Synopsys' Interoperable Process Design Kit

R. Goldman[1], K. Bartleson[1], T. Wood[1], V. Melikyan[2]
[1] *Synopsys, Inc., CA, USA*
[2] *Synopsys Armenia CJSC, Yerevan, Armenia*
vazgenm@synopsys.com

Abstract

An Interoperable Process Design Kit (PDK) for a 90nm custom design flow, including all the necessary design rules, models, technology files, verification and extraction command decks, scripts, symbol libraries, and Parameterized Cells (PCells) is described. The components of the PDK augment all types of custom design for educational and research purposes. Although the PDK does not contain any foundry information, it allows close to real 90nm technology with high accuracy to be implemented in the designs.

1. Introduction

In the age of nanometer technologies, universities strive to provide the most modern and high quality studies in IC design. In addition to Electronic Design Automation (EDA) tools and Educational Design Kits (EDKs [1]) from leading companies, Process Design Kits (PDKs) for different IC fabrication technologies are also necessary. But the creation of such PDKs is met with numerous difficulties such as labor-intensive development and considerable complexity of verification. The most important challenge is the intellectual property (IP) restrictions imposed by IC fabrication foundries that do not allow universities to copy their technology into PDKs. To overcome this challenge, Synopsys created a PDK that has characteristics very close to real foundry design kits but does not contain confidential information from foundries that restricts its use.

2. Overview of the PDK

Synopsys created an interoperable PDK that is free from intellectual property restrictions and is targeted for educational and research purposes. It is aimed at programs for training highly qualified specialists in the area of microelectronics at universities, research centers and professional training facilities. The PDK enables students to master today's advanced design methodologies and the capabilities of Synopsys' state-of-the-art custom implementation IC design tools. It allows students to design different analog ICs and IPs using 90nm technology and Synopsys' EDA tools.

The Synopsys PDK contains the following: technology files: physical verification files, parasitic extraction files, Spice models, schematic symbols, PCells, and scripts.

For the PDK's development, an abstract 90nm technology was used. While the PDK does not contain actual foundry data, which is confidential information from foundries, it is very close to real 90nm technology. Using abstract technology allowed Synopsys to create an interoperable PDK that can be used for study and research of real 90nm design characteristics.

3. Description of the PDK components

3.1. Technology Files

The technology files provide technology-specific information, such as the names and physical characteristics of each metal layer and the routing design rules. Technology files include technology, display resource and mapping files:

1. Technology file contains layer information, design rule definitions, and associations with packets defined in the Display Resource file.
2. Display Resource file (DRF) contains the colors, fills and stipples, as well as packets, which are combinations of each. These packets are later used by the layout tool to show layers that are associated with a specific packet.
3. Mapping file maps the OpenAccess (OA) layer names to the GDSII layer numbers and data types. It is used during importing and exporting designs in GDSII format.

3.2. Physical verification files

Physical verification files include Hercules runset files for Design Rule Check (DRC) and Layout vs. Schematic (LVS) checks. DRC checks compliance of physical structures to design rules. LVS performs a comparison process that verifies whether the geometric or layout implementation of a circuit matches the schematic representation. Runset files are needed to instruct the tool on the operations performed.

1. DRC runset file is used by the DRC software program that analyzes the data in the layout and calculates its interaction (spacing and overlaps) to other layers.
2. LVS runset file is used by the LVS software program that extracts the intended devices and their parameters to compare with schematic netlists.

3.3. Parasitic extraction files

Parasitic extraction files are used by the parasitic extraction software program that extracts parasitic capacitance, resistance and/or inductance from circuit layout. StarRC, the parasitic extraction tool from Synopsys, uses the Interconnect Technology File (ITF), TLUPlus models (TLU+) as well as mapping and command files as its input. ITF defines a cross section profile of the process. TLUPlus models are a set of models containing advanced process effects that can be used by the parasitic extractors in place-and-route tools for modeling. The Mapping file maps layers in the technology file to the layers defined in ITF. The command file is used to instruct the tool on the extraction process.

3.4. Spice Models

The generic Spice model library is based on the Predictive Technology Model [2] and contains the following devices:

- transistors
  - 2.5V devices: thick oxide MOSFETs
  - 1.2V devices: thin oxide MOSFETs with typical, high, and low threshold voltages.

