NTNU
Norwegian University of
Science and Technology

Subthreshold CMOS Cell Library by 22 nm FDSOI Technology

Stian Østerhus

Master of Science in Electronics
Submission date: June 2018
Supervisor: Snorre Aunet, IES
Co-supervisor: Trond Ytterdal, IES

Norwegian University of Science and Technology
Department of Electronic Systems

NTNU – Trondheim
Norwegian University of
Science and Technology

TFE4915 - Analog Circuit Design and Radio Systems

Subthreshold CMOS Cell Library by 22 nm FDSOI technology

Master's Thesis

Stian Østerhus

June 7, 2018

1 Abstract

Two different CMOS transistors with a low threshold voltage, given by a commercial available 22 nm FDSOI CMOS technology were investigated and assembled into several libraries of logic gates. The logic gates provided in the cell library should be sufficient to create most digital logic circuits, and are in addition designed to work in the subthreshold region with a supply voltage of 350 mV. Physical layout designs were made for the different digital ports, where parasitic capacitances were then extracted to provide more realistic simulations and performance results. Compared to schematic simulation, layout design and parasitic capacitances proved to reduce speed by a factor of 5 to 10, as well as increasing the transistors' threshold voltage by 14.6 % for the NMOS, and 32.5 % for the PMOS. The increased threshold voltage thus led to a reduced static power consumption and increased switching energy.

The transistor with the lowest threshold voltage showed especially good performance results with respect to low power consumption while still maintaining speed requirements. This transistor is throughout the report referred to as ***mosfet_low***. Two cell libraries were made for this transistor, where one applies a forward body-bias of ±2 V while the other have the bulk nodes connected to ground, which gives a 0 V body-bias. The libraries are supplied with schematics and layout designs, and are in addition mapped for performance data such as static power consumption, delay and switching energy consumption for every logic gate.

A minimum speed of 40 MHz with a lowest possible power consumption for a 16by12-bit adder, was the aim of the project. Presented in this report is a 16by12-bit Adder built by Ripple-Carry Adders, which were simulated to reach a speed of 44.26 MHz at a supply voltage of V_{DD}=350 mV with 0 V body-bias. Static power and switching energy consumption were simulated to 26.60 μW and 207.95 fJ, respectively.

Acknowledgements

I would like to thank my supervisor, professor Snorre Aunet, and co-supervisor professor Trond Ytterdal, for great advices, guidance and patience throughout the project.

Contents

2 Introduction

Though Moore's law seems to be reaching it's end in the coming years, computer performance still increases regularly [1]. The downscaling of transistor dimensions increase the potential for increased performance, but also introduces several challenges like increased chip power dissipation, leakage currents, difficulties related to lithography accuracy among many more challenges [2] [3].

It is generally assumed that other creative and new methods will compensate for the physical limit of downscaling transistor sizes in the near future. An alternative method, which is highly researched today, is stacking transistors in several layers above each other, also called *3D chip technology* [4] [5]. Though this does not remove the heat dissipation issue, which becomes more significant as operation frequency increases and more transistors are placed on the same area. To accommodate heat related issues, as well as power consumption, one may design the circuits for subtreshold operation [6] [7].

Whatever the solution to the downscaling challenge will be, problems related to heat dissipation, and power consumption due to the higher performance still apply more than ever. In addition, advancements in battery technology does not exponentially follow the progress of integrated circuits. This gives complications when designing battery powered electronics, like mobile phones, laptops or sensor networks. To accommodate a reasonable battery time with respect to performance, compromises must be made; like increasing the battery capacity or restricting several functions of the device. Heat development is also an increasing problem when scaling down transistor sizes while increasing performance. An approach to accommodate these challenges are as mentioned subthreshold design, which essentially is reducing the supply voltage of the system. As power consumption in CMOS circuits is an exponential function of the supply voltage, reducing the supply voltage may result in significantly decreased power consumption [8] [9].

The aim of this project is to further investigate the properties of subthreshold- and near-subthreshold circuits, and develop digital logic circuits with high emphasis on low power dissipation, while maintaining a certain performance. Circuit schematics and layout designs are organized into blocks with common height, enabling synthesizing into larger systems of a desired function. The complete library will consist of the most common building blocks for digital design, supplied with data describing performance properties. Design will be performed by applying a commercial available 22 nm FDSOI CMOS technology, using *Cadence Virtuoso* for simulation and layout design. Parasitic capacitances introduced by layout design will also be extracted by *Cadence Virtuoso*, while layout design will be checked for errors with DRC and LVS by *Mentor Graphics Calibre.*

2.1 Project requirement specifications

The purpose of this project is to investigate and develop a CMOS library for medical ultrasound applications, also investigated in [10]. For the specific application, a speed of minimum 40 MHz is required, followed by a lowest possible power consumption. All the logic gates must fulfill the time restraints, where the most complex gate, a 16by12bit-adder will most likely have the greatest delay and therefore determine the lowest possible supply voltage.

These requirements suggest a near-subthreshold system, or ideally a subthreshold system as an appropriate technology to build on.

2.2 Content organization

This report is divided into 4 main sections, which is *theory*, *methods*, *results* and *discussion*. The theory section provides some relevant and essential background theory for the project. The method section describes the methods used to gather results and data, and the purpose of collecting them. The results section describes the results extracted in the methods section.

Nothing is reflected upon before the discussion section, where results are discussed and compared with expected results.

3 Theory

3.1 CMOS

CMOS, which is an abbreviation for *Complementary Metal Oxide Semiconductor*, is a method of applying transistors in integrated circuits and was patented in 1963 by Frank Wanlass [11]. The technology may be applied for logic circuits, data converters and amplifiers, though in this project the CMOS transistors are used for digital logic circuits only. The term *complementery* originates from the method of applying n- and p-type MOSFETs in a symmetrical and complementary pattern, dependant of the logic function. The main advantages of applying MOSFETs in this manner, is the properties of low power consumption and higher noise immunity. Low power consumption is where CMOS technology really stands out, especially when applied in digital computation circuits [12].

CMOS transistors are mainly characterized by their ability to conduct current from drain to source when voltage is applied to the gate node. The general equation for CMOS transistors under normal conditions are given by equation 1 [13].

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

Where μ_n is the majority carrier mobility, C_{ox} is the gate capacitance and W and L is respectively transistor width and gate length. V_{GS} is the voltage between the gate- and source node on the transistor, and V_{th} is the transistor's threshold voltage.

However, this is under normal circumstances. CMOS transistors can operate in three different modes; Weak inversion, moderate inversion and strong inversion. This may also be referred to as subthreshold operation, triode region and active region respectively.

3.2 Subthreshold operation

The state of weak inversion on a silicon structure was initially mentioned in 1955 [14], though no further research for this specific topic was done. Subthreshold operation of MOSFET circuits was later discovered in the late 1960s [15], and further investigated in the 1970s [16] [17]. In general, the threshold voltage is given as the boundary value for when a transistor starts to conduct current from drain to source. Still, some leakage current occurs through the transistor below the threshold value, due to the weak inversion [18]. Therefore one can say that in subthreshold circuits, some of the transistors are more "off" than others, or "off" and "almost on", and therefore still may function as a logic circuit [19]. Subthreshold circuits are therefore defined by operating with a supply voltage, $\mathrm{V_{DD}}$, lower than the transistors' threshold voltage. The subthreshold operation area is illustrated in figure 1.

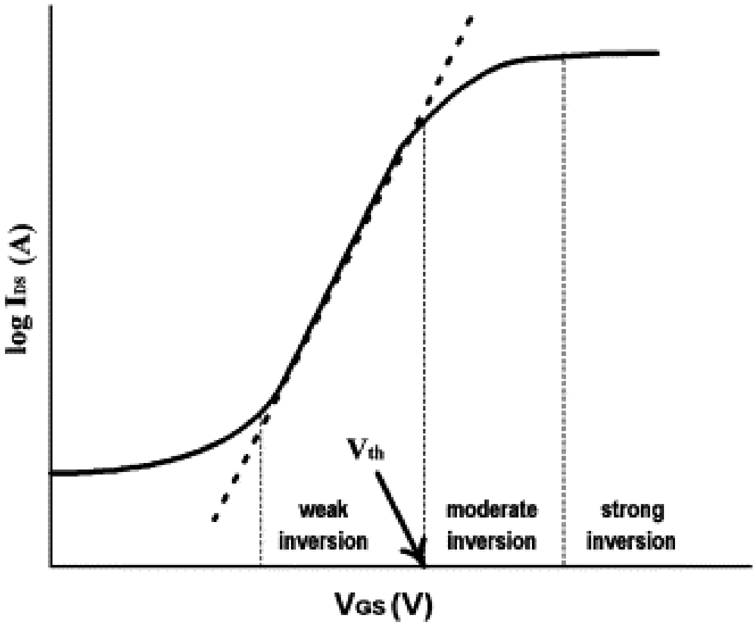


Figure 1: Subthreshold operation is shown in the weak inversion area, below the transistor's threshold voltage, V_{th}. Illustration from [20].

In 1977, Eric Vittoz and Jean Fellrath suggested to exploit the weak inversion effect rather than diminish it [21]. Subthreshold circuits were also by then adopted by Swiss electronic wristwatch manufacturers in the 1970s to extend the battery life of their products [22]. This is generally the motivation behind subthreshold circuits today, as battery technology does not keep up with the integrated circuit performance development. So instead of increasing the battery capacity, one may instead reduce the active circuit's power consumption, which is given by equation 2. This will not only reduce the power consumption, but also heat dissipation of the circuit which is an increasing problem when scaling down transistors.

$$P_{dynamic} = N * C * f * V_{DD}^2 * \alpha \tag{2}$$

The equation gives the general power consumption (dynamic power) of an active CMOS circuit, where N is the the number of transistors, C is the average load capacitance, f is the operation frequency and V_{DD} is the supply voltage. α is the activity factor of the circuit, which is given by a number from 0 to 1. From this equation, one can see that the first and most effective step to minimize energy consumption, is to reduce the supply voltage, V_{DD}.

