TÉCNICO LISBOA

Microelectronics

Lab 2

Design of a NAND2 standard cell

Bologna Degree in Physics Engineering (LEFT)
Instituto Superior Técnico
2nd semester, 4th period

Group 4
Duarte Miguel de Aguiar Pinto e Morais Marques | 96523
Francisco Tavares Braz Mêda | 96530
José Rocha | 98839

May 25th 2022

1 Introduction

In this laboratory assignment, the main objective is to design and simulate a NAND gate with two inputs (**NAND2**) using Virtuoso (version 6.1.7). In order to do this, the libraries `STUDENTS` and `STUDENTS_sim` were created. As shown in Table 1, the cell `nand2` in the first library contains the views `schematic` and `symbol`, which will be used to run a transient simulation of the voltage in the output of the NAND2 gate - including its respective corner analysis -, as well as `layout`, in which the layout of the cell was designed and its validation was made using DRC and LVS check. These procedures will be further discussed throughout this report.

Library	Cell	View
STUDENTS	inverter1	layout
		schematic
		symbol
	nand2	layout
		schematic
		symbol
STUDENTS_sim	inverter1	transient
	nand2	transient
		transient_upgraded

Table 1: Libraries, cells and views created in Virtuoso

It is worth pointing out that similar analyses were made for an inverter logic gate, in the cells named `inverter1`, according to the instructions shown in the tutorial "Full-Custom Design with Cadence Tutorial" [1] - a transient simulation, corner analysis and the layout of the cell were made. These views were than used as examples when developing this assignment for the NAND2 gate.

Logic gates are the basis of digital circuits. They are usually made using CMOS (Complementary Metal-Oxide Semiconductor) technology and are commonly used in microcontrollers, microprocessors, among many other devices essential to our day-to-day lives. A **NAND2** gate, as the name implies, has an output which is complement to that of an AND2 gate. Thus, a '0'/False output exists only if the two inputs are both '1'/True. The ANSI/MIL (American National Standards Institute/'Military') representation of this device - with inputs IN1 and IN2 and output OUT (as they will be named in Virtuoso) - is shown below (in Figure 1) as is its respective truth table (Table 2).

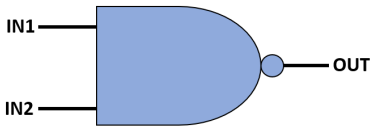


Figure 1: ANSI/MIL symbol for a NAND2 gate

IN1	IN2	OUT
0	0	1
0	1	1
1	0	1
1	1	0

Table 2: Truth table of a NAND2 gate

The inverter and the NAND gates can both be designed using CMOS technology - nMOS and pMOS transistors. The first can be made by using a pMOS transistor - connected in its source (S) to the supply V_{DD} - and an nMOS transistor - connected to ground in its source. A length of $L = 0.35\mu m$ was used for both transistors, while widths of $W = 4\mu m$ and $W = 1\mu m$ (respectively) were used. As for the NAND2 gate, two of each type of transistor must be used, as will be discussed in Section 2. The saturation (drain) current for an nMOS transistor is given by equation 1, where μ_n is the mobility of carriers (in this case, electrons) and C_{ox} is the oxide layer capacity, while W and L are the channel width and length, respectively. On the other hand, the source current in saturation for a pMOS transistor is given by equation 2 (where μ_p is the mobility of holes).

$$I_D = \frac{\mu_n C_{ox}}{2} \frac{W}{L} (V_{GS} - V_{TH})^2 \tag{1}$$

$$I_S = \frac{\mu_p C_{ox}}{2} \frac{W}{L} (V_{SG} - V_{TH})^2 \tag{2}$$

Finally, the variation of the voltage v between a capacitor's terminals is given by equation 3, where i is the current through the capacitor and C its capacitance.

$$\frac{dv}{dt} = \frac{i}{C} \tag{3}$$

2 Transient simulation

Before working on the NAND2 standard cell, a tutorial regarding an inverter [1] was repeated. Firstly, a schematic with a pMOS transistor and an nMOS transistor was drawn, as well as the symbol of the inverter, in a separate cellview. Moreover, a new library was created, in which another schematic was used to run a transient simulation, as well as a corner analysis. Finally, a layout of the inverter was created and a DRC (Design-Rule-Check) and LVS (Layout-versus-Schematic) Check were performed. A similar process was applied for the NAND2 standard cell, which will be discussed throughout this report.

2.1 CMOS Logic-Gate Circuit [2]

CMOS digital circuits utilize nMOS and pMOS transistors. A MOS transistor can operate as an on/off switch by using the gate voltage to operate the transistor. Specifically, an nMOS transistor behaves as a closed switch, exhibiting a very small resistance between its drain and source terminals when its gate voltage is "high," usually at the power-supply level V_{DD}, which represents a logic 1. Conversely, when the gate voltage is "low" (i.e., at or close to ground voltage), which represents a logic 0, the transistor is cut off, thus conducting zero current and acting as an open switch. The pMOS transistor operates in a complementary fashion: to turn the transistor on, its gate voltage is made low ($0V$/logic 0). Raising the gate voltage to V_{DD} (logic 1) turns the pMOS transistor off. The gate terminal of the MOSFET is used as the controlling node, and thus it is usually the input terminal of the logic gate.

The CMOS logic gate consists of two networks: the **pull-down** network (PDN) constructed of nMOS transistors, and the **pull-up** network (PUN) constructed of pMOS transistors. To synthesize the PDN for the **two-input NAND Gate (NAND2)**, the input combinations that require OUT to be low can be considered firstly: there is only one such combination, namely, IN1 and IN2 both high. Thus, the PDN simply comprises two nMOS transistors in series. To synthesize the PUN, the input combinations that result in OUT being high should be considered - IN1 low or IN2 low. Thus, the PUN consists of two parallel pMOS transistors with IN1 and IN2 applied to their gates. Putting the PDN and PUN together results in a CMOS NAND2 gate implementation.

2.2 NAND2 gate implementation using Virtuoso

Taking this into account, in the library `STUDENTS`, the cell `nand2` was created from the `inverter1` cell developed in the tutorial. In the view `schematic`, the two inputs NAND gate shown in Figure 2 was designed. Here, `pmos4` and `nmos4` instances were added from the library PRIMLIB. The sources (S) of both pMOS transistors were connected to `vdd` symbols from analogLib, both drains (D) were connected to the output pin (OUT) and the gates (G) to inputs IN1 and IN2. Finally, each bulk (B) was connected to its respective source.

Regarding the nMOS transistors, both gates were connected to the input pins added previously, the drain of the upper nMOS to the output and the source of the lower transistor to an instance `gnd` from analogLib. The source and drain of the upper and lower transistors were connected to each other and their bulks to ground and the respective source, respectively.

Once a CMOS gate circuit has been generated, a significant step remaining in the design is to decide on W/L ratios. These are usually selected to provide the gate with current-driving capability in both directions equal to that of the basic inverter. One way to think about this is that **connecting MOS transistors in series is equivalent to adding the lengths of their channels while the width does not change; connecting MOS transistors in parallel does not change the channel**

length but increases the width to the sum of the W's. In order to have a NAND2 gate with the **same driving strength** of the inverter - in which the pMOS and nMOS transistors had a length of $L = 0.35\mu m$ and widths of $W = 4\mu m$ and $W = 1\mu m$ (respectively), the same lengths were used for the transistors in the schematic of Figure 2 and the width of the pMOS transistor was maintained (because, in practice, usually only one pMOS transistor will work at at a time); however, the width of the nMOS was doubled to $w = 2\mu m$, in order to cancel out the effect of "doubling" the length, as was just described.

