.9 for 2.5T and 3.5T is not feasible due to the insufficient number of routing tracks, as shown in Fig. 10. Although configuring a fewer number of in-cell routing tracks in the design kit (DK) offers advantages in terms of area and wirelength, it introduces challenges related to design feasibility in chip routing.

[6] https://github.com/SNU-suwan/CFET_fp

[7] MAR/EOL/VR/PRL/SHR/MPL = 1/1/0/1/2/1 with P-on-N 2.5T structure for comparison with [3] and = 1/2/1/1/2/2 with N-on-P 4.5T for comparison with [2].

[8] No CFET cell synthesis results for the large netlists are available in the literature.

[9] Note that ours utilizes more metals to guarantee the minimum length of the M1 pin, whereas [15] utilizes the M0 pin.

Table 1: Comparison of the CFET cells generated by the single-row CFET cell generator [2], the multi-row CFET cell generator [3], and our CFET-fp.

Cell netlist			Cell width (#CPP)				#Cell row		Metal length†				Usage of M2 tracks				Runtime (sec.)						# Synthesized cells			
			Multi-row		Single-row				Multi-row		Single-row		Multi-row		Single-row		Multi-row			Single-row			Multi-row		Single-row	
Name	#FET	#Net	[3]	Ours	[2]	Ours	[3]	Ours	[3]	Ours	[2]	Ours	[3]	Ours	[2]	Ours	[3]	Ours (1 cell)	Ours (all)	[2]	Ours (1 cell)	Ours (all)	[3]	Ours	[2]	Ours
AND2x2	6	7	6	6	6	6	1	1	85	53	60	56	1	0	0	0	8.65	0.29	0.29	8.12	0.45	0.45	1	1	1	1
AND3x1	8	9	7	6	6	6	1	1	108	59	68	63	2	0	0	0	50.44	0.91	0.91	30.09	1.40	1.40	1	1	1	1
AND3x2	8	9	8	7	7	7	1	1	106	65	76	69	1	0	0	0	50.52	1.44	1.44	28.35	1.64	1.64	1	1	1	1
AOI21x1	6	8	6	5	9	9	2	2	162	75	142	122	2	0	0	0	3575.33	0.79	1.39	119.69	6.25	70.36	1	2	1	10
AOI22x1	8	10	7	6	11	11	2	2	226	97	240	203	2	0	1	1	6808.18	2.63	14.43	363.16	20.47	42.71	1	5	1	2
BUFx2	4	5	5	5	5	5	1	1	60	37	40	39	1	0	0	0	4.22	0.10	0.10	4.92	0.21	0.21	1	1	1	1
BUFx3	4	5	6	6	6	6	1	1	70	51	53	56	1	0	0	0	5.49	0.18	0.18	11.2	0.58	0.58	1	1	1	1
BUFx4	4	5	7	7	7	7	1	1	77	57	59	62	1	0	0	0	8.75	0.24	0.24	7.91	0.77	0.77	1	1	1	1
BUFx8	4	5	12	12	12	12	1	1	123	103	105	107	1	0	0	0	48.64	0.95	0.95	43.65	4.36	4.36	1	1	1	1
DFFHQN	24	17	9	9	16	15	2	2	206	256	182	203	0	2	0	1	21071.8	915.39	1026.77	6831.77	410.27	1359.70	1	8	1	10
FA	24	17	8	7	14	14	2	3	288	164	379	276	4	0	2	3	24394.2	1281.33	1281.33	6653.07	484.55	1568.44	1	1	1	9
INVx1	2	4	3	3	3	3	1	1	44	20	23	22	2	0	0	0	1.48	0.02	0.11	0.49	0.06	0.29	1	4	1	4
INVx2	2	4	4	4	4	4	1	1	52	26	29	27	1	0	0	0	1.94	0.05	0.27	1.03	0.12	0.85	1	4	1	4
INVx4	2	4	6	6	6	6	1	1	70	46	48	47	1	0	0	0	457	0.16	0.72	3.46	0.60	2.79	1	4	1	4
INVx8	2	4	10	10	10	10	1	1	108	86	92	111	1	0	0	0	15.08	0.56	2.56	19.14	3.40	10.09	1	4	1	3
NAND2x1	4	6	6	6	6	6	1	1	103	108	74	70	2	2	0	0	12.76	0.52	1.33	15.88	1.33	11.13	1	3	1	8
NAND2x2	4	6	11	10	10	10	1	1	99	184	131	125	2	2	0	0	41.26	2.27	8.36	33.83	5.28	57.20	1	6	1	10
NAND3x1	6	8	14	7	11	11	1	2	283	175	146	147	2	2	0	0	887.56	5.16	34.34	124.11	12.19	101.95	1	7	1	8
NAND3x2	6	8	26	11	21	21	1	2	441	150	283	303	2	0	0	0	909.03	30.12	55.62	2869.53	81.86	609.56	1	3	1	8
NOR2x1	4	6	6	6	6	6	1	1	103	108	74	70	2	2	0	0	16.96	0.57	1.46	12.89	1.29	12.51	1	3	1	9
NOR2x2	4	6	11	10	10	10	1	1	199	197	131	119	2	2	0	0	44.37	2.73	5.58	27.94	4.99	52.44	1	3	1	10
NOR3x1	6	8	14	7	11	11	1	2	283	178	156	163	2	2	0	0	914.35	9.52	46.44	52.33	11.07	108.84	1	7	1	10
NOR3x2	6	8	26	11	21	21	1	2	510	176	286	320	2	0	0	1	1027.98	34.81	54.86	1897.53	82.25	373.63	1	2	1	5
OAI21x1	6	8	6	5	9	9	2	2	150	85	149	131	2	0	0	0	2122.94	1.07	2.95	52.52	7.93	62.86	1	3	1	8
OAI22x1	8	10	7	6	11	11	2	2	229	111	255	174	2	0	1	0	7043.85	3.96	25.65	612.6	19.34	167.27	1	6	1	8
OR2x2	6	8	6	6	6	6	1	1	85	53	60	56	1	0	0	0	14.22	0.52	0.52	12.99	0.50	0.50	1	1	1	1
OR3x1	8	9	6	6	6	6	1	1	72	59	68	63	2	0	0	0	73.58	1.96	1.96	76.77	1.54	1.54	1	1	1	1
OR3x2	8	9	7	7	7	7	1	1	106	65	76	69	2	0	0	0	95.94	2.42	2.42	89.22	1.65	1.65	1	1	1	1
XNOR2x1	10	9	7	7	11	11	2	2	268	116	223	180	2	0	1	1	5766.47	58.11	99.17	977	47.47	87.45	1	10	1	3
XOR2x1	10	9	7	7	11	11	2	2	256	121	213	175	2	1	1	1	2122.94	63.66	95.70	134.86	50.37	94.69	1	8	1	4
Avg.	6.80	7.70	8.80	7.03	9.30	9.27	1.27	1.43	165.73	102.70	130.70	120.93	1.67	0.50	0.20	0.27	2586.53	45.29	56.81	703.87	42.14	160.26	1.00	3.43	1.00	4.90
Norm.			1.00 ‡	0.95	1.00	1.00	1.00	1.13	1.00	0.62	1.00	0.93	1.00	0.30	1.00	1.33	1.00	0.02	0.02	1.00	0.06	0.23	1.00	3.43	1.00	4.90

