

SMT-based Standard Cell Layout Generator for Vertical Gate-All-Around FET (VFET)

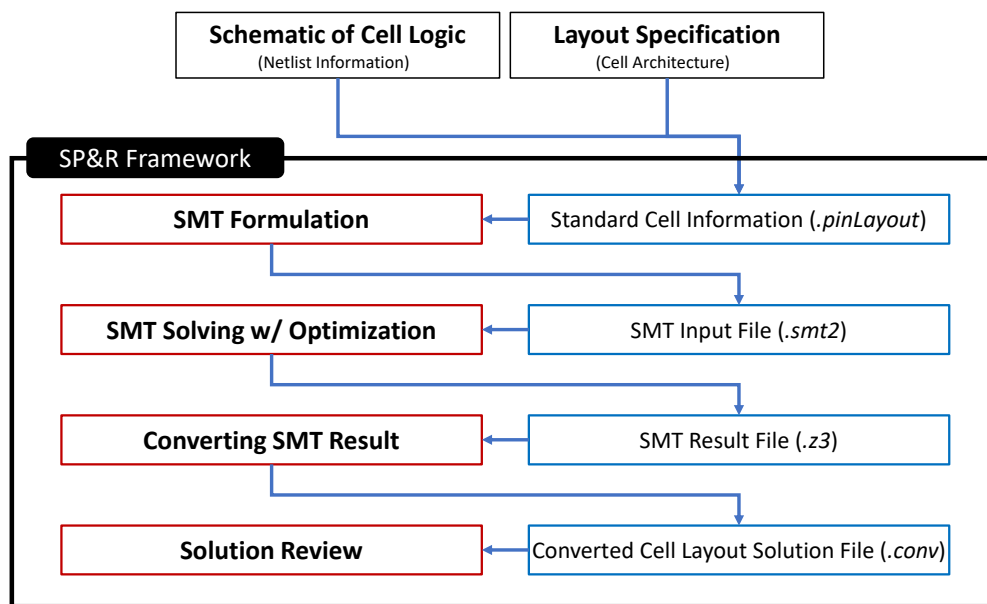
CK Research Group in UC San Diego

Email: ckcheng@ucsd.edu

1. Overview

This manual briefly summarizes the following flows to generate (i) SMT formulation file (*.smt2* file) and (ii) solution files to review the cell layout result. With the given standard cell information inputs (*.pinLayout* file) which are extracted from the ASAP7 PDK library[1], our flow generates the SMT formulation. We provide a solution viewer to validate the transistor placement and in-cell routing result of the SMT formulation. We employ Z3 (Ver. 4.8.5) [2] as our SMT solver. Please find more details from our papers [3, 4].

(1) Flow Chart for Our Proposed Framework



(2) Contents in the TAR ball

```

(Current Path) /VFET/cmd_gen_smt
               /cmd_conv_solution
               /pinLayouts/*.pinLayout
               /pinLayouts_T1/*.pinLayout
               /pinLayouts_T2/*.pinLayout
               /pinLayouts_T3/*.pinLayout
               /pinLayouts_T4/*.pinLayout
               /inputsSMT/*.smt2
               /RUN/*.z3
               /scripts/convSMTResult_Ver1.0.pl
               /genSMTInput_Ver1.0.pl
               /solutionsSMT/*.conv

```

2. Our Tool-Chain Scripts and Commands with User-Specified Options

Our tool-chain scripts are written in *Perl*. SMT solver is Z3 (Ver. 4.8.15). For the information of the Z3 solver, please visit the following link: <https://github.com/Z3Prover/z3>

** Z3 Solver has been frequently updated. We recommend to use the specific version V4.8.15*

(1) Input Standard Cell Information (.pinlayout)

We provide 30 representative standard cell information which are extracted from the ASAP7 PDK library [1]. The list of standard cells is as follows.