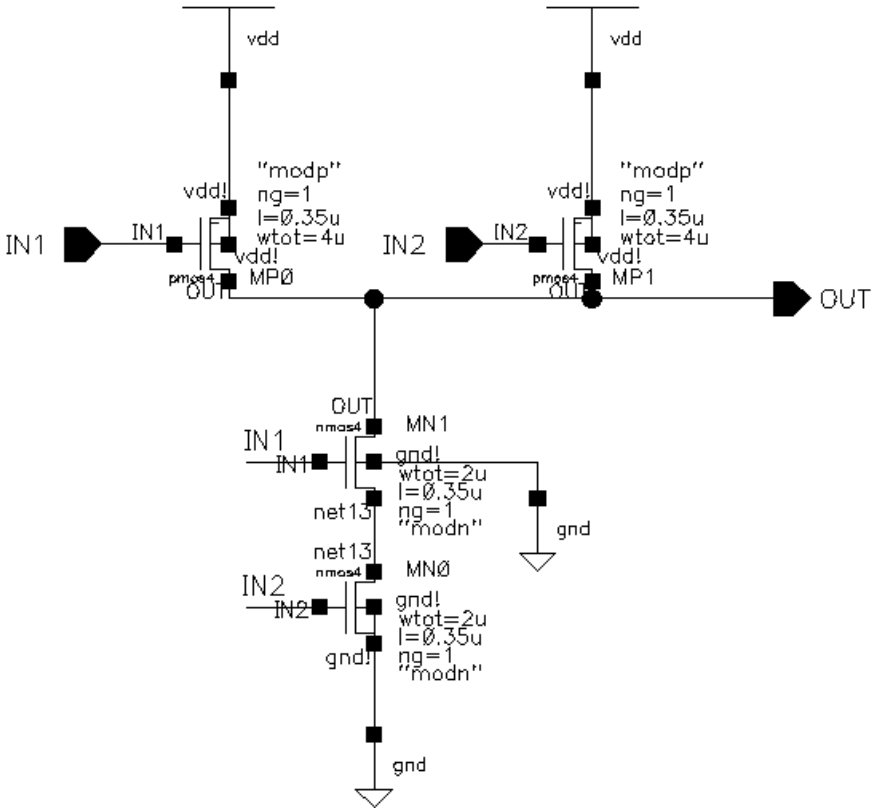


Figure 2: NAND2 gate schematic designed in Virtuoso - View `schematic` (Cell `nand2`, Library `STUDENTS`).
(Note: the arrows in the transistor symbols are not shown in the picture for limitations in the Zoom Out feature of the software.)

Having designed the schematic of the two input NAND gate, the respective symbol was also created and is shown in Figure 3.

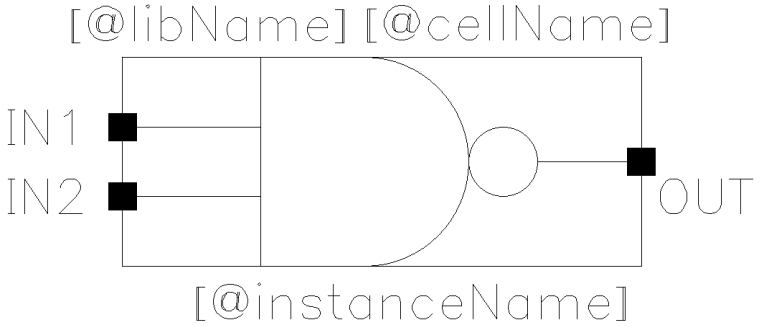


Figure 3: NAND2 gate symbol developed in Virtuoso - View `symbol` (Cell `nand2`, Library `STUDENTS`).

In the library `STUDENTS_sim` and cell `nand2`, the view `transient` was created in order to run a

transient simulation of the circuit shown in Figure 4. In this schematic, the symbol created for the NAND2 gate was added as an instance, as well as two voltage sources of type `vpulse` (rectangular signal) from the library analogLib, one connected to each input terminal. Its characteristics are specified in Table 3. The period of the signal used as an input in IN2 is double the period of the signal in IN1, in order to run the simulation for $300ns$ and have the four possible logic combinations in the two inputs and the transitions between both voltage values shown in the graphs - so that the output of the NAND gate can be evaluated (and a later corner analysis can be made). The capacitor C1 of capacitance $100fF$ is used as a load.

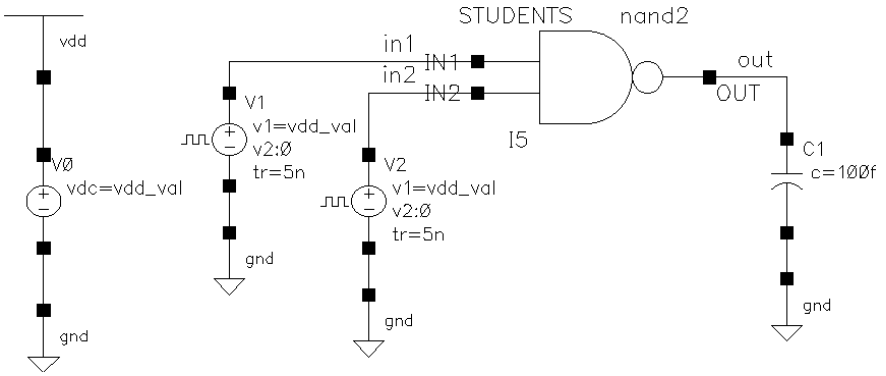


Figure 4: Circuit for simulation in View `transient` (Cell `nand2`, Library `STUDENTS_sim`), made in order to obtain the value of the voltages at IN1, IN2 and OUT over time (shown in Figure 5).

	`vpulse` **voltage source at input IN1**	`vpulse` **voltage source at input IN2**
Voltage 1	vdd val V	vdd val V
Voltage 2	0 V	0 V
Delay time	0 s	0 s
Rise Time	5n s	5n s
Fall Time	5n s	5n s
Pulse width	45n s	95n s
Period	100n s	200n s

Table 3: Characteristics of the voltage sources (pulse signals) shown in Figures 4 and 6.

In the view `transient`, the ADE GXL can be launched, from which a test is created. By selecting `vdd_val`'s value as $3.3V$, the outputs to be plotted as the voltages in `in1`, `in2` and `out` (shown in Figure 4) and the analysis as transient, for $200ns$ and setting the Accuracy Defaults as "conservative", the graphs shown in Figure 5 are obtained.

