Late Breaking Results: Gear-Ratio-Aware SMT-Based Cell Synthesis for CFET

Anonymous Author(s)

1 ABSTRACT

Gear Ratio (GR) is a critical factor for addressing scaling constraints as it allows tight spacing for each layer. To support DTCO pathfinding and prepare for unexpected Metal Pitch and Offset requirements in future technologies, we have developed an SMT-based *Cell Layout Generation* flow featuring **Arbitrary Gear Ratio (AGR)**. Previous approaches to Cell Layout Generation support only uniform grids with 2:1 Gear Ratio. Here, we extend the prior work to encompass unseen GRs in the future, as well as to allow users to define any pitch and offset distance for DTCO explorations. Given the user-defined design rule parameters and complexity of conditional design rules, we also implement a comprehensive rule checking scheme and a lexicographic real-distance based multiple-objective optimization scheme, aiming to offer future-proofing capabilities.

2 INTRODUCTION

The continuous decrease in device footprint implies the need for a novel device architecture that maintains high efficiency while aiming size reduction. To fulfill this scaling requirement, we use 3D Standard Cell Layout, Complementary Field Effect Transistor (CFET)[2], which provides an architecture solution with vertically stacked *PMOS* and *NMOS* devices controlled by a common gate. In order to facilitate the implementation of conditional design rules, we introduce a Satisfiability Modulo Theories (SMT) based graph-based system in previous works[1][3]. Our graph system aims to be future-proofing against **Arbitrary Gear Ratio (AGR)** and fine-grained spacing rules[7]. This motivates us to implement a non-uniform grid graph system, accompanied by a set of distance-based cost functions to adapt to unpredictable Gear Ratio. This work demonstrates a new toolset with arbitrary gear ratios enablement and distance-based design rule encodings in the SMT-based cell generation formulation, leading to potential cell shrinkage and DTCO exploration. Our code is fully open-sourced on GitHub [1].

3 PRELIMINARIES

Offset: Given different pitches and cell width, each layer might not align perfectly at the start of a pitch, thus sacrificing footprint for alignment. Hence, multiple copies of the standard cells are generated with different layer offset to allow placers to take advantage of additional area shrinkage [4].

Gear Ratio (GR): We define gear ratio as the ratio between gate and vertical metal layer pitches in standard cells which is typically given in integers (or half-integers). In this work, we extend this definition with vertical metal layer and gate pitch having greatest common divider less then 5 called **Arbitrary Gear Ratio (AGR)**.

4 GRAPH BASED MAPPING

We define a N-layer virtual grid graph $G = (V, E)$ where $V = \{V^1, ..., V^n\}$ are the vertices, and $E = \{E^1, ..., E^n\}$ are the edges, $n \in N$. MP^n is the corresponding metal pitch on layer n. T_V^n is the number of vertical tracks on layer n. For T_V^1, it is the $\max(W_P, W_N)$, where W_P is the total width of *PMOS* devices and W_N is the total width of *NMOS* devices. Then the total width of the standard cell can be computed as $W_{\text{std}} = MP^1 \times T_V^1 + \epsilon^1$. W_{std} determines the number of vertical tracks (i.e. T_V^3) on each odd metal layers. We have a fixed number of horizontal tracks T_H on each layer which are induced by horizontal layers.

- On an even layer i, we have $T_V^i \le T_V^{i+1} + T_V^{i-1}$ and $|T_H \cdot T_V^i|$ vertices.
- On an odd layer j, we have $T_V^j = \left\lfloor \frac{W_{\text{std}} - \epsilon^n}{MP_j} \right\rfloor$ and $|T_H \cdot T_V^j|$ vertices.

Each vertex is characterized by *metal (layer)*, *row* and *column*, similar to the Cartesian coordinate system to indicate its position. Within the same layer, edges are created with respect to the metal directions between vertices to allow intra-layer connections. Inter-layer connections are allowed through **Vias** (also represented by *E*) if and only if the two vertices are on adjacent layers and have the same *row* and *column*. Much like a classical multi-commodity flow problem, our grid graph is a representation of all possible flow connections (edges), pin access and via access (vertices). We use the **Z3 Solver** [5] to minimize the track usage based on our objectives, flow constraints, and design rule constraints. Hence, not all edges and vertices will be used in the final layout. The final layout is encoded using a set of boolean variables associating metal directions, vertices and edges. **Figure 1** demonstrates an extreme case where MP_1 and MP_3 are set slightly causing almost no overlapping vertices. To maximize via access point, M2 grid graph (in green color) will create a vertex at each point that coincide with M1 and M3 columns. This becomes problematic because we can no longer assume adjacent vertices have uniform distance with respect to some metal pitch value. This motivates us to provide a more fine-grained design rule checking scheme in Section 5.