† The metal length is calculated by the weighted sum of the via and metal grid as [2, 3] did, in which the weighting of via is 4× metal grid considering the parasitic resistance.

‡ The normalized value for the size of a multi-row cell is calculated by multiplying the cell width by the number of rows of the cell.

Table 2: Synthesized results by CFET-fp for large netlists. (Note that none of the prior CFET cell synthesizers ([2, 3]) have reported the CFET cell layout data on those large netlists.)

Cell netlist			Cell width		# Cell row	Metal length		#M2 tracks		Runtime (sec.)				#Syn. cells	
Name	#FET	#Net	MR	SR	(MR only)	MR	SR	MR	SR	MR (1 cell)	MR (all)	SR (1 cell)	SR (all)	MR	SR
DHLx1	16	13	7	11	2	156	117	1	0	5719	5729	441	492	2	10
DHLx2	16	13	7	12	2	154	117	1	0	247	247	121	184	1	10
DHLx3	16	13	8	13	2	149	146	1	0	266	278	131	245	4	8
DFFHQNx2	24	17	9	16	2	223	212	1	1	1549	1682	420	869	8	10
DFFHQNx3	24	17	10	17	2	247	227	1	1	1599	1949	502	2490	7	9
Avg.	19.2	14.6	8.2	13.8	2	185.8	163.8	1	0.4	1876	1977	323	856	4.4	9.4