AND2x2	BUFx2	FAx1	NAND2x1	NOR2x2	OR2x2
AND3x1	BUFx3	INVx1	NAND2x2	NOR3x1	OR3x1
AND3x2	BUFx4	INVx2	NAND3x1	NOR3x2	OR3x2
AOI21x1	BUFx8	INVx4	NAND3x2	OAI21x1	XNOR2x1
AOI22x1	DFFHQNx1	INVx8	NOR2x1	OAI22x1	XOR2x1

(2) SMT Formulation Generation (genSMTinput_VFET_Ver1.0.pl)

[Usage]

```

$ ./scripts/genSMTInput_VFET_Ver1.0.pl [inputfile_pinLayout] [#Tier] [MPO]
[BreakingSymmetry] [CellPartition] [Localization] [Tolerance]

```

* [inputfile_pinLayout] : path for input pinLayout (ex: pinLayouts_T1~T4/AND2x2.pinLayout)

* [#Tier] : Number of Tier, 1~4 (integer)

* [MPO] : Minimum Pin Opening parameter (integer)

* [BreakingSymmetry] : Breaking Symmetry – 0:disable, 1:enable

* [CellPartition] : Cell Partitioning – 0:disable, 1:enable

→ To enable Cell Partitioning Feature, partitioning information should be specified in pinLayout inputs. Please refer to the sample pinLayout (DFFHQNx1.pinLayout). The partitioning info is described in “i ===PartitionInfo===” section.

- * [Localization] : Localization – 0:disable, 1:enable
- * [Tolerance] : Offset Margin for Localization (integer)
- * Please refer our papers [3, 4] for further detailed information of each input parameters.
- * **Cell Partitioning and Breaking Symmetry options can not be used at the same time.**

[Example]

Generating the SMT formulation file (*.smt2*) for the ND2x2 standard cell “AND2x2.pinLayout” with #Tier=2, MPO=2, Breaking Symmetry = Enable, Cell Partitioning = Disable, Localization = Enable with Tolerance = 1.

```
$ ./scripts/genSMTInput_VFET_Ver1.0.pl pinLayouts_T2/AND2x2.pinLayout 2 2 1 0 1 1
```

This will create “AND2x2_T2.smt2” file in the `inputsSMT` directory. For the *.smt2* file format, please refer to the [SMT-LIB Standard Version 2.0](#).

* In our work [3], we set different parameters for combinational and sequential logic cells due to the use of cell partitioning feature for the sequential logic cells. Please refer to the pre-described command list (`cmd_gen_smt`) for the parameters applied to each cell.

(3) RUN SMT Solver (z3)

[Usage]

SMT Solving & Storing solution.

```
$ z3 inputsSMT/[inputFile(.smt2)] > RUN/[solutionName(.z3)]
```

[Example]

Running “AND2x2_T2.smt2” file and storing the result “AND2x2_T2.z3” to the output directory.

```
$ z3 inputsSMT/AND2x2_T2.smt2 > RUN/AND2x2_T2.z3
```

(4) Solution Converter (`convSMTResult_Ver1.0.pl`)

[Usage]

```
$ ./scripts/convSMTResult_Ver1.0.pl [solPath/solutionName] [inputFile_pinLayout (w/o file extension)]"
```

[Example]

Converting “AND2x2_T2.z3” output file generated from the input pinLayout “AND2x2.pinLayout” to the solution output directory

```
$ ./scripts/convSMTResult_Ver1.0.pl RUN/AND2x2_T2.z3 AND2x2
```

This will create “[**solutionName**].conv” file in the solutionsSMT directory.

The converted solution files (.conv) can be reviewed using an excel-based solution viewer.

(SolutionViewer_VFET.xlsm)

(5) Pre-described Command Lists

There are “cmd_conv_solution”, “cmd_gen_smt” files which consist of command lists to generate and convert the whole standard cells provided in this package. You can refer to these command file to modify the parameters or execute each cell generation or sourcing the list file to execute all cases.

3. References

- [1] V. Vashishtha, M. Vangala, and L. T. Clark, “ASAP7 predictive design kit development and cell design technology co-optimization,” in 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 992–998, IEEE, 2017
- [2] Z3, SMT Solver, <https://github.com/Z3Prover/z3>.
- [3] D. Lee, D. Park, C.-T. Ho, I. Kang, H. Kim, S. Gao, B. Lin, C.-K. Cheng, “SP&R: SMT-based Simultaneous Place- &- Route for Standard Cell Synthesis of Advanced Nodes”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020
- [4] D. Lee, C-T Ho, I. Kang, S. Gao, B. Lin, and C.-K. Cheng, “Many-Tier Vertical Gate-All-Around Nanowire FET Standard Cell Synthesis for Advanced Technology Nodes”, IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, 2021