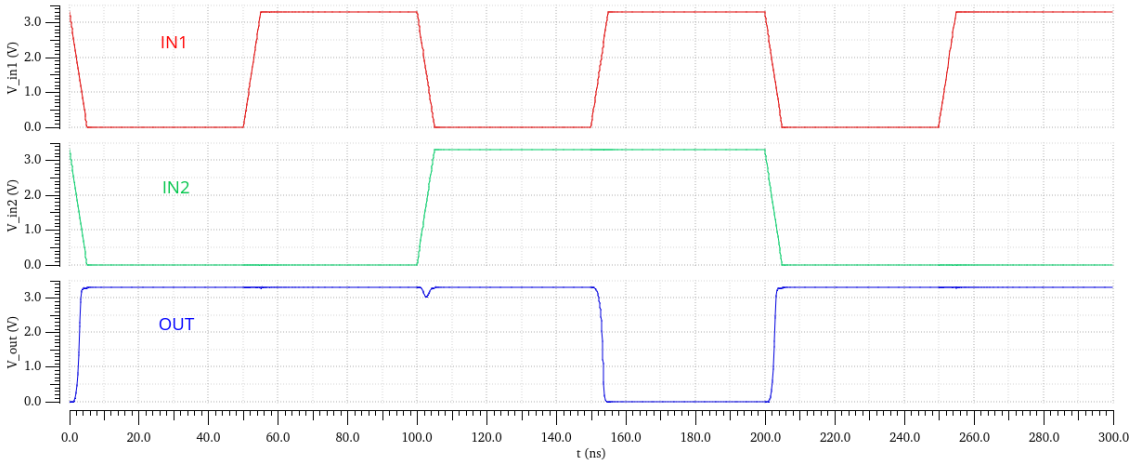


Figure 5: Variation of the voltages in IN1, IN2 and OUT (shown in Figure 4) for a simulation time of $300ns$ and with the input voltages having the characteristics shown in Table 3. Here, the voltage values of $0V$ and $3.3V$ correspond to the logic values of '0' and '1', respectively.

In this simulation, the evolution of the voltages at IN1 and IN2 was observed for 3 and 1.5 periods, respectively. By comparing Figure 5 and Table 2, it is possible to conclude that the desired behaviour of the NAND2 gate is obtained, i.e, all the combinations of IN1 and IN2 voltages give rise to a $3.3V$ output voltage ('1' in boolean), except for when there is a logic value of '1' on both inputs, which results in a $0V$ output voltage ('0' in boolean). It can be noticed in the graph of the voltage in OUT that, when both inputs change its respective values (at around $t = 100ns$), there is a slight "glitch" in the output voltage - this is not very significant to take into account in this laboratory assignment and could be fixed by decreasing the rise and fall times, for example.

2.3 Upgraded circuit

The circuit previously is very simplistic, the load is a $100fF$ capacitor and the inputs are given by ideal voltage sources. Therefore, a more rigorous test, with more realistic input signals and load, was performed. The circuit shown in Figure 6 was created in the view `transient_simulation`. Here, two inverters were included between the voltage sources and the inputs of the NAND2 gate. These symbols were the ones designed as shown in the tutorial [1] (present in view `symbol` of cell `inverter1` and library `STUDENTS`) and makes the input of the NAND2 gate more realistic; two of them were added in order to more easily compare the simulation results with the ones shown before (in Figure 5). Moreover, three other inverters were connected to the output of the NAND2 gate, as well as to a symbol noConn (from library `basic`) - this allows the circuit to have a more realistic load.

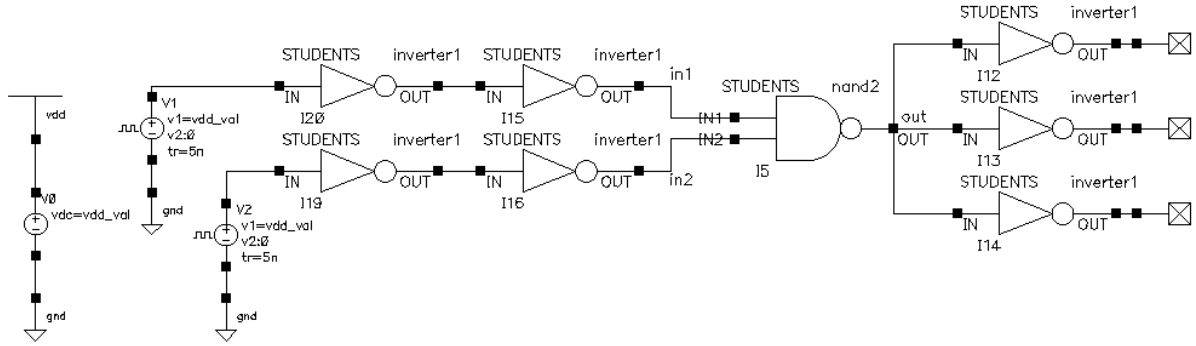


Figure 6: Circuit for simulation in View `transient_upgraded` (Cell `nand2`, Library `STUDENTS_sim`) - in order to obtain the value of the voltages at IN1, IN2 and OUT over time (shown in Figure 7).

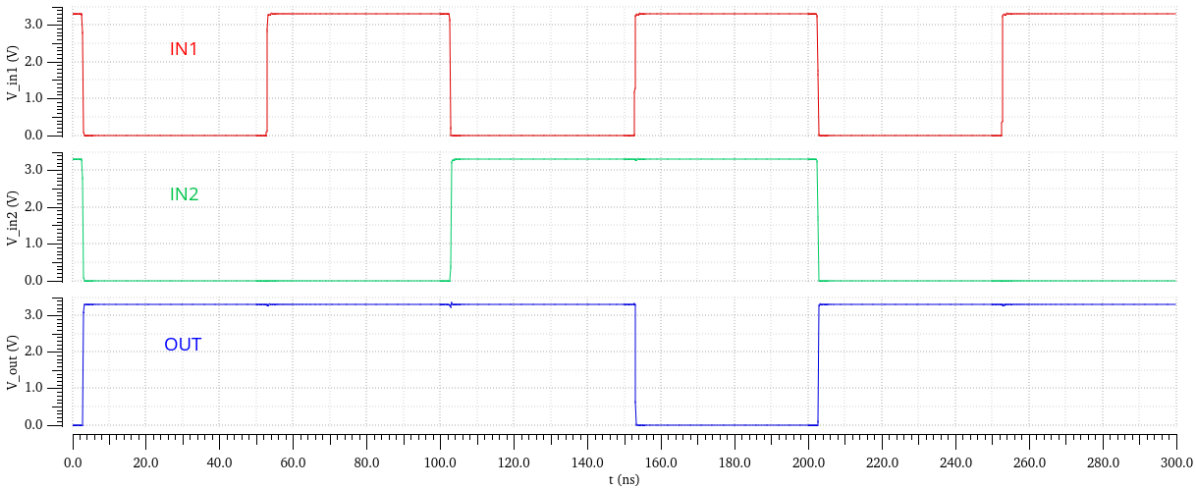


Figure 7: Variation of the voltages in IN1, IN2 and OUT (shown in Figure 6) for a simulation time of $300ns$ and with the input voltages having the characteristics shown in Table 3. Here, the voltages values of $0V$ and $3.3V$ correspond to the logic values of '0' and '1', respectively.

The graphs of the input and output signals obtained with the circuit shown above are present in Figure 7. Just as it was anticipated, the results are very similar to the ones shown in Figure 5 - the output voltage only becomes $0V$ when both the inputs are at $3.3V$. The main difference to the previous case is the bigger value of the slope (in absolute value) of the rise and fall of the input and output signals. This happens because of the inverters (which cancel the effect of the rise time inserted in Virtuoso for the voltage sources). However, a certain slope still remains, including to the fact that the inverters in the load should have a certain capacitance associated with them. Furthermore, the glitch noticed on the previous section is less noticeable - due to the rapid transitions.

3 Corner analysis

Besides the transient simulations shown before, **corner analyses** are made for both circuits in Figures 4 and 6. This type of analysis is extremely useful, since the behaviour of the circuits can be tested in **extreme manufacturing conditions**. Corner analysis is particularly effective in digital electronics, since these variations may have a significant effect on the speed of transistor switching during transitions between logic states.