The current flowing through a CMOS transistor operating at subthreshold voltages, or weak inversion, is shown in equation 3 [23].

$$I_D = I_{D0}(\frac{W}{L})e^{\frac{v_{GS}-v_{th}}{n*v_T}} \tag{3}$$

W and L is respectively the width and length of the transistor channel. V_{GS} is the voltage between the gate and source terminal of the MOSFET, and V_{th} is the MOSFET's threshold voltage. n is the relation between the capacitances $\mathrm{C_{ox}}$ and $\mathrm{C_{j0}}$, which is given by equation 4. $\mathrm{I_{D0}}$ is the current given by equation 5.

$$n = \frac{C_{ox} + C_{j0}}{C_{ox}} \tag{4}$$

$$I_{D0} = (n-1)\mu_n C_{ox} V_T^2 \tag{5}$$

v_T is the thermal voltage, which is defined by equation 6.

$$v_T = \frac{kT}{q} \tag{6}$$

For the thermal voltage, k is Boltzmann's constant, q is the elementary charge of $1.602\text{x}10^{-19}$ C, and T is the environmental temperature in Kelvin.

3.3 Power consumption

The total power consumption in an integrated CMOS circuit is given by several contributions, where dynamic power makes the most significant impact [24]. The sum of all the power contributions are given by equation 7 [25].

$$P_{total} = P_{static} + P_{dynamic} + P_{Shortcircuit} \tag{7}$$

It is the dynamic power which accounts for the greatest consumption, and therefore is the most important part to reduce and optimize.

3.3.1 Dynamic power consumption

Dynamic power, as mentioned earlier, accounts for the most power consumption, and is given by equation 2. The power consumption is given by the supply current dissipated in the load and is illustrated in figure 2.

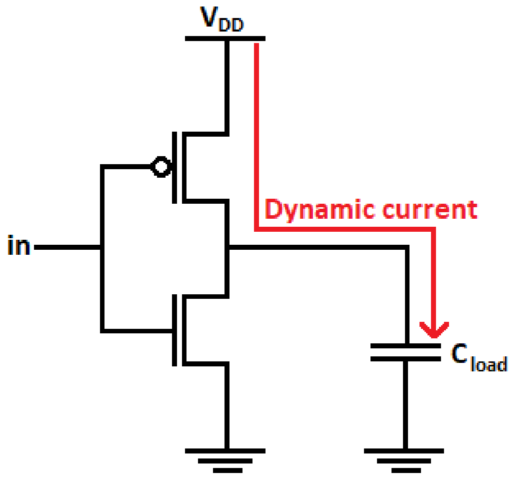


Figure 2: Dynamic current, $\mathrm{I_{dynamic}}$, flowing from $\mathrm{V_{DD}}$ to the circuit's load.

Equation 2 estimates the general switching power for a complete integrated circuit consisting of N transistors, but the energy, $\mathrm{E_{load}}$, stored in the load capacitance, $\mathrm{C_{load}}$, is given in equation 8 [25].

$$E_L = \frac{C_{load} V_{DD}^2}{2} \tag{8}$$

This shows that half of the energy drawn from the power supply is dissipated as heat, while the other half is stored in the load capacitance.

3.3.2 Static power consumption

Static power consumption is the power consumed by current leaking through transistors in their off-state, and is illustrated in figure 3.

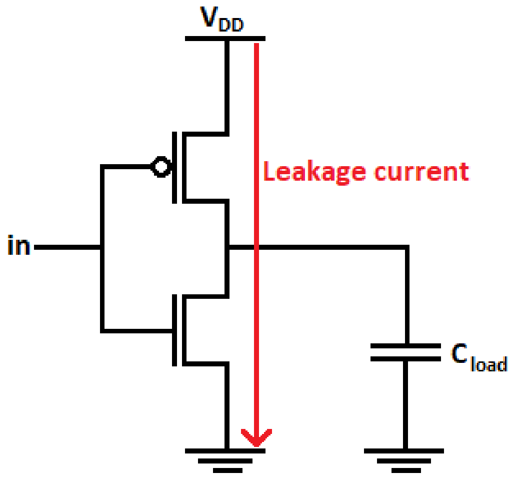


Figure 3: Leakage current, $I_{leakage}$, flowing from V_{DD} to ground.

Though dynamic power is the most significant contribution, leakage currents becomes more important to note as transistor sizes are scaling down [3]. By looking at equation 3, one can see that the leakage current increases proportionally to a decreasing gate length. The static power dissipation is given by several contributions, as seen in equation 9 [13].

$$P_{static} = (I_{sub} + I_{gate} + I_{junct} + I_{contention})V_{DD} \tag{9}$$

The leakage currents combined though, may be given by equation 10 [13], and multiplied with the supply voltage.

$$I_{leakage} = I_{sub} + I_{gate} + I_{junct} + I_{contention} = I_{ds0}e^{\frac{V_{GS}-V_{th}+\eta V_{ds}-k_{\gamma}V_{sb}}{nV_{th}}}(1 - e^{\frac{-V_{ds}}{v_T}}) \tag{10}$$

For this equation, n is a factor which describes the depletion region characteristics and usually lies between 1.3 and 1.7 [13]. v_T is the thermal voltage given by equation 6 and I_{ds0} is the current from drain to source at the transistor's threshold voltage.

3.3.3 Short circuit power consumption

The short circuit power consumed in CMOS circuits are drawn when both the NMOS- and PMOS transistors are active, and can be seen in figure 11.

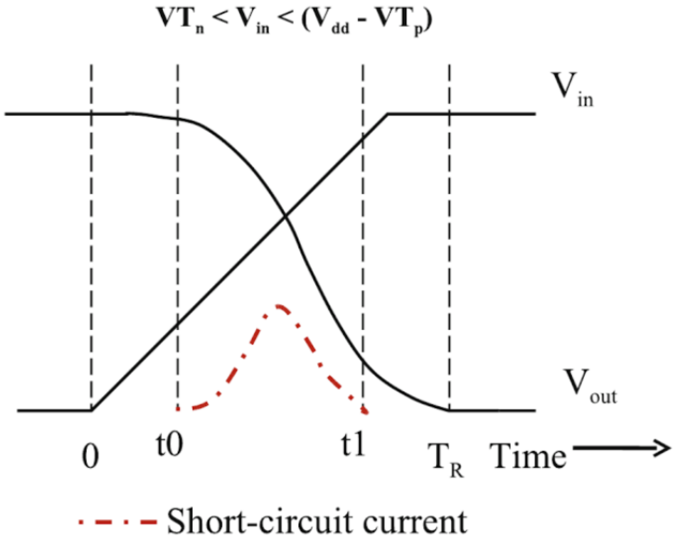


Figure 4: Short circuit current, occurring when both the NMOS- and PMOS transistors are active, leading current directly from the power supply to ground. *t0* and *t1* indicates the threshold voltages of the NMOS- and PMOS transistors, where they respectively start and stop to conduct current. Illustration from [25].

The short circuit power consumed is usually small compared to static- and dynamic power consumption given in equation 7. The short-circuit power consumption may though be estimated by equation 11 [25].

$$I_{shortcircuit} = \frac{1}{12} k \tau F_{clk} (V_{DD} - 2V_{th})^3 \tag{11}$$

Where k is the gain factor of the transistor, τ is the rise and fall time and $\mathrm{F_{clk}}$ is the operating frequency. $\mathrm{V_{DD}}$ and $\mathrm{V_{th}}$ is respectively supply- and threshold voltage.

3.3.4 Power Delay Product

Power Delay Product (PDP) is a figure of merit (FOM), and gives a good indication of how effective the circuit is with respect to speed and power consumption. PDP is given by equation 12.

$$PDP = P_{consumption} * T_{Delay} = P_{consumption} * \frac{1}{f} \tag{12}$$

P is the power drawn from the supply by every switching transition for a logic gate or transistor in watts, and T_{delay} is the time delay in seconds for when the output reaches $\mathrm{V_{DD}}/2$ from either 0 V or $\mathrm{V_{DD}}$. f is the operating frequency, or the inverse of the time delay. A lowest possible value of PDP is therefore desired.

Most logic gates may also be connected into a ringoscillator, connecting the output of one logic gate to the input of the next similar logic gate. With an odd number of devices in a chain consisting of N devices, the circuit will start to oscillate and the PDP can be given by equation 13.

$$PDP = P_{consumption} * \frac{1}{N * f} \tag{13}$$

It is though important to note that connecting logic gates in this manner, not always gives to worst-case transitions, which means that the results may be to optimistic sometimes.

3.4 FDSOI technology & body-bias

FDSOI is an abbreviation for *Fully Depleted Silicon On Insulator*, which means that the depletion region is fully depleted and insulated from the substrate with a thin oxide layer. There are also a similar technology called *Partially Depleted Silicon On Insulator* with the same oxide insulation layer, though here the body region of the transistor will only be partially depleted. A cross section of the regular transistors and FDSOI transistors can be seen in figure 5.

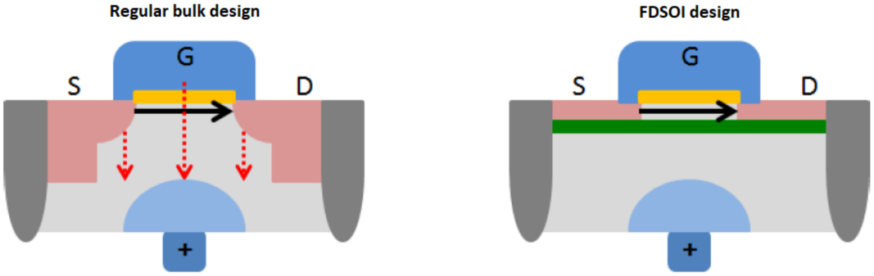


Figure 5: Regular bulk design for CMOS transistors are illustrated on the left, while FDSOI CMOS design is illustrated on the right. The difference lies in the thin oxide layer (green). Illustration from [26].

