IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 8, Issue 5, Ver. I (Sep. - Oct. 2018), PP 18-29
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Educational Introduction to VLSI Layout Design with Microwind

George P. Patsis[1,2]
[1]*Department of Electrical and Electronics Engineering, University of West Attica, Athens, Greece*
[2]*Nanometrisis private company, Scientific & Technological Park "Leukippos", NCSR Demokritos, Neapoleos 27 & Patr. Grigoriou Str., 15341, Agia Paraskevi, Attica, Greece*
Corresponding Author: George P. Patsis

Abstract: *VLSI design course concepts are easier to comprehend with the use of accompanying software examples. Using the student-version of Microwind, students are introduced to the design of circuits in the layout level. In the first lectures the difficult to grasp parts are those related to the concepts of design rules, relation between layout and cross-section view, the role of multiple contacts, n-well polarization, latchup, design of basic mosfet structures using layout generator or custom design and their simulation, and automatic layout generation from a Verilog description of the circuit. These are the topics discussed in the current article with emphasis on the conceptual significance of the cross-section views of a design in order to assist students understand that the layout is a top-down view of a three - dimensional stack of materials. Also some practical examples of complete layouts of basic introductory circuits are presented.*

Keywords: *VLSI Design, Microwind, Layout, Design Rules, DRC, Mask, Material Layers*

Date of Submission: 21-10-2018 Date of acceptance: 03-11-2018

I. Introduction

The fabrication of integrated circuits is based on the layout of patterned material-layers one on top of the other. Several such layers are not accessible to the circuit's layout-designer because they are required for isolation (e.g. the silicon dioxide layer between metal layer tracks) or electrical contacts between the semiconductor device and the supporting interconnect structure (e.g. salicide layer). In order to get from circuit-topology to the actual layout-topology, the designer should have a firm understanding of the representation of devices in a layout and also the ability to do the reverse, i.e. seeing a layout and having an idea of the corresponding circuit and therefore the circuit's function. Specialize software tools aid the designers in both of these directions.

Microwind is a friendly Windows-PC software-tool for designing and simulating microelectronic circuits at layout level. The tool features full editing facilities attractive views such as MOS characteristics, 2D cross-sections, 3D views, atomic views and efficient analog simulator [1, 2]. No SPICE or external simulator is needed. The analog simulator is built-in. The simulator features fast time-domain, voltage and current estimation, with very intuitive post processing: frequency estimation, delay estimation. Even power estimation is on-screen. The simulation can be performed on a wide range of technologies: 1.2 micron down to 35 nm. Microwind uses LEVEL3 and BSIM4 model. LEVEL3 is an older model not suitable for sub-micron Technologies. BSIM4 model is considered as accurate downto nano-scale technology, but inadequacies of such modeling for GHz range analog simulation have been reported. Electrical extraction in Microwind is quite accurate for logic simulation and circuit validation, but the analog simulations may significantly differ from what professional tools could provide, due to the fact that the extractor do not take into account all capacitance, resistance and inductance effects. Therefore, Microwind is considered acceptable for pre-analysis, comparison and educational purposes, but an industry-standard validation is recommended before circuit fabrication.

II. Layout Of Basic Material Layers

The standard layers accessible to the designer in the case of Microwind [1, 2] are seen on its pallete (Fig. 1(a)). These are the n-well, the n+-diffusion, the p+-diffusion, the polysilicon, the contact, the polysilicon 2, and various metal layers (Fig. 1(b)). They appear in plane in the top-down view, but actually they are not. The lambda-grid size obeys the relation $\lambda=L/2$, where L is the characterizing minimum length of the design-rule-technology used. For example, if $L=0.25\mu m$, then $\lambda=0.125\mu m$. The order of the previous list of layer-names is actually depictive of their depth-level in the 2D cross section of the device. Specifically, the n-well regions are fabricated first (they are deeper) than the polysilicon or metal layers (Fig. 1(c)). The contacts are necessary for connecting layers located on different depths. Each contact is actually a via (a metal containing cylinder that

cuts through from one layer to the next) and a metal surface. Figure 2 shows the layout and cross-section of various contacts in Microwind. Also, Fig. 3 shows the part of the Layout-Generator that is used to create complex contacts between several layers. Contacts of metals higher than metal 2 are larger in area. Option layer (Fig. 4(a)) is a virtual layer (not an actual mask layer) used to specify certain options in the area that is drawn. Figure 4(b) shows the options offered by Microwind. One of these options gives the ability to remove the salicide on top of the polysilicon layer, which increases its resistance (Fig. 4(c)).

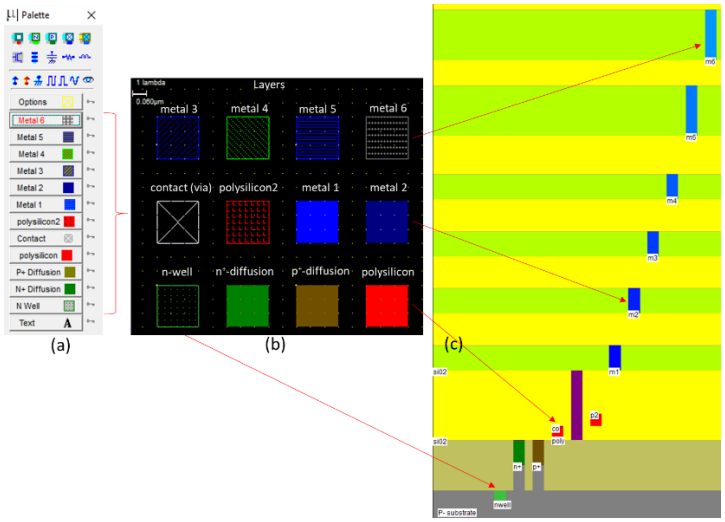

Fig.1. (a) Microwind's pallete showing various layers and tools assisting layout design. (b) 3λ x 3λ squares are depicted one for each layer. Starting from n-well which is the deepest level, the layers become shallower as the series progresses to metal 6. (c) Cross-section among the various layers.

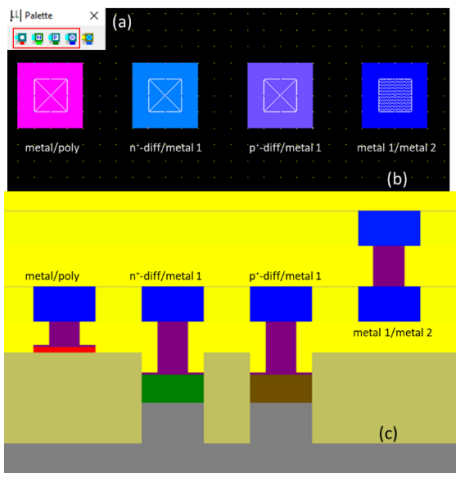

Fig. 2. Contacts in Microwind. Cross-section among various contacts.