In this laboratory assignment, corner analysis is used to sweep the voltage `vdd_val`, the temperature and the CMOS process parameters. In order to obtain a SDB file to import in Virtuoso's ADE, the "hitkit" option in the menubar of the initial prompt should be selected, from which "Corner Analysis" (in "Simulation Utilities") should be clicked on. Here, the variables to be chosen are **temperature** - which base value is set as 25 and corners as -40 and 125 (these values are in ^oC) - and **vdd_val** (inserted as an additional variable) - whose extreme values are $3.0V$ and $3.6V$, indicating a 10% variation below and above (respectively) its base value of $3.3V$. Moreover, the corners for **cmos** should also be selected - these are `wp`, `ws`, `wo` and `wz` (as explained in Table 5). All these extreme conditions lead to 16 different corners ($= 2 \times 2 \times 4$), corresponding to the different combinations between them.

In order to perform this analysis, in the ADE GXL (already launched), a corner is created, in which the `ams_corners.sdb` file, previously obtained, is imported. By running the simulation (option "Single Run, Sweeps and Corners"), 16 plots are obtained for the voltages in IN1, IN2 and OUT, corresponding to the 16 corners mentioned above. By plotting them all, the significant variations for the voltage in OUT (for the circuits in Figures 4 and 6, respectively) are shown in Figures 11 and 12.

The characteristics of each corner, as indicated by the ADE GXL, are specified in Table 4. Moreover, in Table 4, the four CMOS worst case parameter sets are summarized. For WO (worst case one) the mobility μ increases for the nMOS and decreases for the pMOS, while the opposite happens to the threshold voltage, V_{TH}, thus implying a faster nMOS and slower pMOS. For WZ (worst case zero), μ decreases for the nMOS and increases for the pMOS, while the opposite happens to V_{TH}, implying slower nMOS and faster pMOS. For WP (worst case power), μ increases for the nMOS and the pMOS, while the opposite happens to V_{TH}, implying faster nMOS and faster pMOS. For WS (worst case speed), μ decreases for both the nMOS and the pMOS, while the opposite happens to V_{TH}, implying slower nMOS and slower pMOS transistors.

Parameter	corner_0	corner_1	corner_2	corner_3	corner_4	corner_5	corner_6	corner_7
cmos53.scs	cmoswp	cmoswp	cmoswp	cmoswp	cmosws	cmosws	cmosws	cmosws
temperature	-40	125	-40	125	-40	125	-40	125
vdd_val	3	3	3.6	3.6	3	3	3.6	3.6

corner_8	corner_9	corner_10	corner_11	corner_12	corner_13	corner_14	corner_15
cmoswo	cmoswo	cmoswo	cmoswo	cmoswz	cmoswz	cmoswz	cmoswz
-40	125	-40	125	-40	125	-40	125
3	3	3.6	3.6	3	3	3.6	3.6

Table 4: Characteristics of each corner simulated in Virtuoso. Here, the temperatures are in ^oC and the values of `vdd_val` are in volts (V).

Designation	nMOS	pMOS
WO	$\mu \nearrow$ $V_{TH} \searrow$	$\mu \searrow$ $V_{TH} \nearrow$
WZ	$\mu \searrow$ $V_{TH} \nearrow$	$\mu \nearrow$ $V_{TH} \searrow$
WP	$\mu \nearrow$ $V_{TH} \searrow$	$\mu \nearrow$ $V_{TH} \searrow$
WS	$\mu \searrow$ $V_{TH} \nearrow$	$\mu \searrow$ $V_{TH} \nearrow$

Table 5: CMOS corners considered in the corner analysis made in Section 3 - worst case one (WO), worst case zero (WZ), worst case power (WP) and worst case speed (WS) - including the respective variations in carrier mobility μ and threshold voltage V_{TH}.

In Figure 8, the fall and rise of the voltage in the transitions around $t = 150ns$ and $t = 200ns$ (respectively) is shown for the different parameters of the CMOS technology, described in Table 5.

As it is observed in Figure 5, when the output (OUT) voltage varies from $3.3V$ to $0V$, the input IN2 remains at $3.3V$, while the input IN1 changes to $3.3V$ (a similar analysis could be made if it were input IN2 changing to '1' - with IN1 already at '1'- or if both inputs changed to '1' simultaneously; this would require different voltages in the inputs to be used). By looking at the NAND2 gate schematic in Figure 2, it is clear to see that, when this voltage in IN1 has this variation, for both pMOS transistors, $v_{SG} = v_S - v_G = 0V$ - because $vdd = v_S = 3.3V$ and $v_{IN1/IN2} = v_G = 3.3V$. Thus, both pMOS transistors are in the cut-off region, since $v_{SG} < V_{TH}$. On the other hand, when IN1 changes to the logic value '1', the nMOS transistors will not be in the cut-off region, since $v_G = v_{IN1/IN2}$ and $v_{GS} > V_{TH}$. Because of this, in the **fall of the voltage** shown in Figure 8, the **mobility** μ_n for the nMOS transistors should be analysed in order to study the value of the current.

On the other hand, in the **rise of voltage** shown in Figure 8 (below), the output voltage changes to $3.3V$ - this happens when both voltages in IN1 and IN2 change form $3.3V$ to $0V$, as obtained in the transient simulation shown in Figure 5. By looking at the view `schematic` for the NAND2 gate, when the input voltages have these values, the condition $v_{SG} = v_S = 3.3V > V_{TH}$ will be verified by both pMOS transistors, since $v_G = v_{IN1/IN2} = 0V$. As opposed to this, the nMOS transistors will be in the cut-off region, because $v_G = v_{IN1/IN2} = 0V$. Thus, in this case, the **mobility of the holes** μ_p should be taken into account when analysing these graphs.

Now, by analysing the voltage drop in the upper plots of Figure 8, the curves obtained for corners 0 and 8 (corresponding to WO and WP, respectively) have a **slope** that is (in absolute value), higher than the slope for the corners 4 and 12 (for WS and WZ, respectively). This happens because, as shown in Figure 5, the **mobility** for the nMOS transistors increases for the first pair of corners, while it is lower for the latter. The other thing to be taken into account is the effect of the different threshold voltages, V_{TH}. In **WO**, $V_{TH_{nMOS}}$ (threshold voltage for the nMOS transistors) decreases, while $V_{TH_{pMOS}}$ (threshold voltage for the pMOS transistors) increases. Since the nMOS leave the cut-off region and the pMOS enter the cut-off region, it is, in this circumstances, "easier" (meaning that a lower variation in the input voltage is required) for this to happen, since it's the conditions

$v_{GS} > V_{TH}$ and $v_{SG} < V_{TH}$ that need to be verified (respectively). As for the **WP** case, although $V_{TH_{nMOS}}$ decreases, $V_{TH_{pMOS}}$ also decreases, thus it takes longer to achieve the output voltage of $0V$ (higher fall time). In **WS**, even though $V_{TH_{pMOS}}$ increases again, $V_{TH_{nMOS}}$ also increases; for **WZ**, the variation in both values of V_{TH} is undesired for this transition - thus its name ("worst zero"). To sum up, as expected, the order WO, WP, WS, WZ can be observed in Figure 8 (above).

Finally, the transition from '0' to '1' can be analysed in a similar manner. In this case, the slopes for WZ and WP are (slightly) higher, since μ_p is higher. However, the order of the curves is now WZ, WP, WS, WO (as seen below). This happens because, now, the pMOS and nMOS transistors will now respectively be in the saturation and cut-off regions, thus a lower threshold voltage for the former and a higher threshold voltage for the latter will make it easier for the relations $v_{SG} > V_{TH}$ and $v_{GS} < V_{TH}$ to be verified. In **WZ**, $V_{TH_{pMOS}}$ decreases and $V_{TH_{nMOS}}$ increases - which is ideal; for **WP**, however, $V_{TH_{nMOS}}$ also decreases. For the **WS** situation, although the threshold voltage for the nMOS increases, the other increases as well; the worst case scenario for the transition from '0' to '1' happens in **WO**, when $V_{TH_{pMOS}}$ increases and $V_{TH_{nMOS}}$ decreases - thus its name, "worst one".