    For these devices five corners were created depending on the performance of NMOS and PMOS transistors (fast, slow or typical) with varying threshold voltage and gate oxide thickness: TT - both typical; FF - both fast; SS - both slow; SF – slow nmos/fast pmos; FS – slow pmos/fast nmos.
- diodes
- unsalicided and salicided poly resistors, P+ resistor
- varactors
- inductors
- capacitors
- BJTs

In order to estimate the accuracy of the SPICE models, the models' parameters were scaled to 0.25um technology to compare them with the characteristics of known 0.25um models (Figure 1). A set of DC transfer curves was obtained and the middle curve from the set was chosen as a typical corner for 2.5V devices, thereby assuring that it is close to a real foundry process.

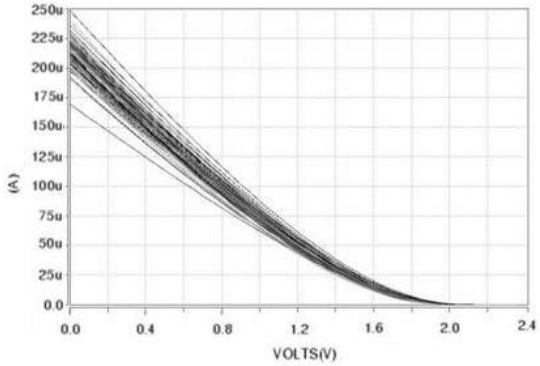


Figure 1. Set of .25um Transfer Curves

FF, SS, SF and FS corner models were formed by changing the threshold voltage (v_{th0}) and oxide thickness (tox) in the range of +/-5%. Figure 2 shows the transfer curves for TT, FF and SS corners of a thick oxide NMOS model.

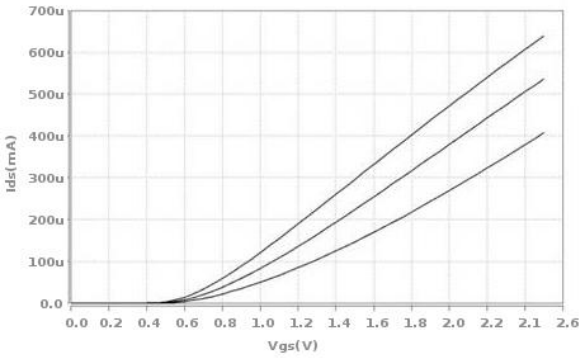


Figure 2. TT, FF and SS corners of 2.5V thick oxide NMOS

3.5. Schematic Symbols

The interoperable PDK includes an OA library that contains schematic symbols for devices such as MOS transistors, resistors, BJTs, diodes, varactor, inductor, and capacitor. The symbols were developed to work in any OA compatible [3] tool so the 90nm PDK is interoperable. The Synopsys OA compatible tool is Custom Designer.

3.6. PCells

A parameterized cell (PCell) is a layout cell generator programmed with a scripting language. Each PCell has a pre-defined set of parameters that can be strings, boolean, or numeric values. After execution of code by an interpreter it generates a cell view in an OA database. PCells included in the interoperable PDK are written in Python, which enables use of object-oriented techniques. The full advantages of object-oriented methodologies are applied to the creation of PCells. They can be used to reuse the code by defining well formed inheritance trees thereby reducing development time.

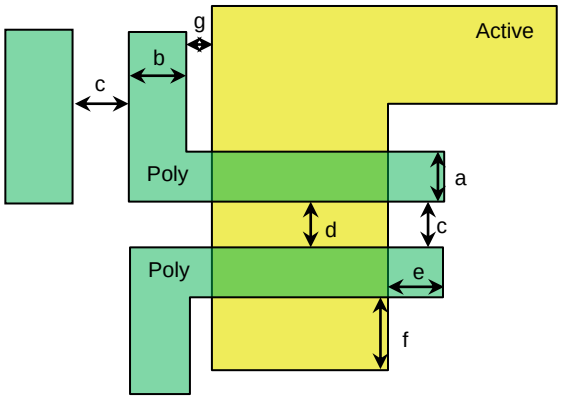
3.7. Scripts

Scripts are commonly used to do various functions in an OA compatible editor while using symbols and PCells. The common uses of scripts are: LVS, DRC and Extraction setup, invalid input checks, integer checks for parameters (m, nf, series, parallel, etc.), min/max checks and off-grid checks for non-variable inputs. Scripts can throw warning/error messages or even change parameters if the change is unambiguous. The scripts that perform checks are callback scripts tied between the Schematic or Layout editor's graphical interface and the PDK. This approach enables users to fully integrate the PDK components into the editor allowing the user to work with the PDK in a familiar