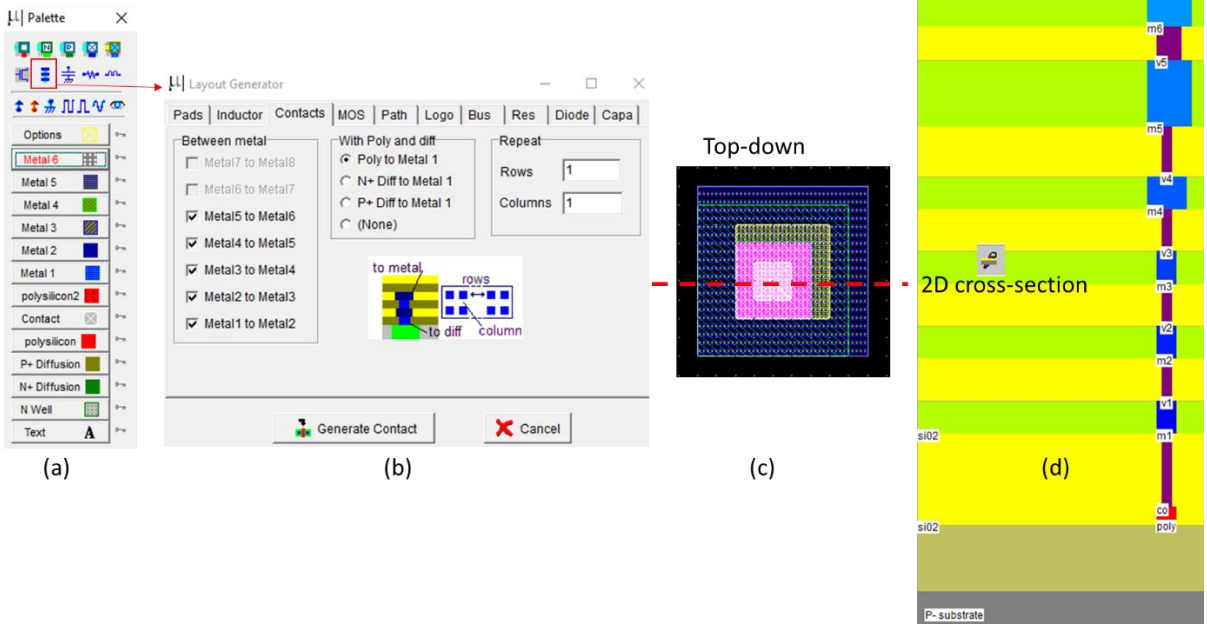

Fig. 3. Layout generator can be used to create complex contacts between various metal layers. Contacts of metals higher than metal 2 are larger.

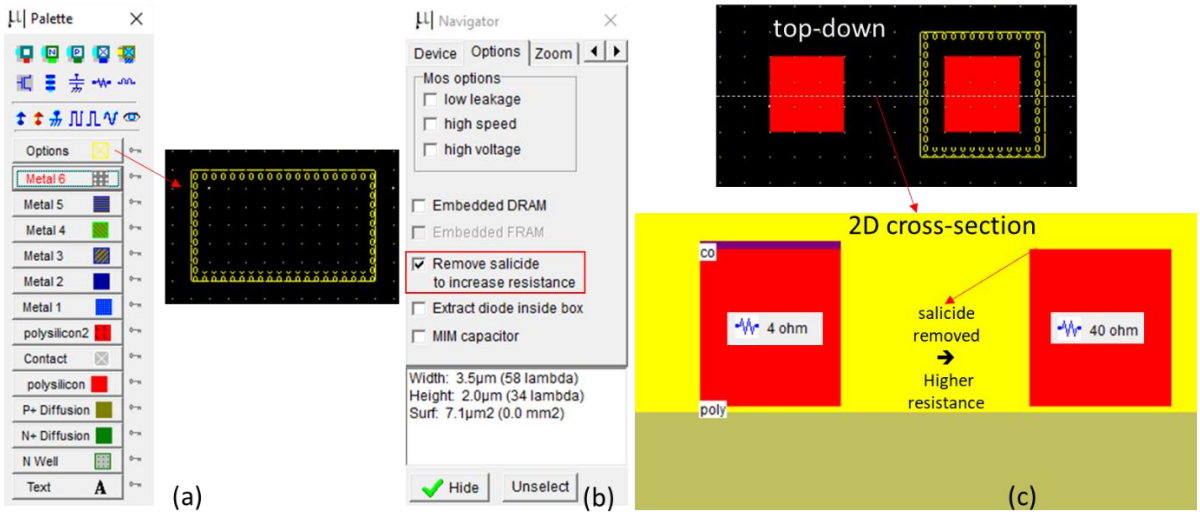

Fig. 4. (a). Option layer. It is a virtual layer (not a mask layer). (b, c). One of its properties is to remove the salicide to increase layer resistance.

III. Design Rules

Microelectronic revolution is based on technology scaling. This is the shrinking of transistor size in deep submicron and to the nanometer range. Due to technical difficulties in the actual fabrication of the designed devices, there are certain rules that a layout has to comply with. These rules pose certain limits about how close can be two areas of the same material (and therefore on the same level) or on the minimum dimensions of the fabricated shapes with a given technology node. These design rules are quantified in µm or in lambda (λ) units. The value of lambda is half the minimum polysilicon gate length or λ = L/2 and L being the minimum dimension of polysilicon gate fabricated in each technology node. For example, in 0.25µm fabrication technology, λ=0.125µm.

Microwind software is based on a lambda grid, not on a micro grid. Consequently, the same layout may be simulated in any CMOS technology. The software can handle various technologies. The process parameters are stored in files with the appendix '.RUL'. The default technology corresponds to a generic 6-metal 0.12 µm CMOS process. The default file is CMOS012.RUL. In order to select a new foundry, the user has to click on File ➔ Select Foundry and choose the appropriate technology in the list. Rule files are simple text description of design rules. For example, Table 1 shows a small portion of the design rule file for the 0.12 µm CMOS technology. So in order to produce circuits functioning correctly it is mandatory to follow these design rules. The software's built-in design-rule-checker (DRC) can check at any moment if any of the rules is broken and will inform the designer of the exact position in the design with a conflict.

Table 1. Partial contents of CMOS012.RUL file showing the design rules for layer dimensions and positioning. The rule values are in lambda units.