The purpose of introducing the oxide layer, is to lower the parasitic capacitances between the source and drain nodes of the transistor. In addition, the insulating oxide layer also drastically reduces leakage currents from source and drain to the bulk node, when applying body-bias [13]. The highly reduced leakage currents to the bulk node enables the opportunity to apply high voltage potentials for body-bias, without significantly increasing power consumption.

The transistors applied in this project, referred to as ***mosfet_high*** and ***mosfet_low***, contain the property of FDSOI technology, with the flipped-wells feature. Flipped-wells means that the n- and p-well area beneath the insulating oxide layer is switched between each other, i.e. the NMOS transistor will have a n-well area and the PMOS will have a p-well area beneath the depletion region, as seen in figure 6 [26].

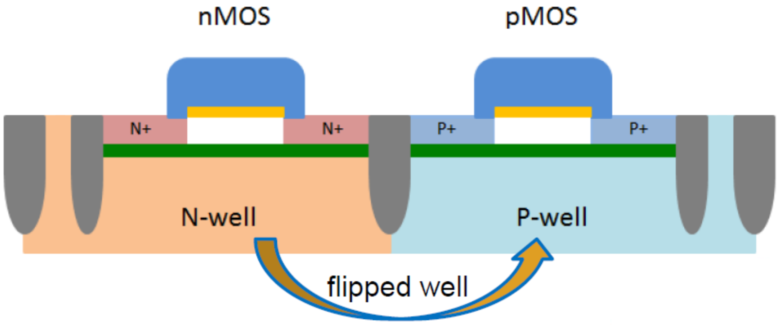


Figure 6: Illustration shows the property of flipped wells, where the n- and p-well below the oxide layer is swapped for the NMOS- and PMOS transistors. Illustration from [26].

As subthreshold circuits operate with more exponential properties compared to regular CMOS transistors operating in the active region, they may attain higher asymmetries in layout design. Therefore, FDSOI technology is well suited for subthreshold circuits, as threshold voltages of transistors in logic gates may be manipulated to a higher degree with body-bias, without worrying about higher power consumption. This technology has earlier proved to be useful for subthreshold circuits [27]. Applying a positive voltage on the body of a NMOS transistor, and negative to a PMOS transistor will result in Forward Body Biasing (FBB). Forward Body Biasing reduces the transistors' threshold voltage and allow faster switching capabilities. Reverse Body Bias (RBB) on the other hand is applying the opposite voltages of FBB, and thus increases the threshold voltage. Higher threshold voltages will further reduce leakage currents for inactive transistors.

3.5 SPICE simulation

SPICE simulation is short for *Simulation Program with Integrated Circuit Emphasis* and is an essential tool for this project. It is a general term used for software tools simulating analog circuit behaviour in integrated circuits. The circuits analyzed may be described in either schematics or netlists, and further developed to a graphical layout design. While schematics and netlists describe circuit behaviour under absolutely perfect theoretical conditions, layout design can give a more realistic behaviour in the real world. From layout design, one can extract parasitic resistance and capacitance which will occur due to wiring properties between component connections.

For this project, *Cadence Virtuoso* will be used for circuit simulations, and *Mentor Graphics Calibre* will be used to extract parasitic capacitances from layout design.

3.6 Basic logic gates

This project includes a cell library of several basic logic gates, stored in a cell library with schematics and layout design. The designed logic gates included are the most basic and general, and should be sufficient to build most of the more complex systems.

All the gates have different properties and functions, and will be explained in the following sections.

3.6.1 Inverter

The inverter is the most basic and fundamental logic gate, found in just about every digital circuit. It is simple in function, and as its name imply, it inverts the input signal. An input of ”low” will change the output to ”high”, and opposite for an input of ”high”. The function can be described mathematically by Boolean algebra as seen in table 1.

Table 1: Truth table for an inverter gate, where a denotes the input while z denotes the output value.

Inverter	
a	**z**
0	1
1	0

The symbol used to simplify the view of an inverter is shown to the right in figure 7, while the schematic description is shown to the left. For the schematic, the usual design consists of a NMOS- and PMOS transistor, though several transistors can also be connected in series, also called *stacked transistors* [28] [10] [29]. Transistors may also be stacked in parallel to increase speed in subthreshold circuits [30]. These methods will not alter the function, but may enhance the performance especially in subthreshold circuits with respect to leakage currents for stacked transistors and speed for parallel stacks. Both methods will though come at the cost of increased dynamic power consumption, and in addition increased leakage for parallel stacking.

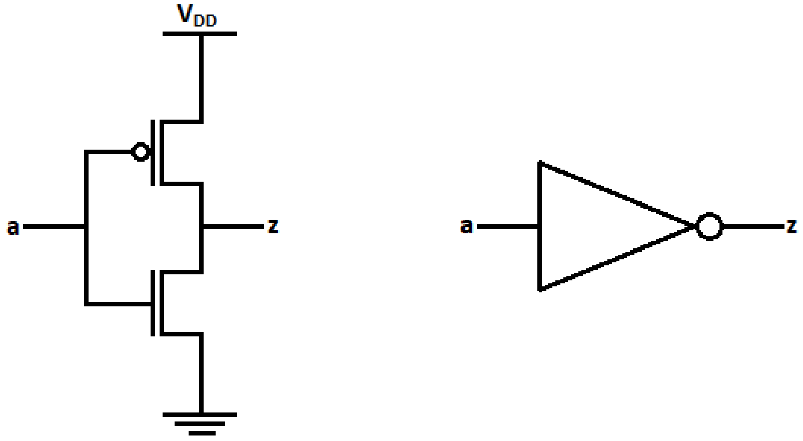


Figure 7: Schematic design of an inverter to the left, and the symbol used to simplify the view is shown to the right.

3.6.2 NAND

The NAND-gate is a two-input logic gate, which returns a single value. It is a widely used gate, and are used in addition with other gates to form most logic devices. The function is expressed in table 2.

Table 2: Truth table for a NAND-gate, where *a* and *b* denotes the input while *z* denotes the output value.

NAND		
a	**b**	**z**
0	0	1
0	1	1
1	0	1
1	1	0

The schematic design is shown in figure 8, with the symbol design illustrated on the right side. It is built in the same manner as the inverter, i.e. the traditional CMOS fashion [11] where NMOS- and PMOS transistors complement each other.

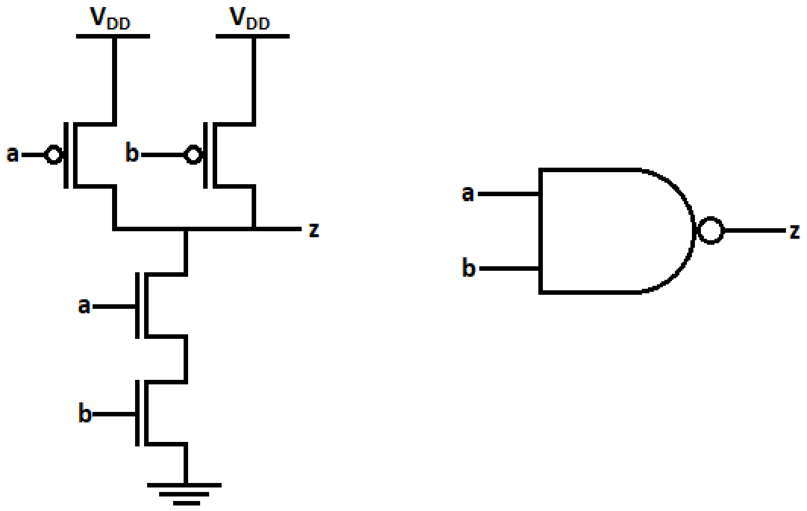


Figure 8: Schematic design of a NAND-gate to the left, and the symbol used to simplify the view is shown to the right.

3.6.3 NOR

The NOR-gate is, together with the NAND-gate, the most fundamental and most used gates in most digital circuits. These two gates, together with the inverter, may build up just about any complex digital circuit. The Boolean function for the NOR-gate is shown in figure 3.

Table 3: Truth table for a NOR-gate, where *a* and *b* denotes the input while *z* denotes the output value.

NOR		
a	**b**	**z**
0	0	0
0	1	0
1	0	0
1	1	1

Regarding the schematic design, the NOR-gate has the opposite symmetry of the transistor design compared to the NAND-gate. The design is shown together with the logic symbol in figure 9.

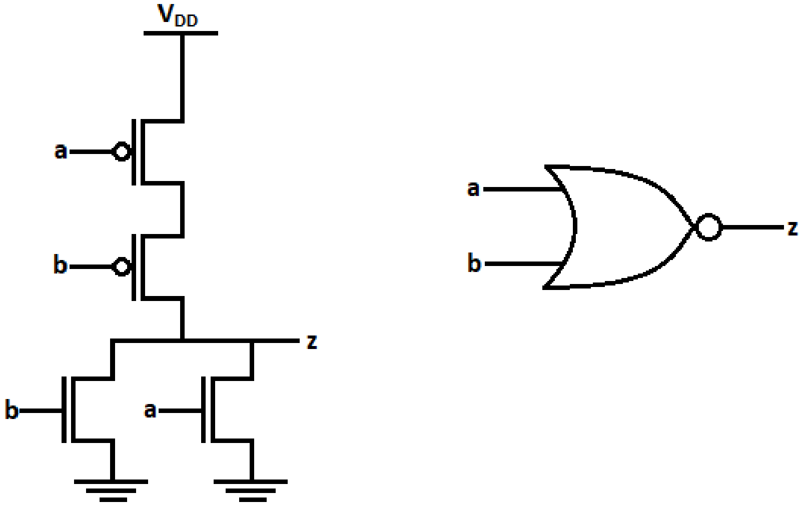


Figure 9: Schematic design of a NOR-gate to the left, and the symbol used to simplify the view is shown to the right.