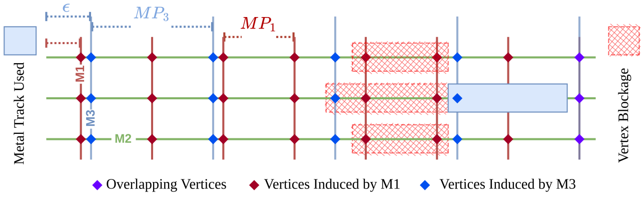

Figure 1: Top-down view of an uneven grid system induced by different metal pitch

5 DISTANCE-BASED DESIGN RULE CHECKING AND OBJECTIVE FUNCTIONS

The main challenges in our GR enablement are graph construction and design rule encoding. As shown in **Figure 1**, horizontal layers can have overlapping vertices and non-overlapping vertices with a small margin. The naive approach of using greatest common divider may impact runtime for vertices creation with *AGR*. The nonuniform distance also becomes problematic when trying to encode

[1] Anonymous for submission.

Table 1: Comparison Table of Metrics for CFET 4.5T Given Parameter in L1 (EOL=30/PRL=30/SHR=60/MAR=30)

Cell Name	#MOSFET	#Net	Cell Width (CPPs)		Metal Length		#Via		#M2 Tracks		Z3 Runtime(s)	
			CFET	CFET_AGR	CFET	CFET_AGR	CFET	CFET_AGR	CFET	CFET_AGR	CFET	CFET_AGR
BUFx2	4	5	4	4	792	**638**	14	**7**	0	0	2	19
AND2x2	6	7	6	6	**836**	914	10	10	0	0	4	55
AOI22x1	8	10	11	11	3036	**2508**	31	**28**	1	**0**	54	860
XNOR2x1	10	9	11	11	**2904**	4098	**30**	38	**1**	8	57	570
OAI22x1	8	10	11	11	3014	**2508**	32	**28**	1	**0**	60	2274
DFFHQNx1	24	17	15	15	**3168**	3883	26	26	0	0	6608	9518

design rules. For graph construction, we "jump" through columns only by different MP. For instance, we traverse along M2 twice: one for MP^1 and the other for MP^3 and merge the unique vertices together as shown in **Algorithm 1**. This allows us to avoid "minimal steps" (e.g. with greatest common divider) with arbitrary GR. We use map data structures and build multiple lookup tables for T_V, T_H, *column*, *row*, *column_index* and *row_index* for efficient design rule encoding. With GR-aware grid graph, design rule parameters $\Lambda = \{\lambda_{EOL}, \lambda_{MAR}...\}$ should be defined in Manhattan distance. Our flow incorporates design rules described in [3][7] for SADP-friendly designs (end-of-line spacing (EOL), step heights rule (SHR), parallel run length (PRL), minimum area (MAR), etc). While traversing through adjacent vertices and encode each forbidden access points in SMT constraints, we keep track of the distance traveled along the edges until the desired distance is reached. Additionally, we perform the lexicographic real-distance based multiple-objective optimization:

$$\textbf{Minimize}\begin{cases} \text{Placement (Cell Area)} & = \max(T_V^n \times T_H^n), n \in N \\ \text{Routability (\#Top Track)} & = \displaystyle\sum_{e_{u,v} \in E^{\text{Top}}} \boldsymbol{\delta}_{\text{Top}} \times m_e \times \textbf{dist}(e_{u,v}) \\ \text{Total Metal Length} & = \displaystyle\sum_{n \in N} \sum_{e_{u,v} \in E^{\text{n}}} m_e \times \textbf{dist}(e_{u,v}) + \boldsymbol{\delta}_{\text{via}} \times \boldsymbol{v_e} \end{cases}$$

where the distance function, dist, indicates the column/row distance for each edges; m_e is the indicator for metal usage; v_e is the indicator for via usage and δ is a user-determined multiplier for further penalizing M2 Track usage and via usage. This framework provides a robust graph traversal strategy because we allow the arbitrary numbers of access points while alleviate the burden of irregular MP. Disabling one access point will be as simple as deleting one entry in the lookup table, which becomes more future-proofing for unseen design rules.