NAME CMOS 0.12µm - 6 Metal * * Design rules associated to each layer * * Well * r101 = 10 (well width) r102 = 11 (well spacing)	* * Diffusion * r201 = 4 (diffusion width) r202 = 4 (diffusion spacing) r203 = 6 (border of nwell on diffp) r204 = 6 (nwell to next diffn) r205 = 0 (diffn to diffp) r210 = 16 (Minimum diff surface lambda2)
* * Poly * r301 = 2 (poly width) r302 = 2 (gate length) r303 = 4 (high voltage gate length) r304 = 3 (poly spacing) r305 = 1 (spacing poly and unrelated diff) r306 = 4 (width of drain and source diff) r307 = 3 (extra gate poly) r310 = 16 (Minimum poly surface lambda2)	* * Poly 2 * r311 = 2 (poly2 width) r312 = 2 (poly2 spacing)

* * Contact * r401 = 2 (contact width) r402 = 4 (contact spacing) r403 = 1 (metal border for contact) r404 = 1 (poly border for contact) r405 = 1 (diff border for contact) r406 = 2 (contact to gate) r407 = 1 (poly2 border for contact)	* * Pad rules * rp01 = 1330 (Pad width 80µm) rp02 = 1330 (Pad spacing 80µm) rp03 = 40 (Border of Vias) rp04 = 40 (Border of metals) rp05 = 200 (to unrelated active areas)
* * metal * r501 = 3 (metal width) r502 = 4 (metal spacing) r510 = 16 (minimum surface)	* * via * r601 = 2 (Via width) r602 = 4 (Spacing) r604 = 1 (border of metal) r605 = 1 (border of metal2)
* * metal 2 * r701 = 3 (Metal 2 width) r702 = 4 r710 = 16 (minimum surface)	* * via 2 * r801 = 2 (Via width) r802 = 4 (Spacing) r804 = 1 (border of metal2) r805 = 1 (border of metal3)
* * metal 3 * r901 = 3 (width) r902 = 4 (spacing) r910 = 16 (Minimum surface)	* * via 3 * ra01 = 2 (Via width) ra02 = 4 (Spacing) ra04 = 1 (border of metal3) ra05 = 1 (border of metal4)
* * metal 4 * rb01 = 3 (width) rb02 = 4 (spacing) rb10 = 16 (Minimum surface)	* * via 4 * rc01 = 2 (Via width) rc02 = 4 (Spacing) rc04 = 1 (border of metal4) rc05 = 3 (border of metal5)
* * metal 5 * rd01 = 8 (width) rd02 = 8 (spacing) rd10 = 64 (Minimum surface)	* * via 5 * re01 = 5 (Via width) re02 = 5 (Spacing) re04 = 2 (border of metal5) re05 = 2 (border of metal6)
* metal 6 rf01 = 8 (width) rf02 = 8 (spacing) rf10 = 144 (minimum surface)	

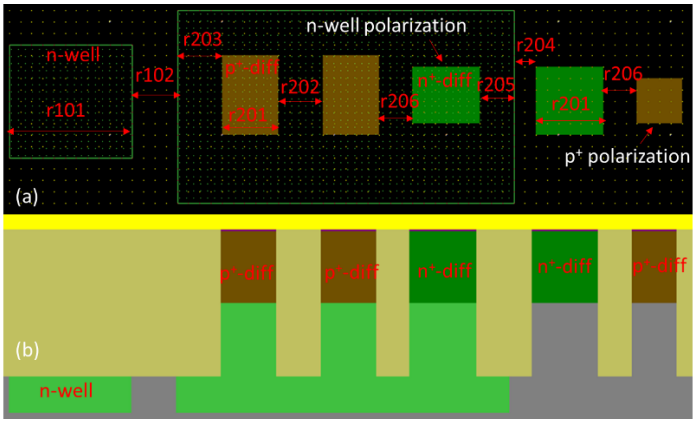

Fig. 5. Detailed top-down and cross-section view of n-well, n+, p+ diffusion placement design rules.

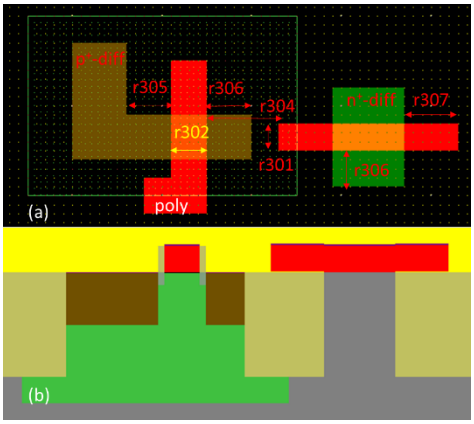

Fig. 6. Detailed top-down and cross-section view of polysilicon placement design rules.

Figure 5(a) shows examples of design rules related to n-well, p^+-diffusion, n^+-diffusion, and the corresponding polarizations. Figure 5(b) shows the corresponding cross-section. Figure 6(a) shows design rules

related to polysilicon layer, and Fig. 6(b) the corresponding cross-section. Figure 7(a) shows design rules related to polysilicon-2 layer, and Fig. 7(b) the corresponding cross-section. Figure 8(a) shows design rules related to contact (via) placement and Fig. 8(b) the corresponding cross-section. Figure 9(a) shows design rules related to metal-1 and Fig. 9(b) shows the corresponding cross-section. Figure 10 shows design rules related to pads design and the corresponding cross-section.

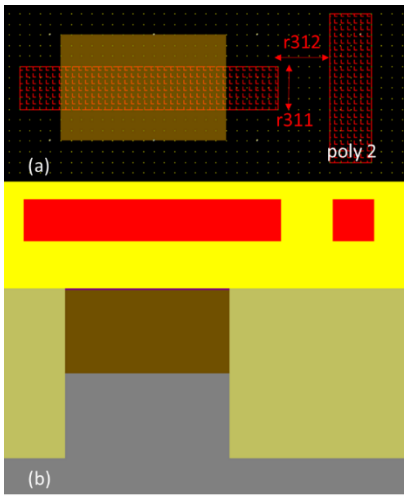

Fig. 7. Detailed top-down and cross-section view of polysilicon 2 placement design rules.