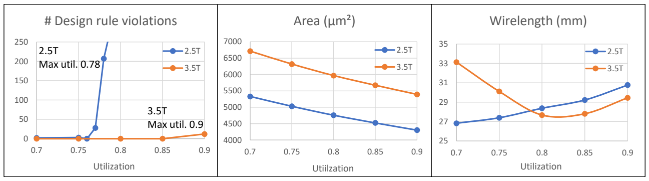


Figure 10: Design quality exploration for WB_DMA through DTCO by using CFET-fp, sweeping in-cell routing tracks (2.5T, 3.5T)

5 CONCLUSION

In comparison with the conventional non-CFET cell synthesis, synthesizing CFET cells involves much more complication due to diverse 3D folding shapes and new wiring constraint inside CFET as well as the necessity of multi-row cell layouts. In this respect, this work proposed an optimal method of simultaneous multi-row transistor folding and placement in synthesizing CFET standard cells while considering the constraint of CFET in-cell routing feasibility. Through experiments with ASAP 7nm netlists and PDK, it was shown that our method outperformed the state-of-the-art CFET cell synthesizers performing simultaneous transistor placement and in-cell routing, but no folding shape exploration.

Acknowledgements: This work was supported by Samsung Electronics Company, Ltd. (IO201216-08205-01, and IO221227-04376-01), Samsung Advanced Institute of Technology (IO230223-05124-01), National Research Foundation grant by Korea Government (2021-R1A2C2008864), Institute of Information and communications Technology Planning and Evaluation (IITP) grant by Korea government (2021-0-00754), Software Systems for AI Semiconductor Design and Artificial Intelligence Semiconductor Support Program to nurture the best talents (IITP-2023-RS-2023- 00256081) grant by Korea government, National R&D Program through NRF by Ministry of Science and ICT (2020M3H2A1078119), and BK21 Four Program of Education and Research Program for Future ICT Pioneers, Seoul National University. The EDA tool was supported by IC Design Education.

REFERENCES

[1] K. Baek and T. Kim. 2021. Simultaneous transistor folding and placement in standard cell layout synthesis. In *IEEE/ACM ICCAD*.

[2] C. K. Cheng et al. 2021. Complementary-FET (CFET) standard cell synthesis framework for design and system technology co-optimization using SMT. *IEEE TVLSI* (2021).

[3] C. K. Cheng et al. 2021. Multirow complementary-FET (CFET) standard cell synthesis framework using satisfiability modulo theories (SMTs). *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits* (2021).

[4] D. Prasad et al. 2019. Buried power rails and back-side power grids: Arm® cpu power delivery network design beyond 5nm. In *IEEE IEDM*.

[5] G. Bae et al. 2018. 3nm GAA technology featuring multi-bridge-channel FET for low power and high performance applications. In *IEEE IEDM*.

[6] J. Ryckaert et al. 2018. The complementary FET (CFET) for CMOS scaling beyond N3. In *IEEE Symposium on VLSI Technology*.

[7] L. D Moura et al. 2008. Z3: An efficient SMT solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*.

[8] L. T. Clark et al. 2016. ASAP7: A 7-nm finFET predictive process design kit. *Microelectronics Journal* (2016).

[9] M. G. Bardon et al. 2015. Dimensioning for power and performance under 10nm: The limits of FinFETs scaling. In *IEEE ICICDT*.

[10] N. Loubet et al. 2017. Stacked nanosheet gate-all-around transistor to enable scaling beyond FinFET. In *IEEE Symposium on VLSI Technology*.

[11] S. M. Y. Sherazi et al. 2019. CFET standard-cell design down to 3-track height for node 3nm and below. In *SPIE Advanced Lithography*.

[12] S. Subramanian et al. 2020. First monolithic integration of 3d complementary fet (cfet) on 300mm wafers. In *IEEE Symposium on VLSI Technology*.

[13] IEEE. 2018. International Roadmap for Device and Systems. https://irds.ieee.org

[14] Oliscience. [n. d.]. OpenCores. https://opencores.org

[15] E. Park and T. Song. 2022. Complementary FET (CFET) standard cell design for low parasitics and its impact on VLSI prediction at 3-nm process. In *IEEE TVLSI*.

[16] H.-S. P. Wong. 2005. Beyond the conventional transistor. In *Solid-State Electron*.