3.6.4 XNOR

The XNOR-gate, also referred to as *Exclusive NOR-gate*, outputs a digital *high* value whenever the inputs signals are equal, that is when either both inputs are *low* or *high* at the same time. The function is shown in table 4.

Table 4: Truth table for a XNOR-gate, where a and b denotes the input while z denotes the output value.

XNOR		
a	**b**	**z**
0	0	1
0	1	0
1	0	0
1	1	1

The schematic layout design for the XNOR-gate is shown in figure 10, showing a total of 8 transistors. It is though important to note that some inputs are marked as an inverted input signal, that is $\overline{a}$ and $\overline{b}$. Therefore, 2 additional inverters are required for a single XNOR-gate, adding 4 more transistors to the design. However, if the same signal is to be applied for several XNOR-gates, they could share any common inverted signals, and thus reduce the required amount of tranistors.

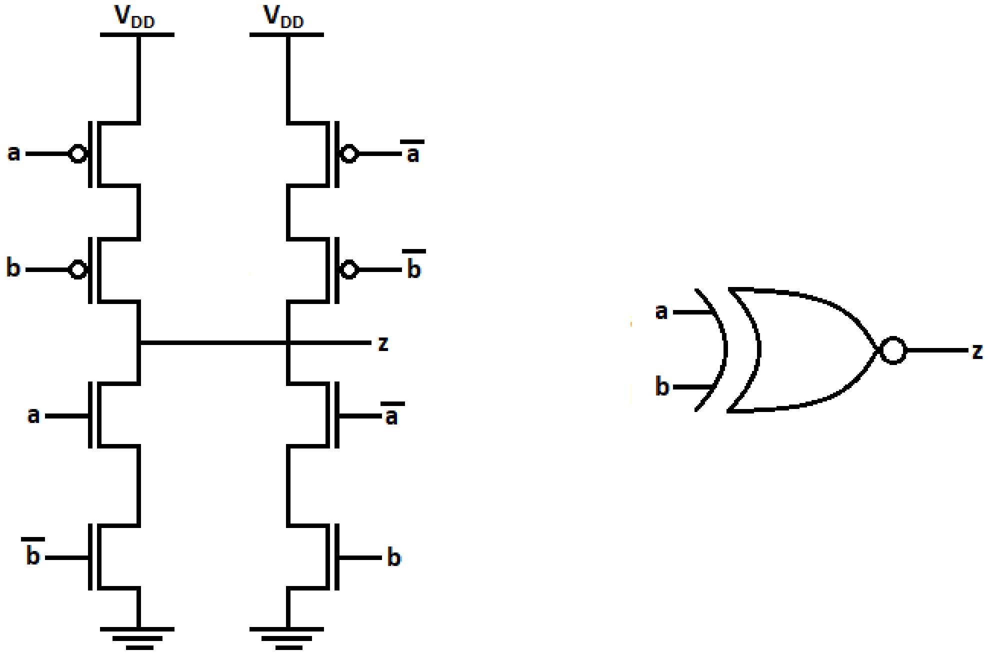


Figure 10: Schematic design of a XNOR-gate to the left, and the symbol used to simplify the view is shown to the right.

3.6.5 XOR

The XOR-gate, also called *exclusive OR-gate*, only supplies a positive output whenever the two inputs are of different logic values. The function can be seen in table 5, and the schematics in figure 11.

Table 5: Truth table for a XOR-gate, where a and b denotes the input while z denotes the output value.

XNOR		
a	**b**	**z**
0	0	0
0	1	1
1	0	1
1	1	0

As with the XNOR-gate, inverters are required to invert some of the input signals.

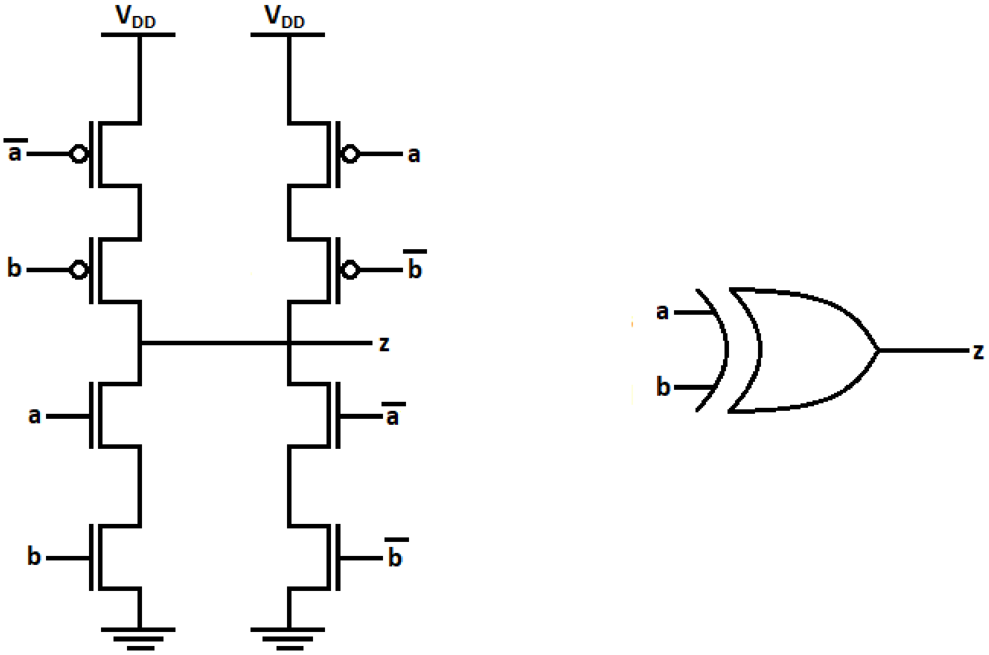


Figure 11: Schematic design of a XOR-gate to the left, and the symbol used to simplify the view is shown to the right.

3.6.6 Minority-3

The minority-3 gate is a logic device with three input terminals and a single output. The function is given in table 6, where it returns the minority of the input values, hence its name. The Minority-3 gate is a commonly used gate, and the 10-transistor design given in this paragraph has shown to be particularly useful and robust operating in the subthreshold region considering speed, power consumption and layout area [31] [32] [33] [34].

Table 6: Truth table for a Minority-3 gate, where a, b and c denotes the input while z denotes the output value.

Minority-3			
a	**b**	**c**	**z**
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

The design topology is shown in figure 12 with the logic symbol on the right side.

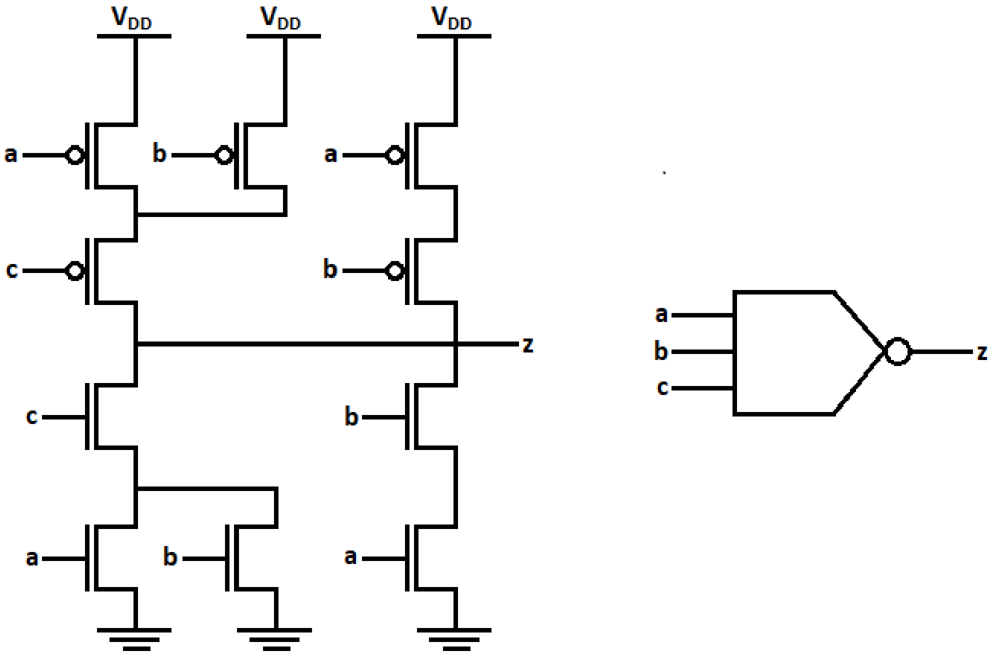


Figure 12: Schematic design of a Minority-3 gate to the left, and the symbol used to simplify the view is shown to the right.

3.6.7 Half-Adder

The Half-Adder is an essential part of any digital circuit computing numbers. It has two inputs, which usually represents two bits which will be added together in the circuit. The device returns either a *sum* or *carry* signal, though never both at the same time as seen in the truth table 7.

Table 7: Truth table for a Half-Adder gate, where *a* and *b* denotes the input while *sum* and *carry* denotes the output values.

Half-Adder			
a	**b**	**sum**	**carry**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

There are several ways to design a Half-Adder gate, where a design of XOR- and AND-gates has proven to be a reasonable choice for subthreshold circuits with respect to power, delay and noise margin [35]. The Half-Adder design in this project was though composed of XNOR- and NAND-gates, together with inverters. This was done as no XOR- or AND-gates were made for this library initially. Since Half-Adders will rarely be used in this project, a compromise is made and can be seen in figure 13.