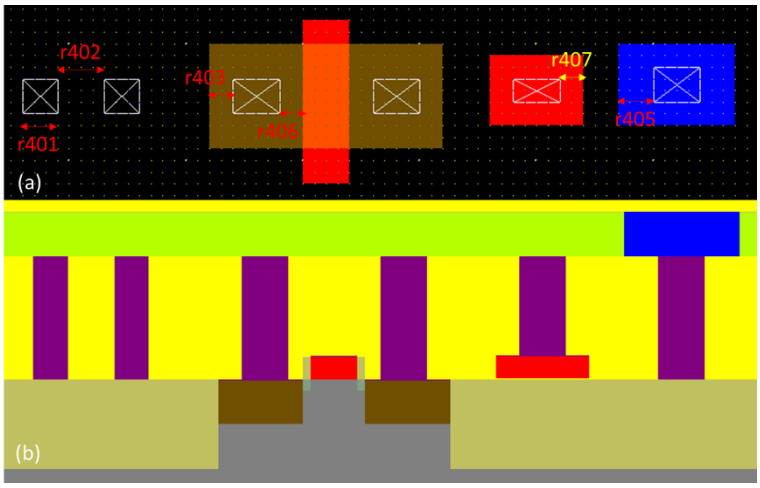

Fig. 8. Detailed top-down and cross-section view of contact (via) placement design rules.

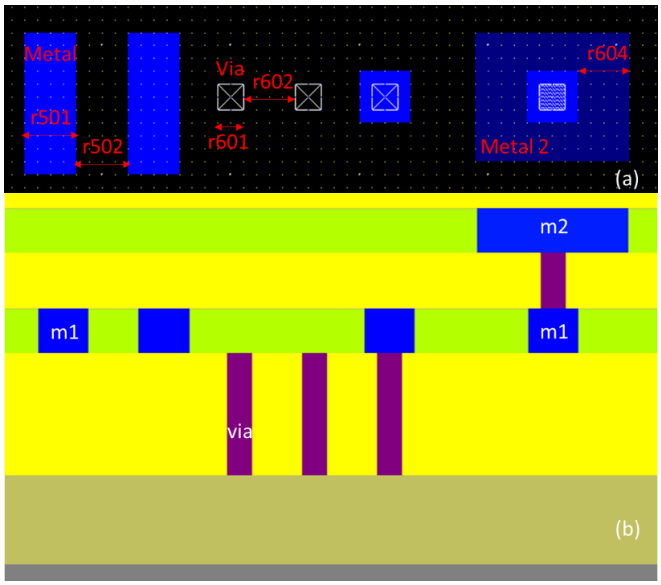

Fig 9. Detailed top-down and cross-section view of metal and via design-rules. Similar rules apply to higher metal layers and corresponding viases.

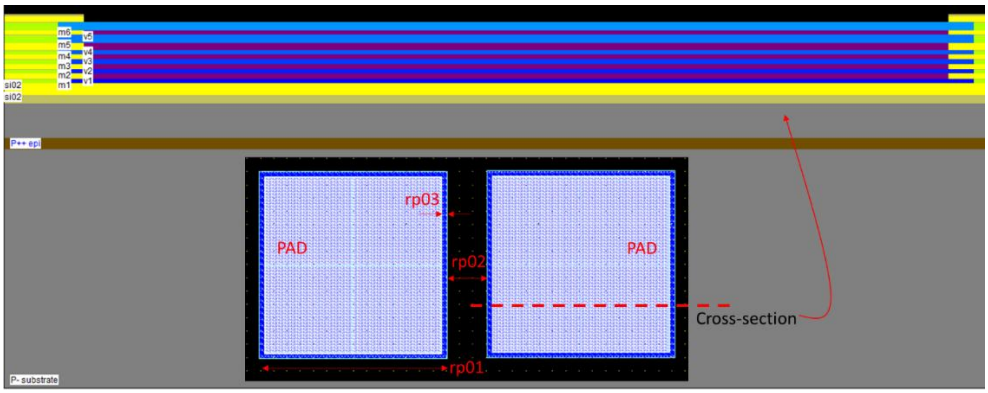

Fig. 10. Detailed top-down and cross-section view of basic PADs design-rules.

IV. Design Of MOSFETS

The next step in this introduction to VLSI, is to design simple MOSFET devices. This can be done either by drawing each layer by hand or using the software's Layout-Generator (the button with the MOS symbol on Microwind's pallete). New kinds of MOS devices have been introduced in deep-submicron technologies, starting with the 0.18 µm CMOS process generation. Microwind's version used in the current article incorporates the low-leakage MOS, the high-speed MOS, and the high-voltage MOS. All these layouts of the MOS can be generated easily using the Layout-Generator. Figure 11(a) shows the options in the Layout-Generator for these types of MOS, while Fig. 11(b) and Fig. 11(c) present their layout and cross-section respectively. There is no difference between the high speed MOS and the low leakage MOS from a layout point of view, except the option-layer for setting the high speed option. The high voltage MOS has a significantly different layout, due to the enlarged channel-length and width. Figure 12(a) and (b) show the nMOS and pMOS layouts and cross-sections respectively for comparison. These are created automatically using the Layout Generator.

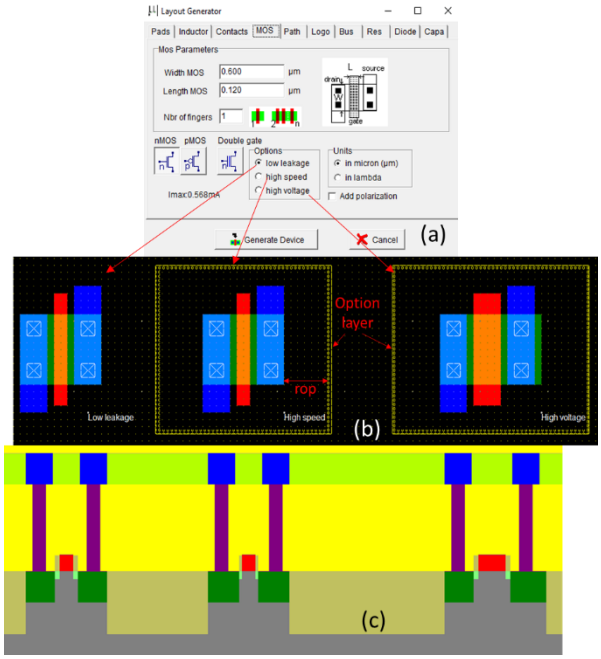

Fig. 11. (a) Layout generator, (b) top-down and (c) cross-section view of low leakage, high speed and high voltage nMOS device. In (b) also is shown the detailed top-down view of option-layer placement design rules.