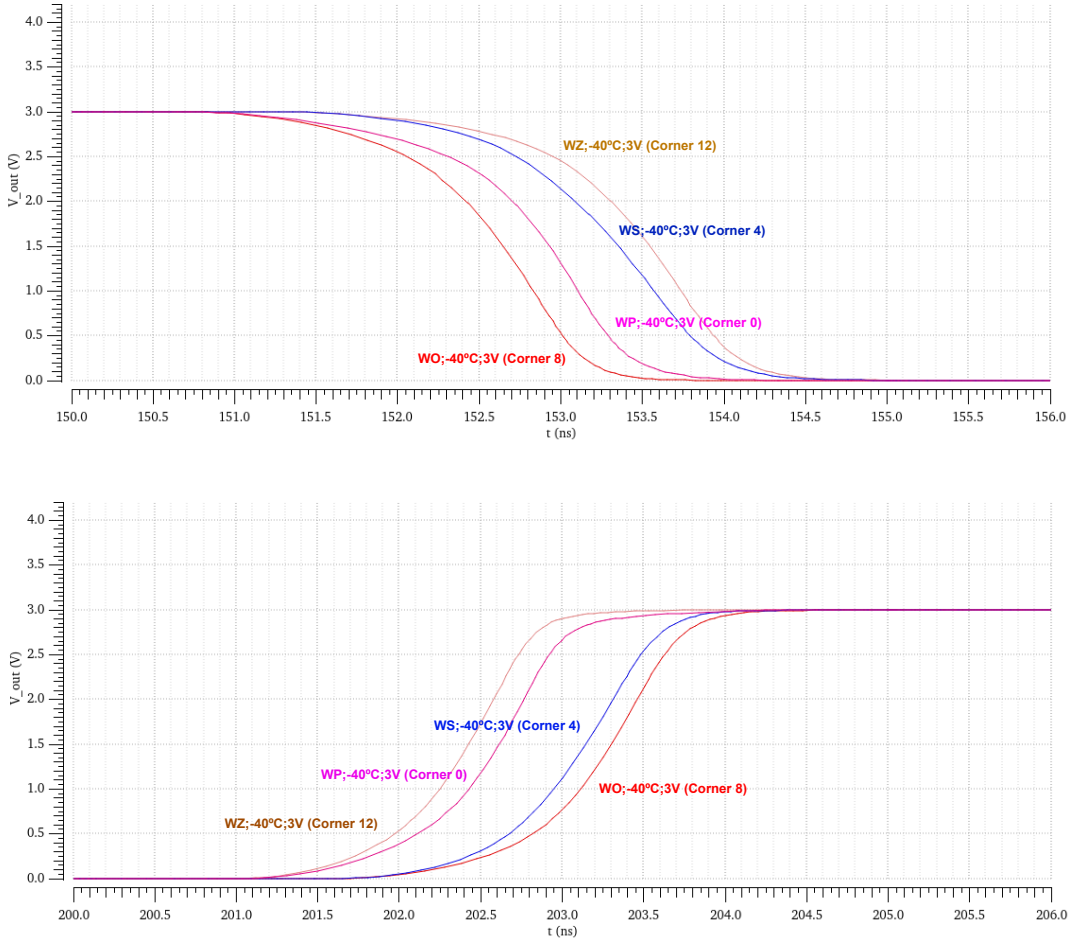


Figure 8: Value of the voltage in the output of the NAND2 gate (circuit in Figure 4) at the initial and final transitions, i.e., from logic value '1' to '0', around $t = 150ns$ [above] and from logic value '0' to '1', around $t = 200ns$ [below], for varying CMOS parameters and for corners 0, 4, 8 and 12.

The mobility of charge carriers in MOS devices increases at low temperatures. At lower temperatures, carriers move more slowly, so there is more time for them to interact with charged impurities. As a result, generally, **as the temperature decreases, the mobility increases**. Therefore, as we can see in Figure 9, as the temperature increases, the lower mobility μ leads to a lower value of the current, given by equations 1 and 2, for the nMOS and pMOS transistors, respectively. Since the slope is given by i/C (in absolute value and according to equation 3), its value becomes smaller - in

this case, the graphs corresponding to corner 0 have clearly a bigger slope (in absolute value) than those for corner 1.

In this case, even though the threshold voltages should change with the temperature, both V_{TH} - for the nMOS and the pMOS transistors - should change in the same manner, which doesn't allow us to analyse the graphs shown below through this parameter, as was done for Figure 8.