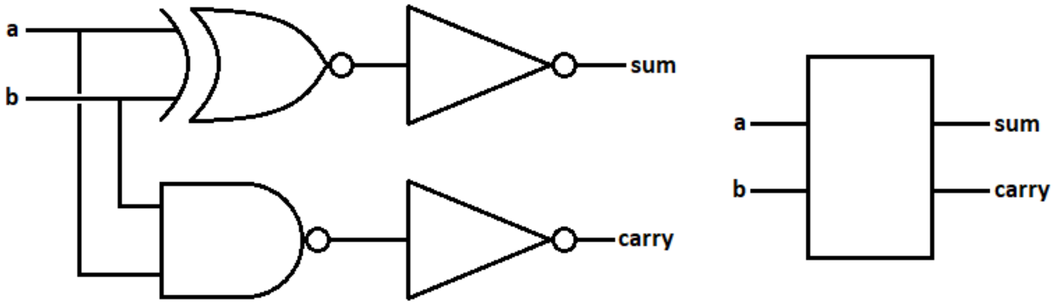


Figure 13: Schematic design of a Half-Adder gate on the left, composed of XNOR- and NAND-gates, together with two inverters. The symbol used to simplify the view is shown on the right side.

3.6.8 Full-Adder

The Full-Adder performs addition of two input bits, but unlike the Half-Adder, it can receive an additional carry-signal (denoted *c* in this report) from an adder circuit wired in series. This opens up for connecting several Full-Adders together and enable addition of several bits, i.e. computation of larger numbers.

Table 8: Truth table for a Full-Adder gate, where *a*, *b* and *c* denotes the input while *sum* and *carry* denotes the output values.

Full-Adder				
a	**b**	**c**	**sum**	**carry**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

There are several ways to design a Full-Adder in CMOS, and a few will be presented and investigated in the following sections. All of them will function according to the same truth table, but performance properties and layout design will vary. The standard Full-Adder and Minority-3 based Full-Adder will later be referred to as simply ***Full-Adder (Std.)*** and ***Full-Adder (Min-3)*** respectively. All Full-Adders will however have the same symbol, as shown in figure 14.

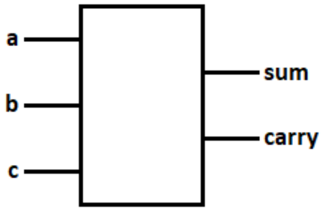


Figure 14: Full-Adder symbol for simplified schematics.

3.6.9 Standard Full-Adder

The standard Full-Adder is composed of only 28 transistors allowing compact design, and is the most common design for regular Full-Adder CMOS circuits. This design is considered a steady Full-Adder design allowing good matching of fall- and rise times, and has also shown good promises for ultra-low voltage operation [36].

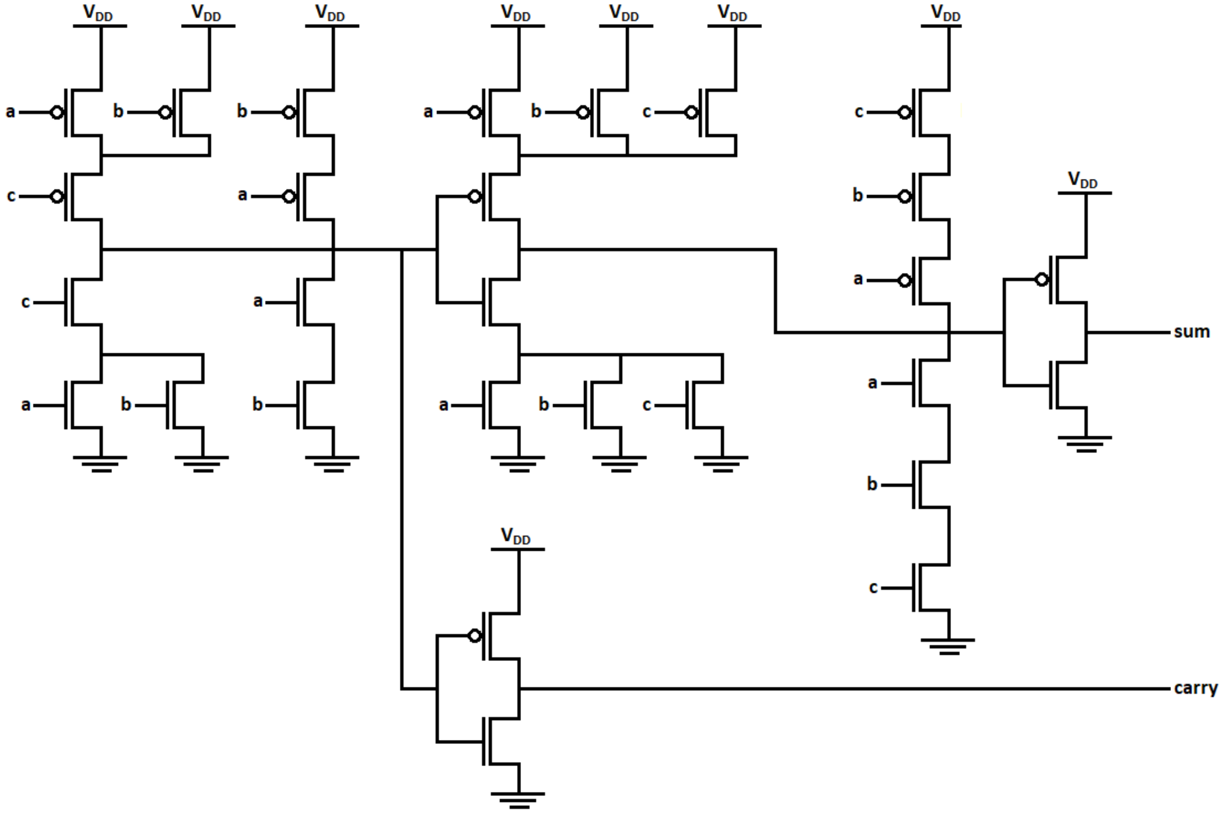


Figure 15: Schematic design of a standard Full-Adder gate on the top. The symbol used to simplify the view is shown in figure 14.

3.6.10 Minority-3 based Full-Adder

A Full-Adder composed by minority-3 gates and inverters seems to be a good compromise between delay and power consumption when operating in the subthreshold region according to [36] [37]. This is despite the relatively higher count of 34 transistors, or 3 Minority-3 gates and 2 inverters, which can be seen in figure 16. It can also be seen in table 19 that inverters and Minority-3 gates of ***mosfet_low*** transistors have among the lowest static power consumption of the logic gates used in this project. According to [38], a Full-Adder designed by these gates also seems to be the most promising composition for subthreshold operation with respect to Power Delay Product.

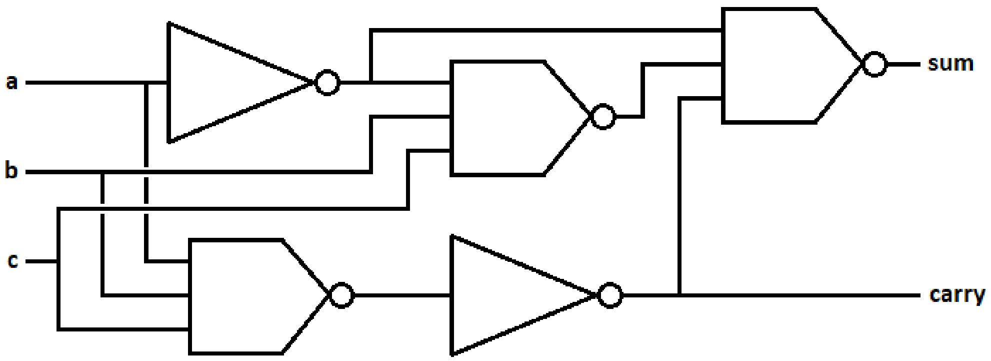


Figure 16: Schematic design of a Full-Adder gate on the top, composed of Minority-3 and inverter gates. The symbol used to simplify the view is shown in figure 14.

3.6.11 NAND & XOR based Full-Adder

A Full-Adder composed of NAND- and XOR-gates were investigated in [39], and seems useful with its low fan-in and transistor count, i.e. 32 transistors.

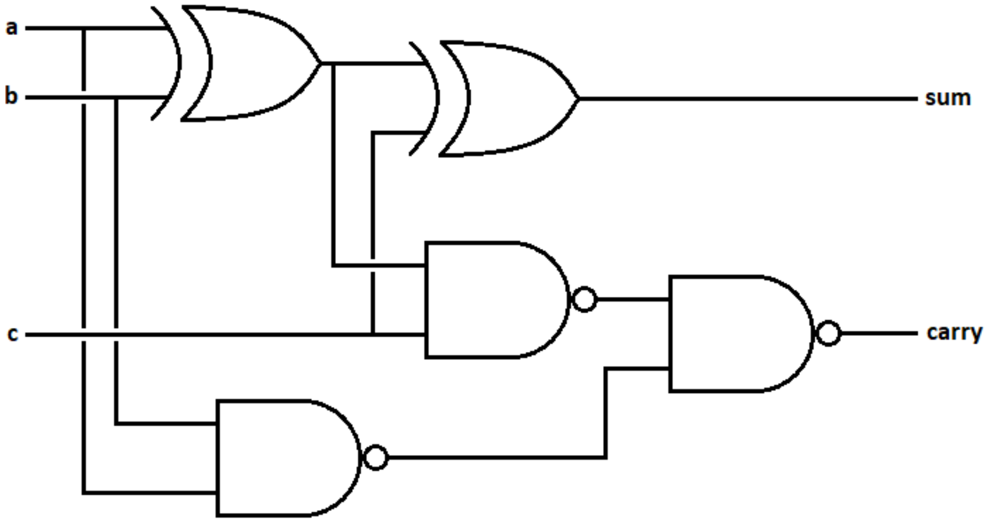


Figure 17: Schematic design of a Full-Adder gate on the top, composed of NAND- and XOR gates. The symbol used to simplify the view is shown in figure 14.