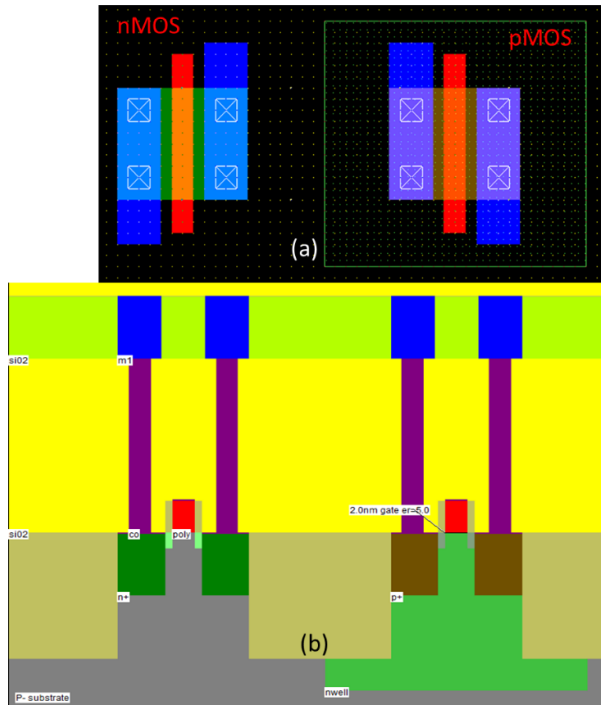

Fig. 12. nMOS and pMOS layout and cross-section view.

If one wishes to do custom design of the layout, then for the nMOS device has to 1) draw the gate, 2) add the n+ diffusion, 3) add contacts on drain and source region (adding more contacts is better), 4) Adding

simulation properties (Fig. 13(a)). For the pMOS device the corresponding steps are 1) draw the gate, 2) add the p+ diffusion, 3) add n-well with space for polarization, 4) add contacts, 5) add n-well polarization, and 6) add simulation properties (Fig. 13(b)).

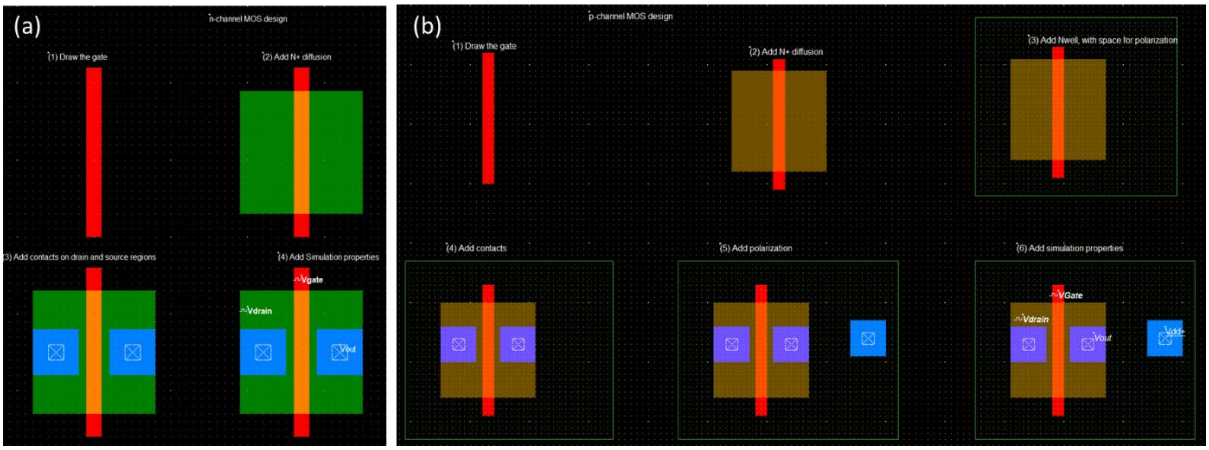

Fig. 13. Steps for custom design of MOS devices. (a) Simple nMOS. (b) Simple pMOS.

Students in VLSI design courses, usually don't realize initially the significance of a) adding multiple contacts, and b) adding n-well polarization. These two concepts are discussed below.

The metal contacts are necessary to enable an electrical signal access to the source and drain regions of the device. The reason for the addition of multiple contacts is the intrinsic current limitation of each elementary contact plug, as well as the high resistance of one single contact. One single contact can stand less than 1 mA current without any reliability problem. When the current is stronger than 1 mA, the contact can be damaged. The effect is called electromigration: if too much current flows within the contact, the metal structure starts to change as atoms move inside the conductor. A very strong current such as 10 mA would destroy one lonely contact. Adding as many contacts as the design rules permit, also limits the contact resistance. The equivalent resistance of the access to the drain and source regions is reduced proportionally to the number of contacts. Figure 14 shows this trend.

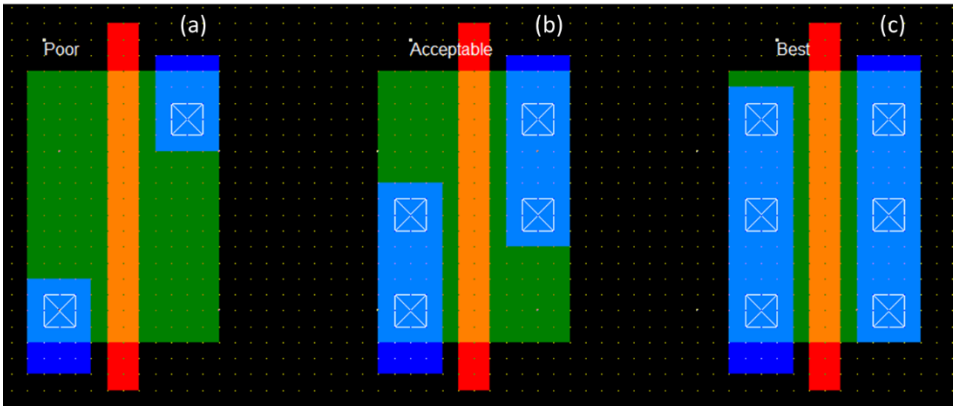

Fig. 14. Adding more contacts is desirable. (a) Poor nMOS design. (b) Acceptable nMOS design. (c) Best nMOS design.

In the case of the pMOS device, the n-well region cannot be kept floating. Refer to Fig. 15(a) and (b) for the layout of a pMOS without and with n-well polarization, and in Fig. 15 (c) and (d) for the corresponding cross-sections. A specific contact that can be seen on the left side of the n-well in Fig. 15 (a) and (b), serves as a permanent connection to high voltage (V_{DD}) on metal-2 layer. Substrate (p-type) is assumed connected to V_{SS}. Source is on metal 1. In the case seen on Fig. 15 (c), the n-well is floating. The risk is that the n-well potential can decrease enough to turn on the pn-junction. This case corresponds to a parasitic pnp device. The consequence may be the generation of a direct path from the V_{DD} supply of the drain to the ground supply of the substrate. In many cases the circuit can be damaged. The correct approach is indicated in Fig. 15(d). A polarization contact carries the V_{DD} supply down to the n-well region, thanks to an n+ diffusion. A direct contact with the n-well would generate parasitic electrical effects. Consequently, the n+ region embedded in the n-well area is mandatory. There is no more fear of parasitic pnp device effect as the p+/n-well junctions are in inverted

mode, and thus may be considered as a junction capacitance. Therefore, adding the polarization contact inside the n-well region, and add a supplementary V_{DD} property on top of the contact, is mandatory. Adding the V_{DD} polarization in the n-well region is a very strict rule in VLSI designs. Similar concept correspond to nMOS substrate, so it is also mandatory to add p-substrate polarization in this case. This is seen in the next section.