environment. These scripts, though invisible to the user, provide a means to work with device parameters in the editor's native dialog boxes. They also perform checks, report errors and calculate device electrical parameters based on the physical dimensions and vice-versa. For example, callback scripts for resistor can calculate resistance if the user provides the dimensions, and can calculate dimensions for the needed resistance value.

3.8. Design Rules

These rules were created by using the MOSIS Scalable CMOS (SCMOS) design rules. They provide greater portability of designs than if 90nm rules were developed because the sizes in 90nm rules can be larger by 5-20% than those in real foundry processes. An example design rule is illustrated in Figure 3.



a=0.1, b=0.3, c=0.18, d=0.2, e=0.18, f=0.16, g=0.05um

Figure 3. Example Design Rule

Also design rules contain a GDSII Layer Map. It maps layer names to the corresponding GDSII layer numbers used in the 90nm process. Some layers such as dummy, marking, and text, have been added to the MOSIS layer map. A sample of the layer map is shown in Table 1.

Table 1. Sample of Layer Map

Layer #	Data type	Tape Out Layer	Drawing or Composite Layer	Layer name in TechMap File	Layer Name in DRC	Layer Name in LVS	Layer usage description
1	0	YES	Drawing	NWELL	NWELLi	NWELL i	NWELL
2	0	YES	Drawing	DNW	DNWi	DNWi	Deep NWELL

…

4. PDK's deployment

Currently the PDK is in use for both educational and research purposes by students in microelectronics area at a number of universities: State Engineering University of Armenia, Yerevan State University, Russian-Armenian Slavonic University, Moscow Institute of Electronic Technology, etc. Use of the PDK is highlighted in several courses: RF IC Design, Analog IC Design, Mixed-Signal IC Design, etc. The PDK is used in these courses for implementing both laboratory works and course projects by students. For example, in the Analog IC Design course, the laboratory works cover all the schematics that are taught during the course and students implement them using devices included in the 90nm PDK. Additionally, the course project is based on designing different operational amplifier schematics using the 90nm PDK and extracting various characteristics. Results show that using the 90nm PDK in course projects and laboratory works is more effective than relying on generic PDKs because they lack the necessary technology-specific effects. As the field of microelectronics constantly evolves, students need to be able to study as much real effects as possible to become competitive. Also schematics in such courses as RF IC Design or Mixed-Signal IC Design usually include devices like resistors, capacitors and inductors that have complex layouts. The presence of PCells for these devices not only enables students to easily design large circuits, but also lets students learn their complex structure. Although foundry-specific design rules are not included in the PDK it does not impact the educational process because most of the real effects are included.

5. Future PDK development

To keep pace with industry advancements and to support smaller process technology nodes, new PDKs may be created. Development to support new device structures in the 90nm Interoperable PDK is ongoing.

6. Conclusion

An interoperable Process Design Kit (iPDK) was created for educational and research purposes. It is free from intellectual property restrictions and is representative of industrial design kits. It can be used in a wide range of design flows for digital, analog and mixed-signal designs using Synopsys' EDA tools.

7. References

[1] Goldman R., Bartleson K., Wood T., Kranen K., Cao C., Melikyan V. Synopsys Open Educational Design Kit: Capabilities, Deployment and Future. Proceedings of the International Conference on Microelectronic Systems Education, San Francisco, USA, 25-27 July, 2009, pp. 45-48

[2] Stine J., Castellanos I., Wood M., Henson J., Love F. Free PDK: An Open-Source Variation-Aware Design Kit.- Proceedings of the International Conference on Microelectronic Systems Education, San Diego, USA, 3-4 June, 2007, pp.173-174

[3] OpenAccess Documentation (http://www.si2.org/?page=621)