6 EXPERIMENT

In this section, we analyze the potential of our AGR enabled framework with 2 : 1 in terms of cell width, metal length, and M2 track usage, Via usage and Z3 runtime in **Table 1**. Our experiment based on ASAP7 standard cell library [6] (for pin/net/instance information). *CFET* has gear ratio of 44:22 and *CFET_AGR* has gear ratio 44:10. Design rule parameters are given in Manhattan distance. Under these settings, we observe that *Cell Width* has no further shrinkage but in some cases, *Metal Length* and *#Via* are reduced. However, we also recognize that runtime for Z3 solver becomes a bottleneck of our toolset due to the increase in the number of vertices. For instance, the regular Gear Ratio has mostly overlapping vertices in the grid graph, where *AGR* creates vertices for every adjacent pitch distance as shown in **Figure 1**. This potentially doubles the variables and literals counts in the SMT formulation. For future work, we will aim more efficient rule encoding scheme and explore the benefit of different *AGR* and design rule settings.

Algorithm 1: Vertex Generation

```
for $metal \in \{M1...M^{top}\}$ do
    if $metal$ is horizontal then
        $step_{low} \leftarrow MP^{metal-1}$
        $step_{high} \leftarrow MP^{metal+1}$
        Iterate through $col$ and $row$ with $step_{low}$.
        Iterate through $col$ and $row$ with $step_{high}$.
        Merge unique vertices and store them into $V$.
    else if $metal$ is vertical then
        $step \leftarrow MP^{metal}$
        Iterate through $col$ and $row$ with $step$.
        Store vertices into $V$.
return $V$
```

REFERENCES

[1] C. Holtz D. Lee C. Cheng, C-T Ho and B. Lin. 2022. Machine Learning Prediction for Design and System Technology Co-Optimization Sensitivity Analysis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 30, 8 (2022), 1059–1072. https://doi.org/10.1109/TVLSI.2022.3172938

[2] D. Lee B. Lin C. Cheng, C-T Ho and D. Park. 2021. Complementary-FET (CFET) Standard Cell Synthesis Framework for Design and System Technology Co-Optimization Using SMT. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 29, 6 (2021), 1178–1191. https://doi.org/10.1109/TVLSI.2021.3065639

[3] C-T Ho I. Kang H. Kim S. Gao B. Lin C. Cheng D. Lee, D. Park. 2021. SP&R: SMT-Based Simultaneous Place-and-Route for Standard Cell Synthesis of Advanced Nodes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 40, 10 (2021), 2142–2155. https://doi.org/10.1109/TCAD.2020.3037885

[4] P. Gutwin X. Zhu L. Liebmann, V. Gerousis and J. Petykiewicz. 2017. Exploiting regularity: breakthroughs in sub-7nm place-and-route. In *Design-Process-Technology Co-optimization for Manufacturability XI*, L. Capodieci and J. P. Cain (Eds.), Vol. 10148. International Society for Optics and Photonics, SPIE, 101480F. https://doi.org/10.1117/12.2259981

[5] L. De Moura and N. Bjørner. 2008. Z3: An Efficient SMT Solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (Budapest, Hungary) *(TACAS'08/ETAPS'08)*. Springer-Verlag, Berlin, Heidelberg, 337–340.

[6] M. Vangala V. Vashishtha and L. Clark T. 2017. ASAP7 predictive design kit development and cell design technology co-optimization: Invited paper. In *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 992–998. https://doi.org/10.1109/ICCAD.2017.8203889

[7] H. Yoshida Y. Wang J. Kye Y. Ma, J. Sweis and H. J. Levinson. 2012. Self-aligned double patterning (SADP) compliant design flow. In *Design for Manufacturability through Design-Process Integration VI*, M. E. Mason (Ed.), Vol. 8327. International Society for Optics and Photonics, SPIE, 832706. https://doi.org/10.1117/12.917775