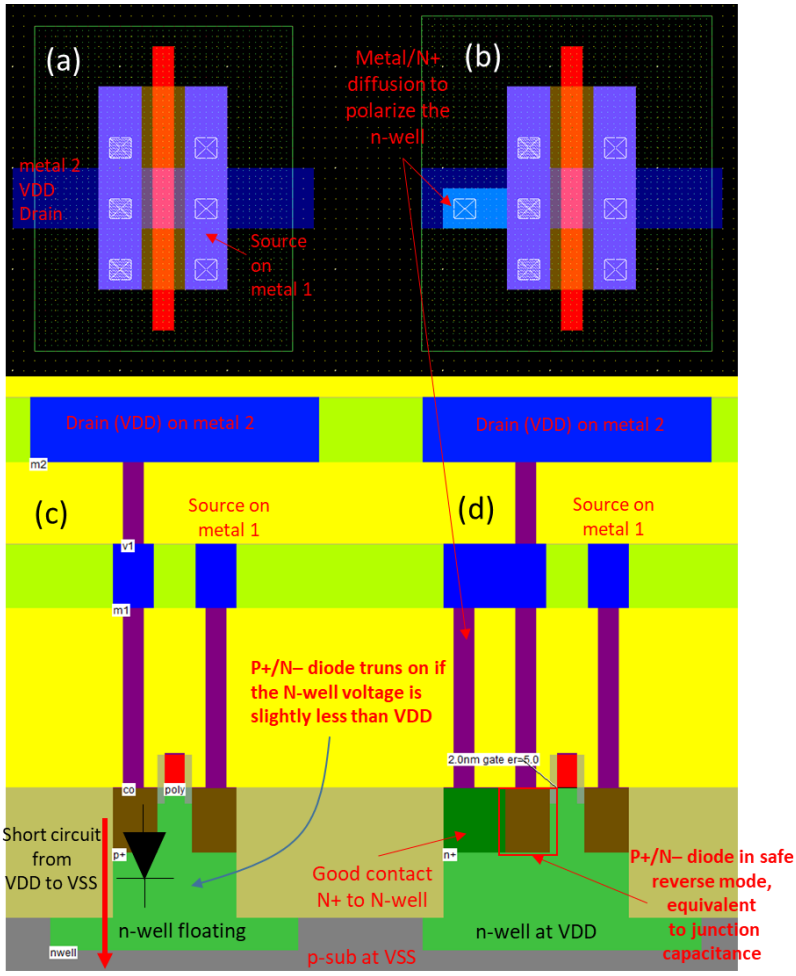

Fig. 15. (a) pMOS layout without n-well polarization. (b) pMOS layout with n-well polarization. (c) Cross-section of pMOS without n-well polarization. (d) Cross-section of pMOS with n-well polarization.

V. Simulation Of Mos Dynamic Behavior

The response of nMOS and pMOS on clock signals on their gate and source is examined in this section. Figure 16 shows the nMOS and pMOS devices designed with 0.12 µm rules and with the addition of suitable polarization on their substrates. The same clock signal Vg is added to both gates (but only one is made visible to simulation). Figure 17 (a) shows this Vg clock properties. The same clock signal Vs is also added to both sources (but again only one of them is made visible to simulation). Figure 17 (b) shows this Vs clock. V_{DD} voltage of 1.2V is assigned on each polarization. Finally, the Voutn and Voutp variables are assigned and made visible to the simulator on the drain of each device. Figure 18 shows the result of timing analysis of both devices. Theoretically, nMOS device conducts when its gate is above threshold or in digital terminology, when the gate is "1" and is off when the gate is "0". On the other hand, the pMOS device conducts when its gate is "0" and is off when its gate is at "1". This situation is illustrated on Fig. 18 with the pMOS on/off and nMOS on/off remarks. The Vg clock is approximately half the frequency of the Vs clock, in order to test all possible combinations of conductance for both nMOS and pMOS. Fist consider the nMOS. When nMOS is on, the signal from Vs should theoretically appear on Voutn. So if nMOS is on (Vg = "1"), then if Vs = "1" ➔ Voutn = "1" and if Vs = "0" ➔ Voutn = "0". This is actually what is observed on the Voutn timing line, with the remark that only the passage of a "0" is good while the passage of "1" is poor (i.e., Voutn does not reach 1.2V but V_{DD}-V_{tn}, where V_{tn} the threshold voltage of the nMOS). Now consider the pMOS case. When pMOS is on, the signal from Vs should appear on Voutp. So if pMOS is on (Vg = "0"), then if Vs = "1"➔ Voutp = "1" and if Vs = "0" ➔ Voutp = "0". Observing the Voutp timing line, it is seen that this is practically true but with the remark that

only the "1" is good, while the "0" is poor (i.e., Voutp does not reach 0V, but 0-V_{tp}, where V_{tp}<0 the threshold voltage of the pMOS). So it is clear that an nMOS device does a "good job" in passing a "0" from source to drain, but "not so good" in passing a "1", while the pMOS device does the dual, i.e., it passes reliably a "1" from source to drain, but not so well when it comes to passing a "0". CMOS technology uses both kinds of devices in such a design so as to obtain each time the good "0" or "1". This is discussed in the not-gate and the transmission-gate built with CMOS technology, in the next two sections.

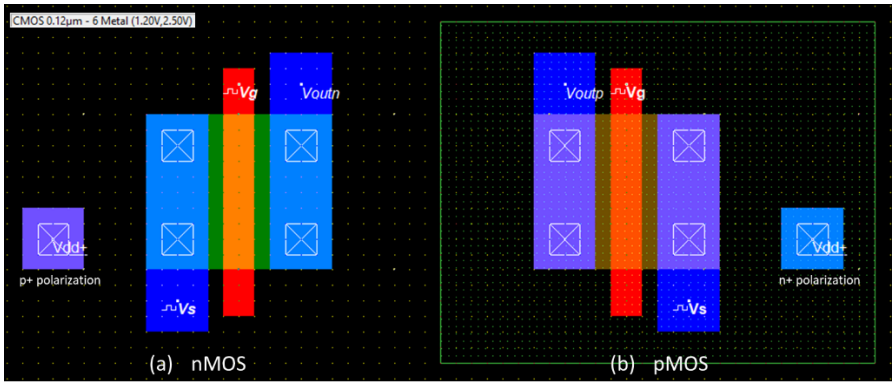

Fig. 16. (a) nMOS and (b) pMOS devices with added simulation properties and appropriate polarization.