3.6.12 NAND based Full-Adder

A Full-Adder may be designed by only NAND-gates, and would then require a total of 9 NAND ports as shown in figure 18. This design will however require a total of 36 transistors.

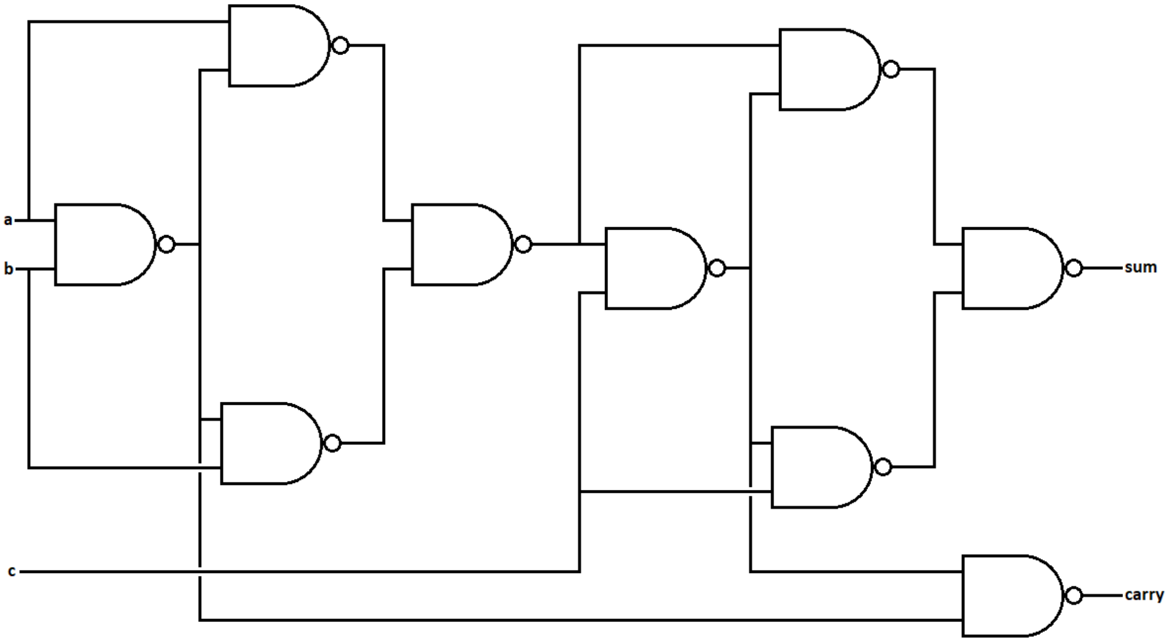


Figure 18: Schematic design of a NAND based Full-Adder gate on the top. The symbol used to simplify the view is shown in figure 14.

3.6.13 NOR based Full-Adder

As with the NAND based Full-Adder, it may also be designed by only using NOR-gates. This design will however require 12 NOR-ports, or 48 transistors in total.

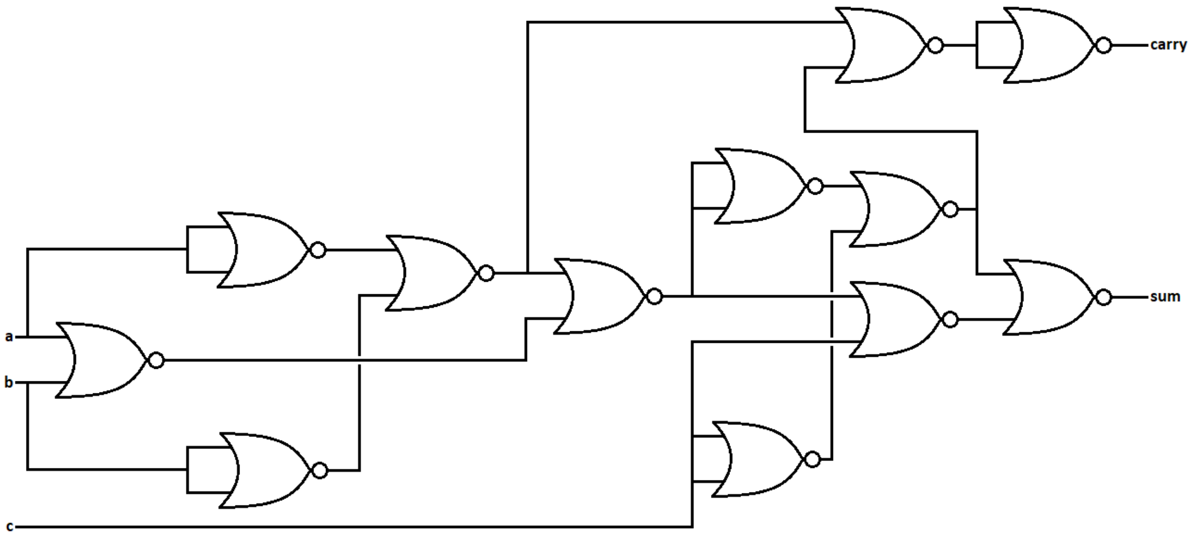


Figure 19: Schematic design of a NOR based Full-Adder gate on the top. The symbol used to simplify the view is shown in figure 14.

3.6.14 Ripple-Carry Adder

A Ripple-Carry adder is an adder capable of adding greater numbers of binary digits. The adder is built up by a Half-Adder followed by several Full-Adders in series. The amount of Full-Adders is relative to the amount of binary numbers which are to be added, though there will always be only one Half-Adder, placed at the *lowest significant bits*. Of course, it is not strictly required that the first adder is a Half-adder, but in general a Half-Adder will consume less power and layout area compared to a Full-Adder. The structure of a Ripple-Carry adder can be seen in figure 20.

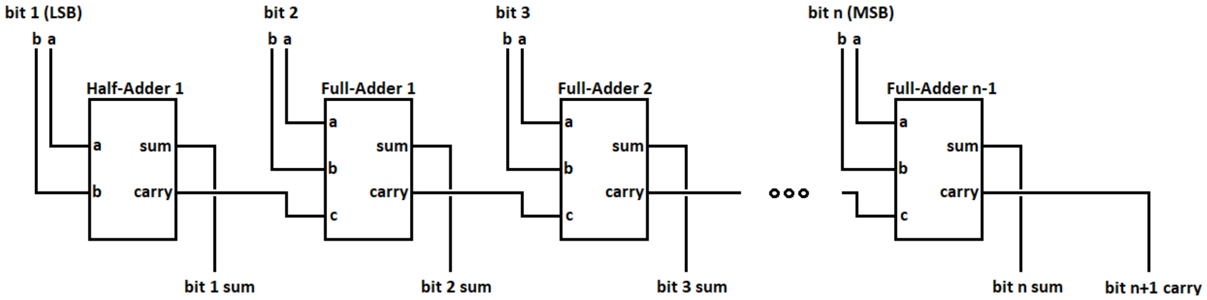


Figure 20: Schematic overview of a n-bit Ripple-Carry adder, composed of a Half-Adder in the first block followed by several Full-Adders in series. port a and b receives two different bit values of similar significance, which are to be added. c is the $carry_{in}$ signal from the previous $carry_{out}$ signal. The complete added bit sequence is given by the sum output ports.

As one can see from the figure, there is a Half-Adder in the first block receiving the *least significant bits* (LSB), from two different binary numbers, respectively bit-a and bit-b. The first block is a Half-Adder as there is simply no need for a Full-Adder to receive a *Carry*-signal which never will occur in the first block. This is because there is no *carry* signal to receive, and therefore one can reduce the amount of transistors by using a Half-Adder instead.

The output binary value is given by the *sum* outputs, which together give a number of bits. In addition, there will be a *carry*-bit from the last Full-Adder of an order higher than the *most significant bit* (MSB) received. This bit will occur if the two added binary values will increase the binary sum with one bit.

The Ripple-Carry adder may not be ideal with respect to speed, considering that the worst-case critical path might have to go through all the Half- and Full-Adders implemented in series. it is though suggested that serial adders composed of minority-3 gates and inverters may be more ideal for subthreshold operation, compared to for example the parallel processing Kogge-Stone adder [40].

3.6.15 D Flip-Flop

The D Flip-Flop is an edge triggered logic gate, which operates with a continuous clock signal, *clk*, on one of the input ports. The clock speed is usually constant, while the other input, a, varies. The output value is changed at either the rising- or falling edge of the clock signal, and will remain unchanged until the next rising- or falling edge of the clock signal, according to the input value. Therefore, the D Flip-Flop stands out from the earlier described logic gates as this gate has a memory property. The functionality is shown in table 9. The D Flip-Flop also has a $\overline{Q}$ output, which is always the opposite of the Q output.

Table 9: Truth table for a negative-edge triggered D Flip-Flop. When input a is noted as "x", it means that any input will not have any effect on the output. ↑ or ↓ is respectively the positive or negative edge of the *clk* signal.

D Flip-Flop		
clk	**a**	**Q(t+1)**
0	x	Q(t)
1	x	Q(t)
↑	x	Q(t)
↓	0	0
↓	1	1

There are almost countless design variations of the D Flip-Flop, where they all have their strengths and weaknesses. Some designs excels with respect to pure performance, others are more robust to environment variations like temperature and radiation, and some are optimized for a low power consumption [19].

The D Flip-Flop applied in this report is a negative-edge triggered gate [31] composed of an inverter and 6 NOR-gates, i.e. it changes its output according to the input value, when the clock signal is falling from

”high” to ”low”, or shown as ↓ in table 9. The choice of design was due to the promising results it had with respect to Power Delay Product and static power consumption. The schematic can be seen in figure 21.