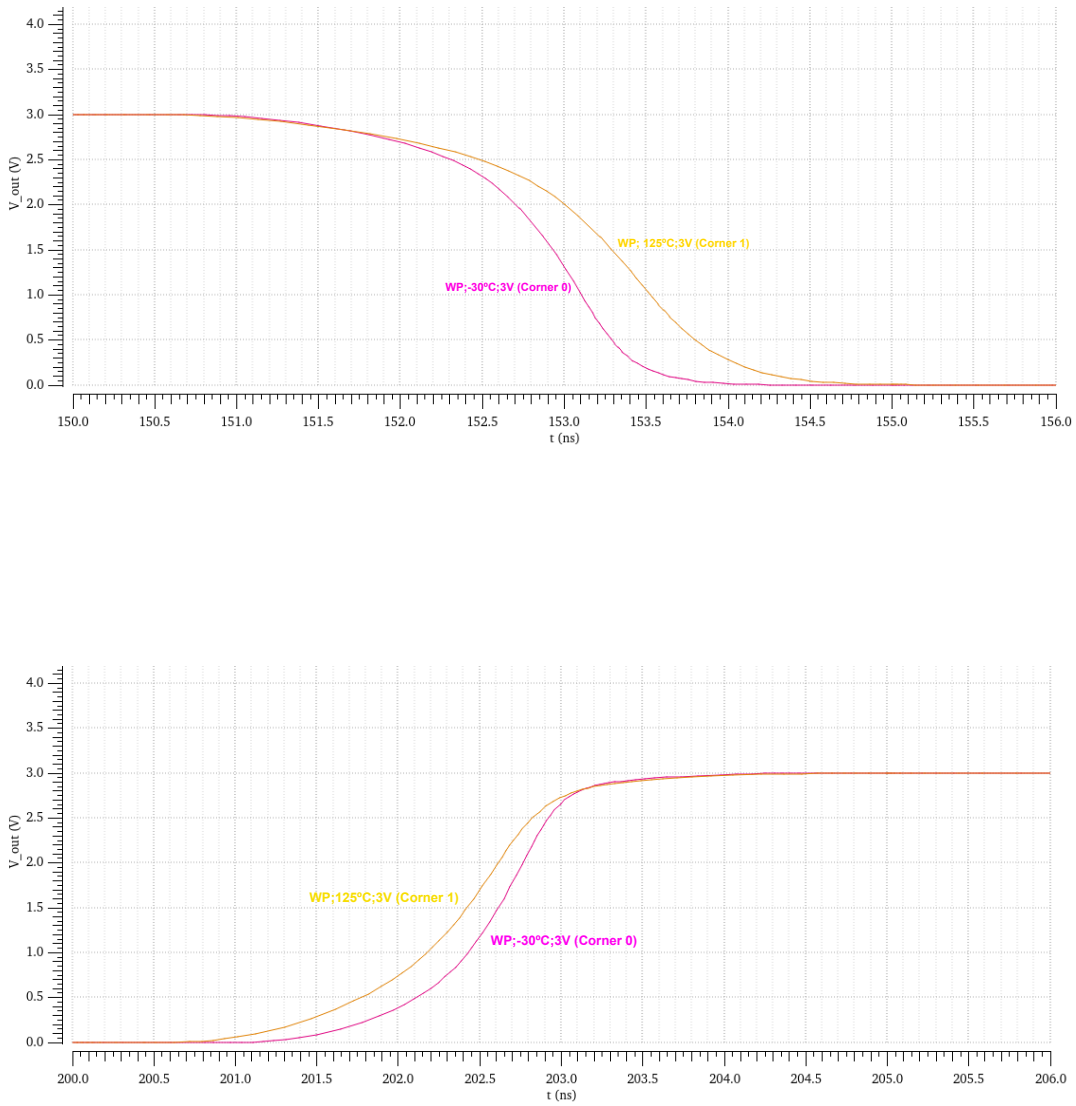


Figure 9: Value of the voltage in the output of the NAND2 gate (circuit in Figure 4) at the initial and final transitions, i.e., from logic value '1' to '0', around $t = 150ns$ [above] and from logic value '0' to '1', around $t = 200ns$ [below], for varying temperature and for corners 0 and 1.

In terms of the graphs obtained for different values of `vdd` (shown in Figure 10, for the same CMOS parameters and temperature), the rise and fall times are approximately the same; since the variation in the voltage is larger when $vdd = 3.6V$, the slope of the graphs for corner 2 (higher `vdd`) must be larger (in absolute value). This goes accordingly to equations 1 and 2 - since the voltage value in the pMOS' source (the voltage supply, as it is shown in the schematic of Figure 2) is bigger, the current's value is larger. Due to the relation given by equation 3, the slope is, therefore, also larger (in absolute value).

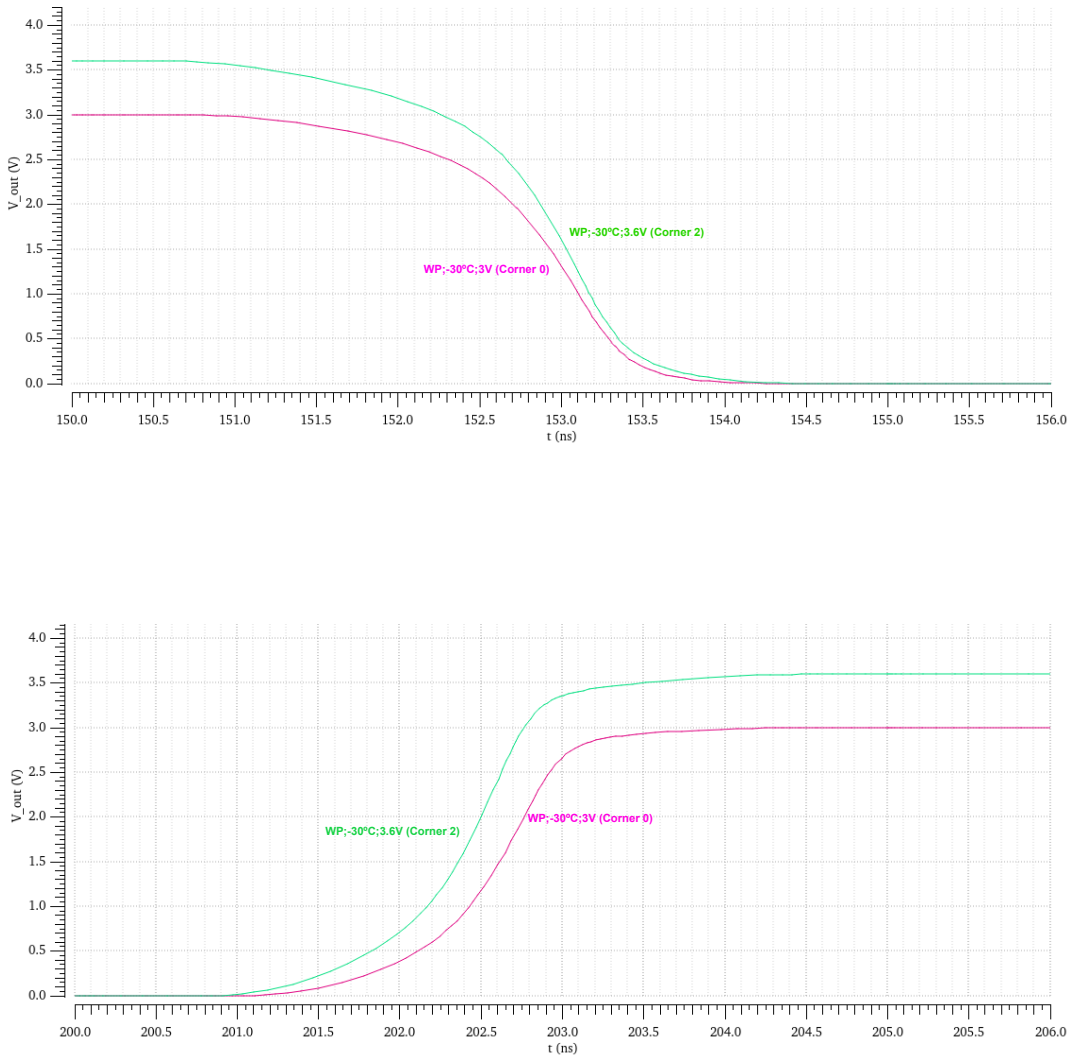
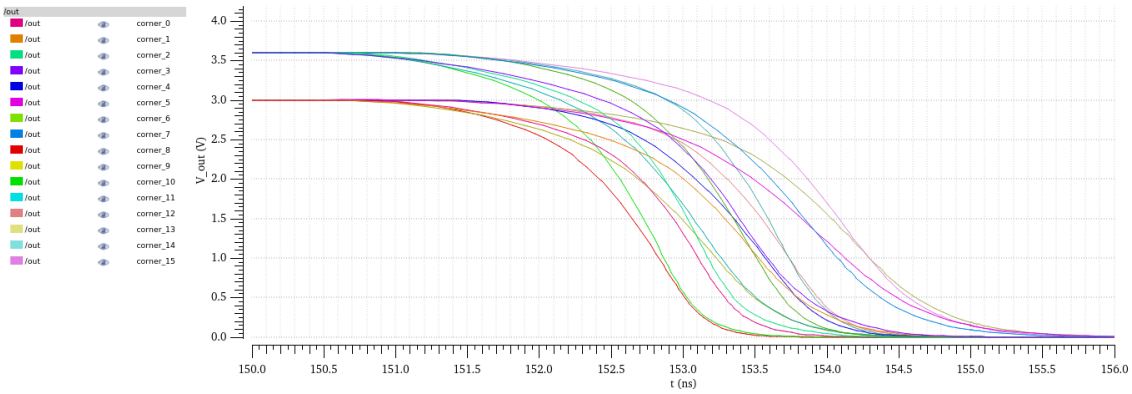


Figure 10: Value of the voltage in the output of the NAND2 gate (circuit in Figure 4) at the initial and final transitions, i.e., from logic value '1' to '0', around $t = 150ns$ [above] and from logic value '0' to '1', around $t = 200ns$ [below], for varying $\mathtt{vdd}$ and for corners 0 and 2.