Fig. 17. Simulation properties. (a)Vg, (b) Vs, (c) Voutn, (d) Voutp.

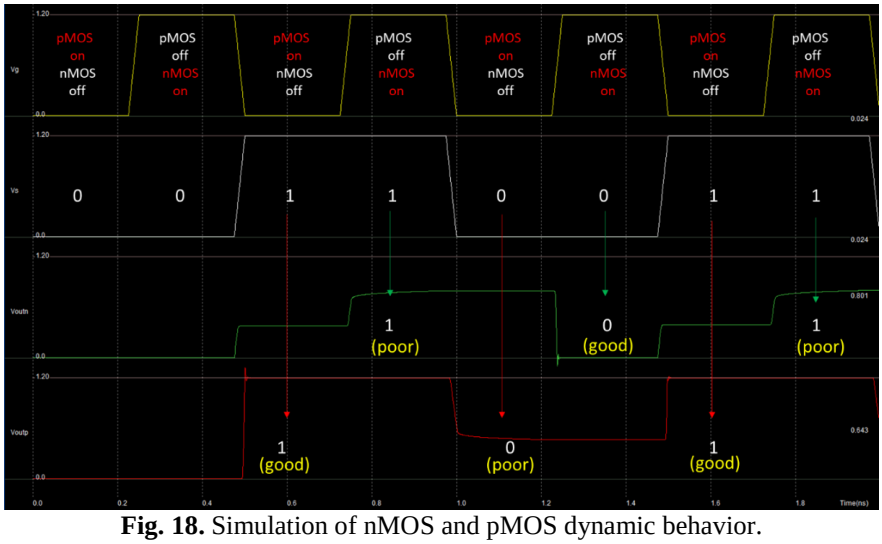

Fig. 18. Simulation of nMOS and pMOS dynamic behavior.

VI. Design And Simulation Of An Inverter (Not-Gate)

Having the nMOS and pMOS layouts it is easy to combine them in an inverter circuit. Figure 19 shows the various layout design steps to create the inverter semi-automatically. Using the Layout generator an nMOS and a pMOS is added. Next a polysilicon bridge is added to connect the nMOS and pMOS bridges, and a metal contact of the polysilicon bridge to metal-1 is created. A metal-1 bridge is used to connect the drains of the two devices. Next metal-2 is added for carrying the V_{DD} and V_{SS} signals. Finally. Via is added between metal-2 and metal-1 and appropriate polarization for the nMOS and pMOS device. Gate In1 is assigned a clock signal and node Out1 is the output of the not-gate. Simulation shown in Fig. 20 verifies the function of this design as a not-gate. When In1 = "0", pMOS is "on" passing to Out1 a good "1", while nMOS is "off", and when In1 = "1", nMOS is "on", connecting Out1 to ground and thus providing a good "0", while pMOS is "off". So in either case the "good for the job" device is selected.

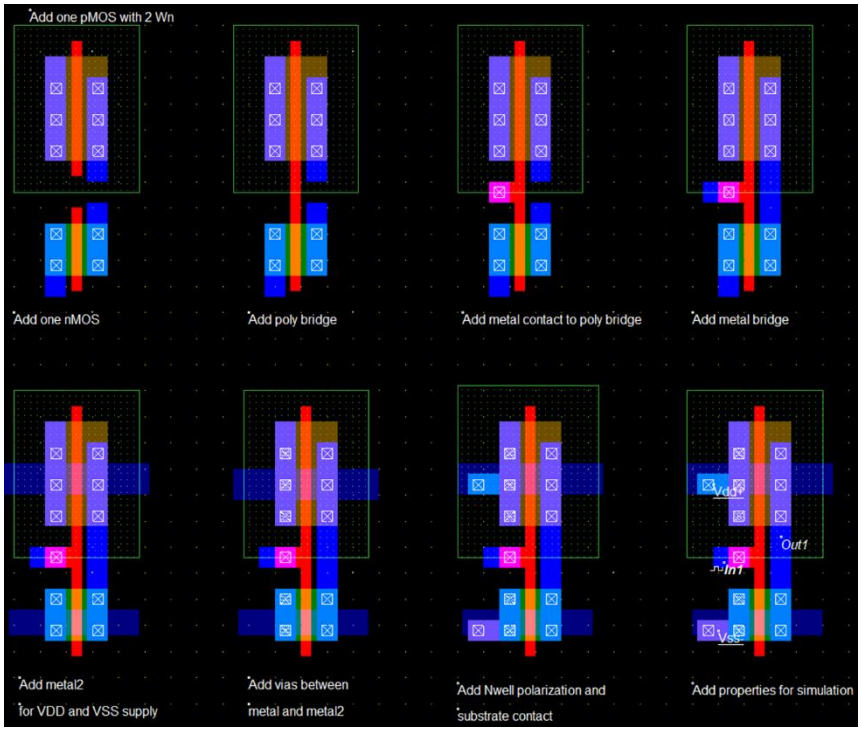

Fig. 19. Layout design steps for a CMOS not gate using 0.12µm technology.

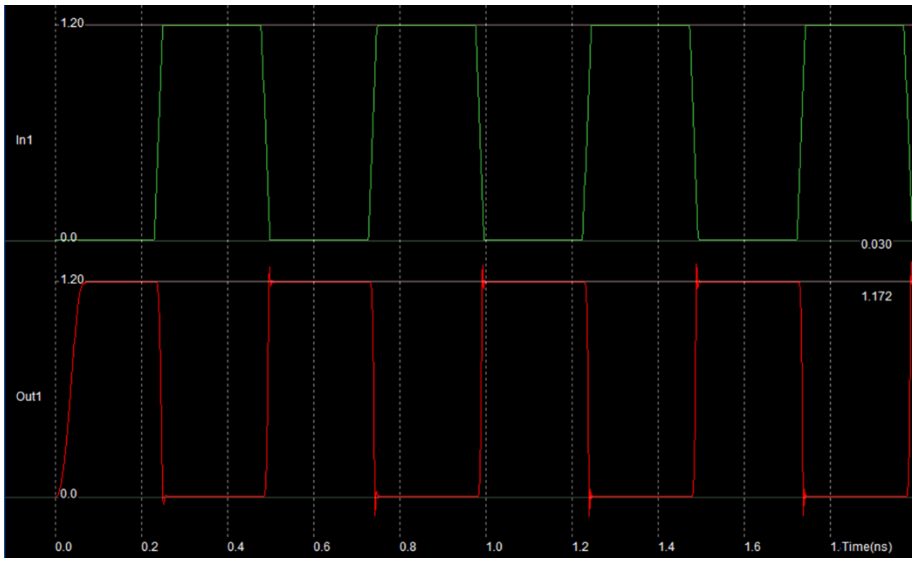

Fig. 20. Timing simulation of the CMOS not gate.