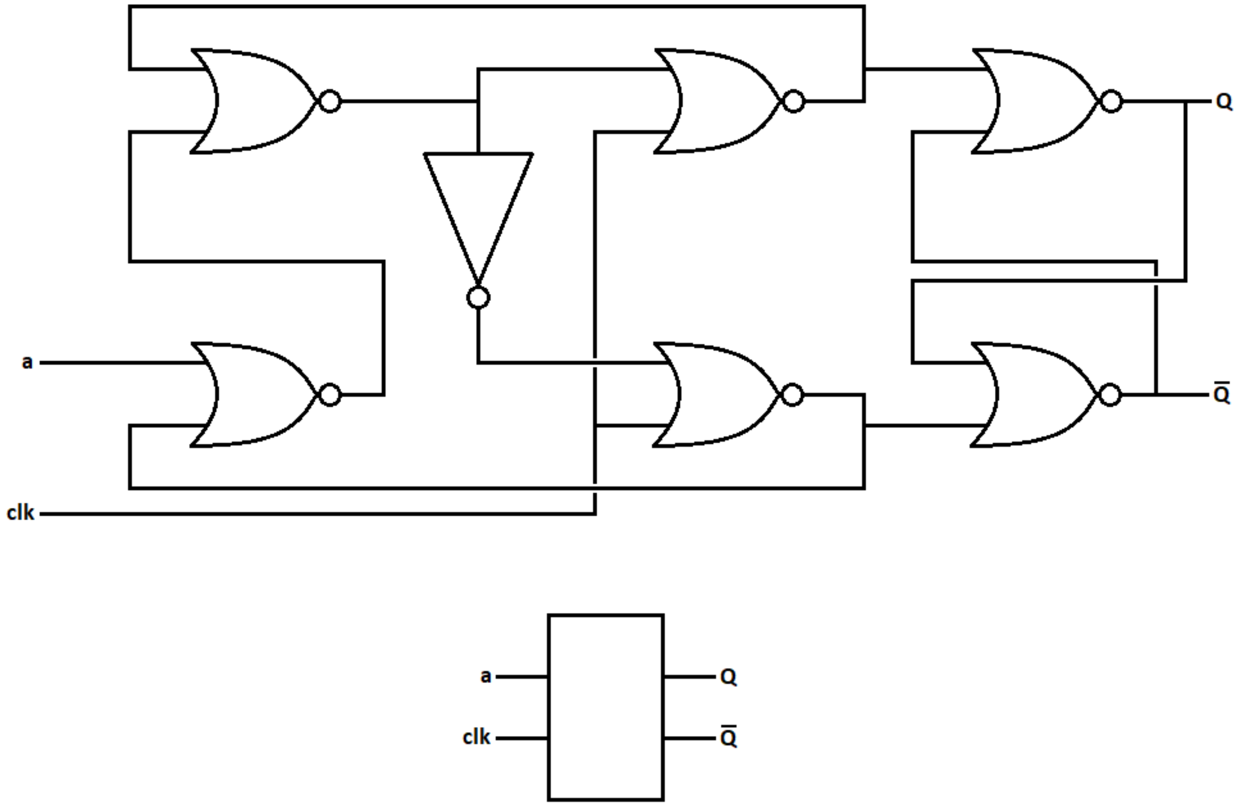


Figure 21: Schematic design of a D Flip-Flop gate on the top, composed of NOR- and inverter gates. The symbol used to simplify the view is shown on the bottom.

4 Methods

4.1 Transistor properties

Several types of transistors were investigated in [41] for a commercial available 22 nm CMOS FDSOI technology. The examined transistors are characterized by different threshold voltages. Two transistors showed promising results, and will be further researched in this project for the same CMOS technology. They will throughout the report be referred to as ***mosfet_high*** and ***mosfet_low***, where ***mosfet_high*** has the highest threshold voltage and the other one has the lowest threshold voltage. There will also be designed a library by ***mosfet_low_bb*** transistors, which means that the library is composed of ***mosfet_low*** transistors with body-bias enabled. Devices denoted as composed by ***mosfet_low*** transistors do not have body-bias enabled, i.e. all bulk nodes are connected to ground.

The test bench applied for extracting the transistor properties can be seen in figure 22. Note that the bulk node for both the NMOS- and PMOS transistors are connected to ground. This is due to the property of flipped-wells as mentioned earlier [26].

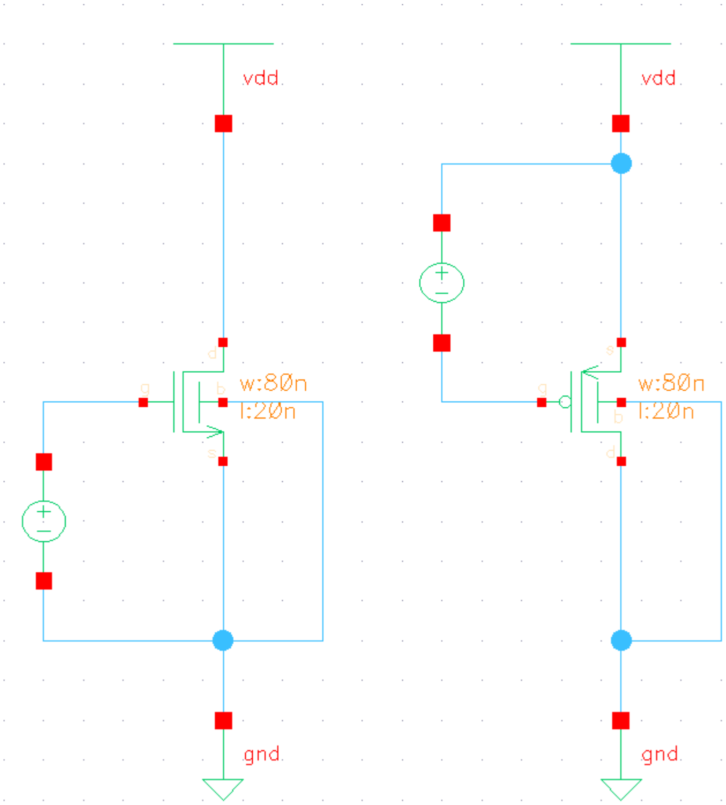


Figure 22: The test bench applied in ***Cadence Virtuoso*** for analyzing the NMOS- and PMOS transistor with 0 V body-bias.

4.1.1 Threshold voltage & body-biasing

As this report investigates subthreshold circuitry, it is essential to extract the threshold voltages, $\mathrm{V_{th}}$ of the two transistors examined. The threshold voltage was earlier explained as the boundary voltage for when MOSFETs starts to conduct current from drain to source. But there are many different opinions and definitions of where exactly the threshold lies. The method applied in this report was the extrapolation method, which is discussed among several other definitions in [42].

Body-bias for the CMOS technology applied in this circuit, is reported to be enabled from the range of -2 V to +2 V for the flipped-well devices [26]. The NMOS may be biased from 0 to +2 V, and the PMOS is enabled from 0 to -2 V on the bulk node, i.e. only forward body-biasing is enabled.

4.1.2 Transconductance

The transconductance, g_m, is an useful parameter to extract when analyzing MOSFET transistors. g_m gives an indication of how strong the transistor, i.e. how well it translates the gate voltage into current flow through the drain- and source nodes. Equation 14 defines the transconductance of a MOSFET transistor [21].

$$g_m = \frac{\partial I_{DS}}{\partial V_{GS}} \tag{14}$$

I_{DS} is the drain-to-source current, and V_{GS} is the voltage applied at the gate node, relative to the source node voltage. The property may be extracted as a parameter in ***Cadence Virtuoso***, and its unit is given by A/V.

4.1.3 Transresistance

Transresistance, g_{ds}, is the inverse of the resistance between the drain and source node of an MOSFET. It is given by the relation of the voltage and current between the two nodes, and is given by equation 15.

$$g_{ds} = \frac{1}{r_{ds}} = \frac{\partial I_{DS}}{\partial V_{DS}} \tag{15}$$

I_{DS} is the drain-to-source current, and V_{DS} is the voltage between the drain and source nodes. As with transconductance, the property may also here be extracted as a parameter in ***Cadence Virtuoso***. Transresistance unit is given in Ω^{-1}.

4.2 Worst-case scenario input

A logic gate will receive several different combinations of input signals throughout operation, where the different combinations will affect the performance of the logic device. Therefore worst-case scenarios are important to map for the individual devices, as best-case scenarios will give too optimistic results. If worst-case input is not considered, it may result in designing devices which will only function in certain cases. To optimize a circuit for a given supply voltage, it is important to do so under the worst-case conditions to ensure reliable operation.

For the simpler logic gates (i.e. the inverter, NOR, NAND, XOR and XNOR) with relative few in- and outputs, the worst-case scenarios are defined for both delay and switching energy. Worst-case transitions are not only dependant of the different possible input combinations, but also affected by the earlier state, i.e. what value the output has at the time it receives a new combination of input values. To determine the amount of different possible input- and output combinations for a logic gate, on can use the equation 16, where n is the amount of possible cases, and N_{port} is the total amount of both input- and output ports on the device.

$$n = \prod_{i=0}^{N_{port}-1} 2^i \tag{16}$$

For the inverter, the worst-case is relative simple to map as there are only four different states to compare ($0 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 0$ and $1 \rightarrow 1$). 2-input gates like NAND, NOR, XOR and XNOR have on the other hand $2^0 * 2^1 * 2^2 = 8$ different combinations. The Minority-3 gate and the Full-Adder with their 4 ports give a total of $2^0 * 2^1 * 2^2 * 2^3 = 64$ different combinations. One can see that the more ports a logic gate has, the amount of different possible situations increase exponentially. Therefore, for the more complex gates like the Minority-3, Half-Adder and the Full-Adder, this gives too many different cases to investigate if it is to be performed manually. The Full-Adder has for example a total of 64 different input combinations [43].