In order to make the corner analysis detailed in this section, only selected curves (for selected corners) were used - a certain parameter (CMOS characteristics, temperature or voltage supply) was analysed at a time, while the others were fixed for each group of curves. The conclusions made with Figures 8, 9 and 10 can be extended to the other curves (for different combinations of the parameters in corner analysis) - these are shown in Figure 11. The different effects, described above, will act simultaneously and lead to the 16 different curves (for each transition) shown below.



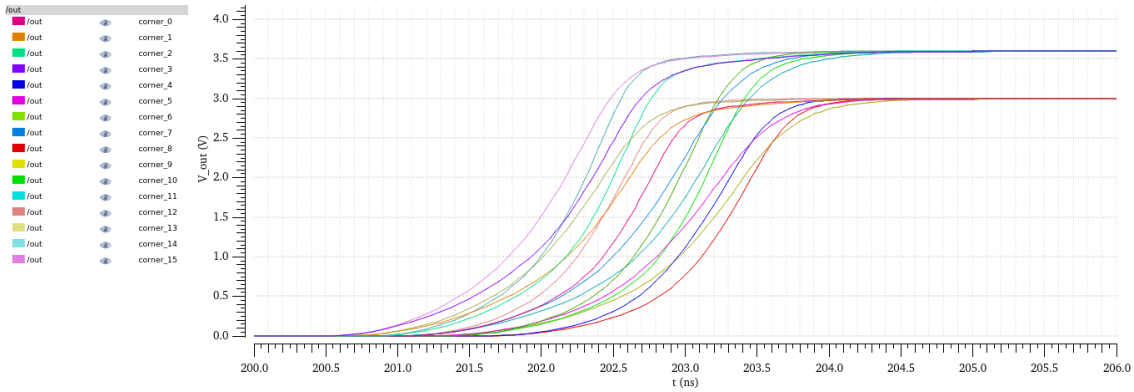


Figure 11: Value of the voltage in the output of the NAND2 gate (circuit in Figure 4) at the initial and final transitions, i.e., from logic value '1' to '0', around $t = 150ns$ [above] and from logic value '0' to '1', around $t = 200ns$ [below], for corners 0 to 15.

3.1 Upgraded circuit

The same corner analysis simulation was made for the upgraded circuit shown in Figure 6 and the curves of the voltage in the output near $t = 150ns$ and $t = 200ns$ (respectively) are shown in Figure 12. By comparing these results with the ones shown in Figure 11 (obtained for the other circuit), it is possible to see that the relative position of the curves between each other is the exact same as for the upgraded circuit's corner analysis. The only significant difference is that the slopes are higher (in absolute value), with the rise and fall times being much smaller. As it was explained in Section 2, this is due to the fact that the capacitance due to the inverters used in the inputs and as a load. The analysis made throughout this section is thus entirely similar to the one presented before - for this reason, only the graphs with for the 16 corners (plotted simultaneously) are shown below.

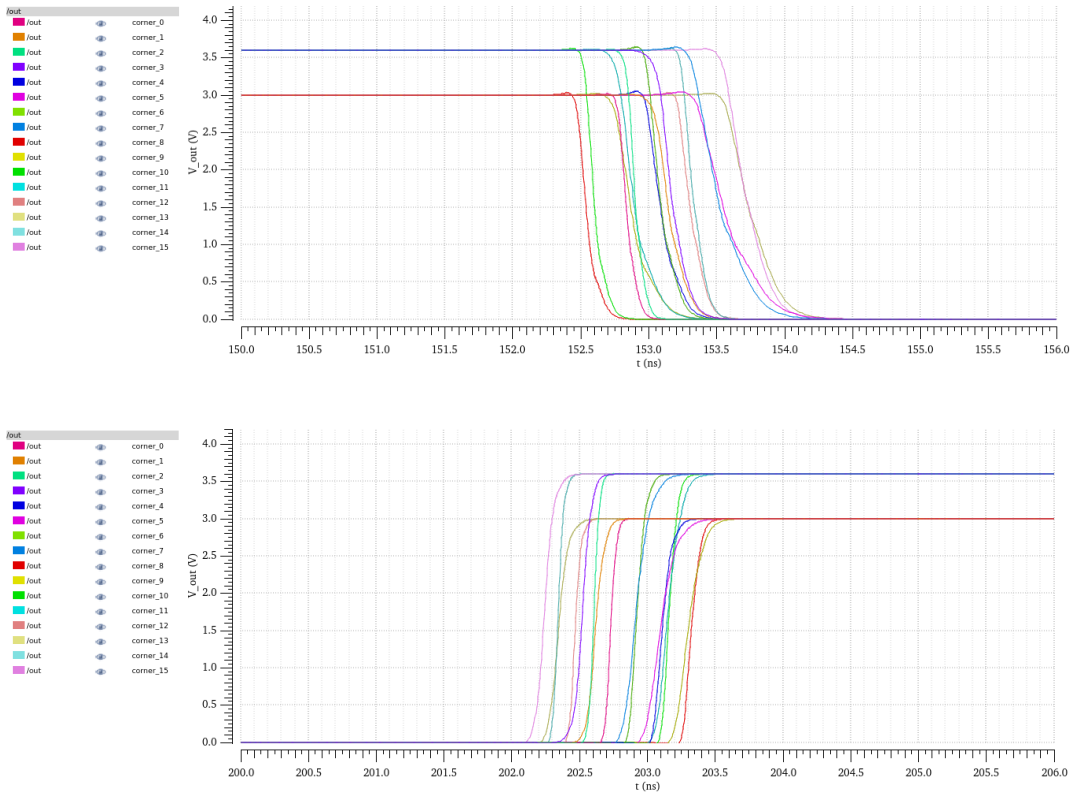


Figure 12: Value of the voltage in the output of the NAND2 gate (circuit in Figure 6) at the initial and final transitions, i.e., from logic value '1' to '0', around $t = 150ns$ [above] and from logic value '0' to '1', around $t = 200ns$ [below], for corners 0 to 15.

4 Layout

The final objective of this laboratory assignment is to design the **layout** of the NAND2 gate, formed by the two pMOS transistors in parallel and the two nMOS transistors in series (as shown in Figure 2). In order to do this, the view `layout` in `inverter1` (library `STUDENTS`) was copied into the cell `nand2` of the same library. This schematic view is thus operated in mode Layout XL. Most layers appear in the LSW window with purpose `drawing` or `pin`. The X and Y Snap Spacing are set to 0.025 and the Hierarchy Depth in DRD Options to Current&Below (in order to avoid violation of design rules).

The gnd! and vdd! pins are placed over the rectangles of metal 1 in the bottom and top of the cell, respectively. Both those pins, as well as IN1, IN2 and OUT must have their respective labels present in the layout. The connections between the following elements in the layout are performed with the layer MET1 (Metal 1), with purpose `drawing`:

- pMOS transistors' sources and vdd!;
- drain of the nMOS transistor on the left (in Figure 13) and the drain of the pMOS transistors;
- source of the other nMOS and gnd!.

On the other hand, the connections between the gates of the pMOS and nMOS are done in the layer POLY1 (purpose `drawing`). Moreover, the OUT pin shown in Figure 13 is placed over the MET1 connection between the drains of the transistors (accordingly to the NAND2 schematic designed previously); the pins IN1 and IN2 (also shown in the schematic in Figure 2) are each placed over one of the connections between a pMOS' gate and an nMOS' gate (according to the schematic designed previously). In order to connect the n-well to vdd!, the n-well included in the pMOS transistor has also been extended (on the top of the layout), by selecting the layer NTUB (purpose `drawing`) and drawing a rectangle.