VII. Design And Simulation Of A Transmission Gate

Another simple gate with only two transistors is the transmission-gate. It is created by an nMOS and a pMOS connected on their sources and on their drains as shown in Fig. 21(a). The gate of the nMOS is connected to clock signal (Enable) and the gate of the pMOS to its complementary (~Enable). The signal DataIn is the input signal and is located on the mutual source of the devices, while the output DataOut is located on the mutual drain of the devices. Figure 21 (b) shows an example simulation where is seen that DataOut is a replica of the DataIn signal when Enable = "1". This CMOS circuit is another example of using the "best for the job" device to pass the good "0" or the good "1" at the output. Specifically, when Enable = "1", the nMOS is on and so is pMOS (because ~Enable = "0"), and when Enable = "0" the nMOS is off and so is pMOS (because ~Enable = "1").

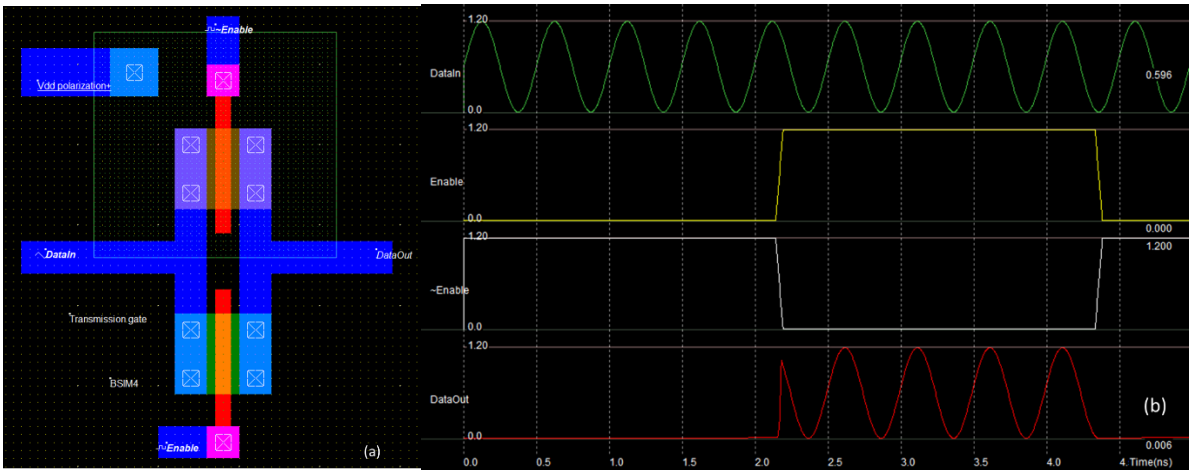

Fig. 21. (a) Layout of the transmission-gate. (b) Example simulation test for the function of transmission gate.

VIII. Design And Simulation Of An Inverter Using Verilog

Another advantage of Microwind, is its built-in Verilog-to-layout compiler. For example, using the menu Compile ➔ Compile-one-Line, and entering the equation inv = ~in and pressing the Compile button, the layout of the inverter is automatically created complying with the design rules of the selected technology rule file at the moment of compilation. The simulation properties have to be added with double-clicking on input (in) and assigning it a clock and visibility in simulation and double-clicking on output (inv) and making it also visible in simulation. Figure 22 shows the layout process and the assigning of simulation properties and Fig. 23 the simulated timing diagram verifying the validity of the circuit as an inverter.

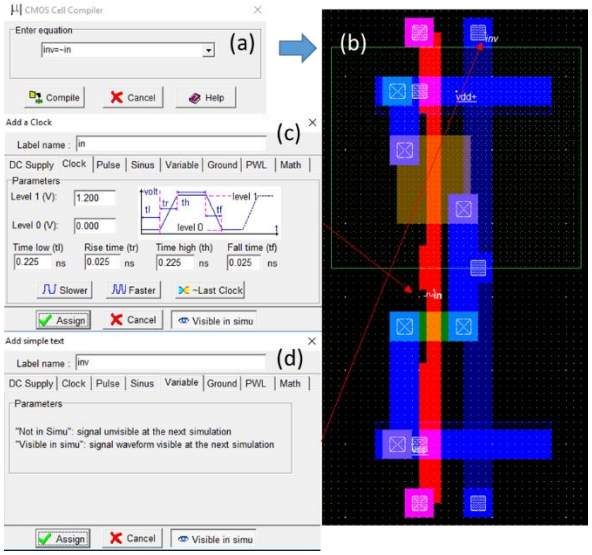

Fig. 22. (a, b) Creation of a CMOS inverter from its Verilog description. (c, d) Assigning simulation properties.

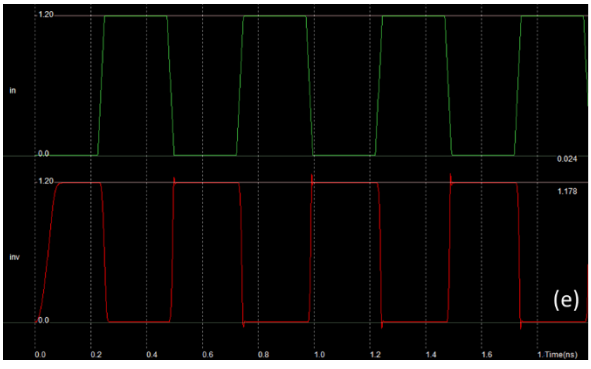

Fig 23. Timing simulation of CMOS inverter layout starting from its Verilog description.

IX. Conclusions

Students introduced to layout design for VLSI circuits are aided with the use of user friendly software tools such as Microwind. In this article the basic principles of layout design were presented along with some specific examples to help gaining deeper understanding of the design process, the basic physics of it and the benefits of simulating simple circuits. The multiple contacts and n-well polarization principles were analyzed and simple examples of simulating nMOS, pMOS, inverter, and transmission gate were presented. More advanced circuits follow the same guidelines delivered with this article.

References

[1] http://srv-wwwperso.insa-toulouse.fr/~sicard/microwind/microwind.html
[2] E. Sicard, S. D. Bendhia, *Basics of CMOS Cell Design* (McGraw-Hill, 2007).

George P. Patsis "Educational Introduction to VLSI Layout Design with Microwind " IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) , vol. 8, no. 5, 2018, pp. 18-29