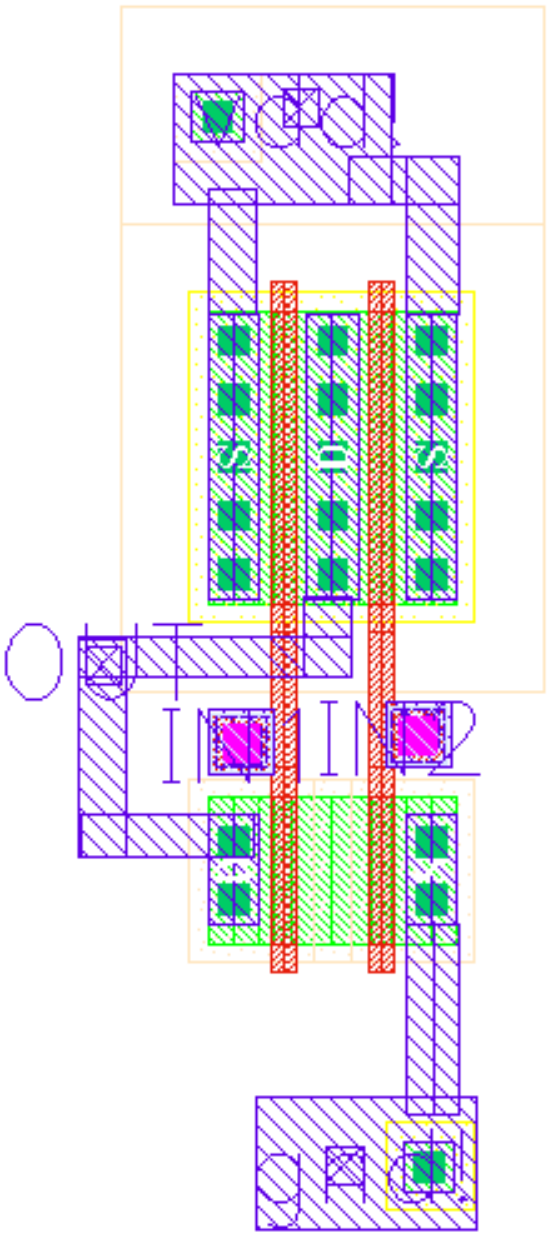


Figure 13: Layout of the NAND2 cell in View `layout` (Cell `nand2`, Library `STUDENTS`).

In order to reproduce the pMOS transistors in parallel in the layout, a single instance is used in Virtuoso, with a width of $8\mu m$ and 2 gates (thus, a width strip of $4\mu m$; the length was maintained as $2\mu m$), since having two transistors in parallel is analogous to having the width summed. The gates, sources and drains were not joined and both the top and bottom contacts have not been removed - these properties can be altered by pressing <q> (Edit Instance Properties) over the transistor. On the other hand, two nMOS transistors were used, each having a width and width strip of $2\mu m$, a length of $0.35\mu m$ and 1 gate. The bottom contacts were maintained, the top contacts were removed and the gates (on the right), sources and drains were joined.

Finally, the **Design Rule Check (DRC)** program was ran in order to verify if the design rules were followed and report any possible violations. To do this, `Run DRC` is selected in the option `Assura` of the top menu bar. Having accepted the default values of the process, the errors were listed as shown in Figure 14. These are not significant and could easily be avoided by altering the DRC options form; moreover, they only make sense in the scope of the whole layout of the chip - where they must be treated as severe errors. The **Layout Versus Schematic (LVS)** verification was also performed, by selecting `Run LVS` with its default values in `Assura`. This will compare the circuit in the view `schematic` of the cell `nand2` with the one extracted from this mask layout to prove that both networks are equivalent, providing more confidence on the designs' integrity and that the layout is an adequate realization of the circuit. It is to be noted, however, that this only verifies the topological match - it won't guarantee that the mask layout of the cell will actually satisfy the performance requirements [1]. The results of this check are also shown in Figure 14: the message "Schematic and Layout Match" is shown, thus no errors (such as unintended connections between transistors, missing connections/devices, etc.) require correction in the mask layout.

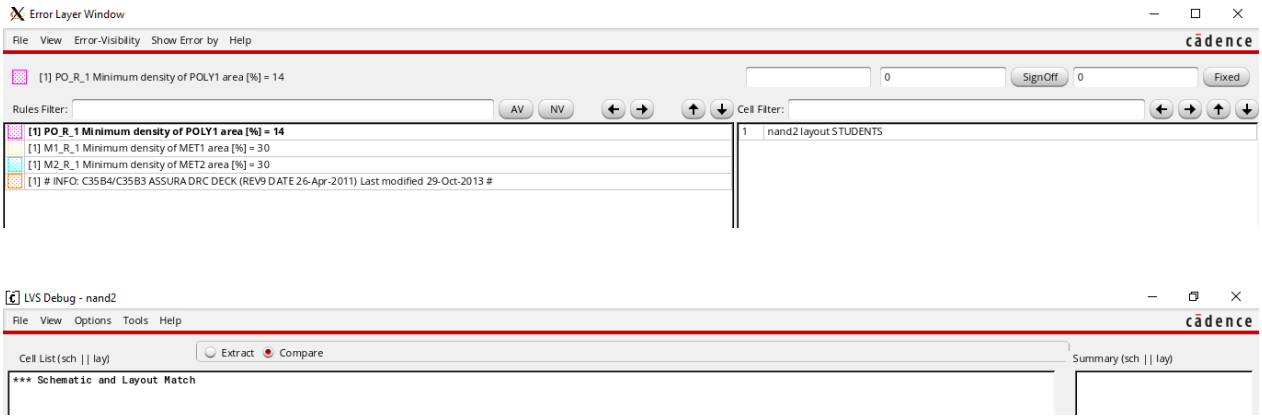


Figure 14: Result of the validation of the layout shown in Figure 13 by performing DRC (Design Rule Check) [above] and LVS (Layout Versus Schematic) [below] with their respective default options.

5 Conclusion

The objectives proposed for this laboratory assignment have been successfully fulfilled. Having started by designing a schematic to simulate an inverter and the layout of this logic gate, a similar procedure was applied for a NAND2 gate. By using two pMOS transistors in parallel and two nMOS transistors in series, the schematic of the NAND gate with two independent inputs was simulated using Virtuoso. In the transient simulation, the expected behaviour of the NAND gate was verified - the logic table shown in Table 2 was graphically obtained. Moreover, an upgraded circuit which used three inverters (instead of a capacitor) as a load was utilized in the same manner, having also originated the expected results (with a few changes from before). Additionally, a corner analysis was performed, in which these circuits' behaviour was evaluated for extreme conditions of temperature, voltage value in the inputs and technological CMOS parameters.

The layout of the NAND2 gate was also designed in Virtuoso, keeping the supply rails size and position of the inverter cell. This layout was validated using both DRC (Design-Rule-Check) and LVS (Layout-versus-Schematic) Check, having the expected errors been highlighted by the former and a match between schematic and layout been indicated by the latter.

References

[1] Marcelino Santos. *Microelectronica - Full-Custom Design with Cadence Tutorial*. AustriaMicroSystems C35B4 (HIT-Kit 4.0), Virtuoso 6.1.3.

[2] Adel S. Sedra; Kenneth C. Smith. *Microelectronic circuits*. 7th ed. Oxford University Press, 2015.