The worst-case scenarios for the simpler logic gates will be extracted by simulating every possible input combination for both delay and switching energy consumption, and then choosing the combination with the highest delay and most energy consumption as the worst-cases for the two properties. Worst-case input may though be given on basis of either delay or switching energy, but for this project, delay was chosen as this is most relevant with regards to requirement specifications. If switching energy and operating frequency (or delay) is known, one may calculate dynamic power consumption for a device under any activity factor, as given in equation 2.

There are also worst-case input combinations for static power consumption, though the number of input combinations will be far less as transitions do not need to be considered. However, the D Flip-Flop is an exception to this statement, due to the memory property mentioned before. For all the other gates, the number of combinations are only given by the amount of input ports, as given by equation 17.

$$n = 2^i \tag{17}$$

n is the number of different scenarios and i is the total number of input ports on the logic gate.

However, no simple method was found to determine the worst-case transition for the D flip-flop, due to the memory property. The worst-case was therefore found manually, by testing every input combination separately. Delay was chosen as a basis for the worst-case input, as speed is the main concern in this report.

As mentioned, the D Flip-Flop is a bit different from the other tested devices due to the memory effect and the repeating clock cycle. Therefore, the switching energy will be dependant of the logic value "stored" in the gate, in addition to the two inputs; *a* and *clk*, and at what times they appear. Input values may vary at nye time while output values remain constant, until the *clk* signal goes low, as it is negative-edge triggered. This gives an infinite amount of different input scenarios, so for this project, only switching while *clk* goes high or low will be considered.

4.3 Test bench

To extract the worst-case input combinations, along with static power, delay and switching energy, a proper test bench setup must be applied to the device under test (DUT), which gives realistic surroundings. The test bench used in this project can be seen in figure 23.

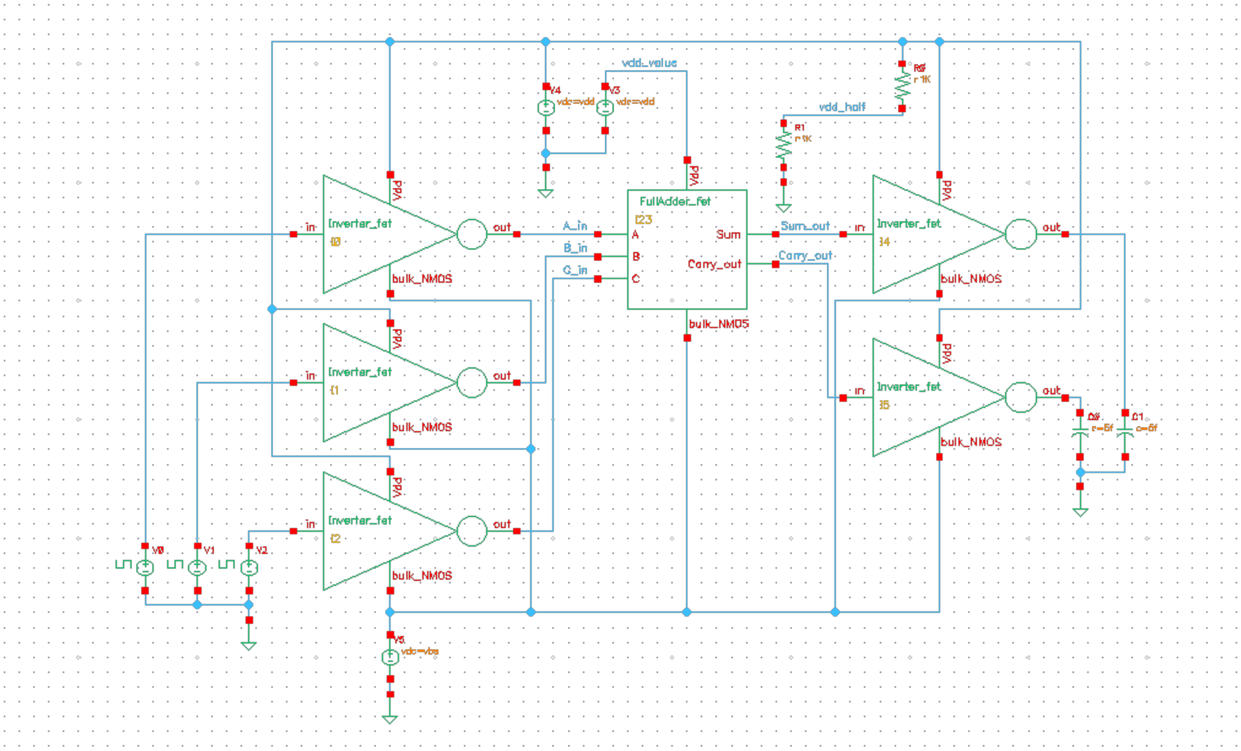


Figure 23: Simple overview of the test bench setup used for different simulations. This setup was specific for simulating Full-Adders.

The device in the test bench is controlled by an input voltage source, generating pulses according to bit sequences, relative to what property to be analyzed. Between the signal source and the device, an inverter is inserted in order to transform the input signal into more realistic waveforms. An inverter acting as a load, is attached to the output of the device with a 5 fF load on its output. Devices with several inputs and outputs, will be attached to an equivalent number of inverters with the same setup. An isolated voltage source is set up to supply the device, in order to measure the exact current the DUT draws.

4.4 Deriving a supply voltage

A speed of minimum 40 MHz was set as a minimum requirement in the introduction, which equals to a maximum delay of 25 ns. The most complex logic gate for the ultrasound application is given to be a 16by12-bit adder. A 16by12-bit adder is an adder capable of adding 32 different 12-bit input values and give the result as a single 16-bit value at the output, with an extra carry signal. This is done by applying 12-, 13-, 14-, 15- and 16-bit Ripple-Carry adders, wired together as seen in figure 24.

To determine a suitable supply voltage for the given speed requirement, a theoretical approach should initially be considered to derive a minimum supply voltage. For this report the adders will be designed as Ripple-Carry adders, i.e. by blocks of an initial Half-Adder followed by several Full-Adders in series.

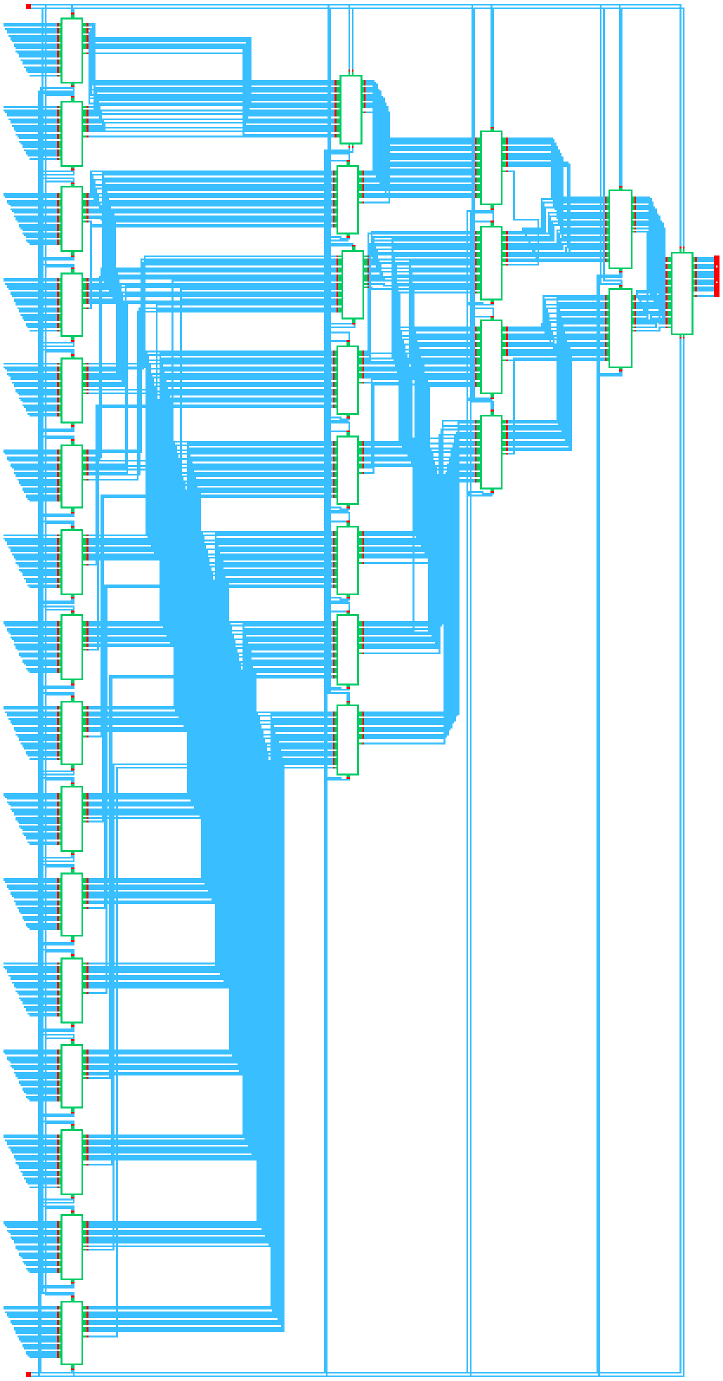


Figure 24: Schematic overview of the 16by12-bit adder. 32 different 12-bit digital values can be given on the input on the left side, into the 12-bit adders. After being added into 13-bit values, they are further sent to the 13-bit adders next in the chain of adders, and so on.

As mentioned in the theory section, the critical path through a Ripple-Carry adder might have to go through all the Half- and Full-Adders from the beginning to the end. The critical path is shown in figure 25.

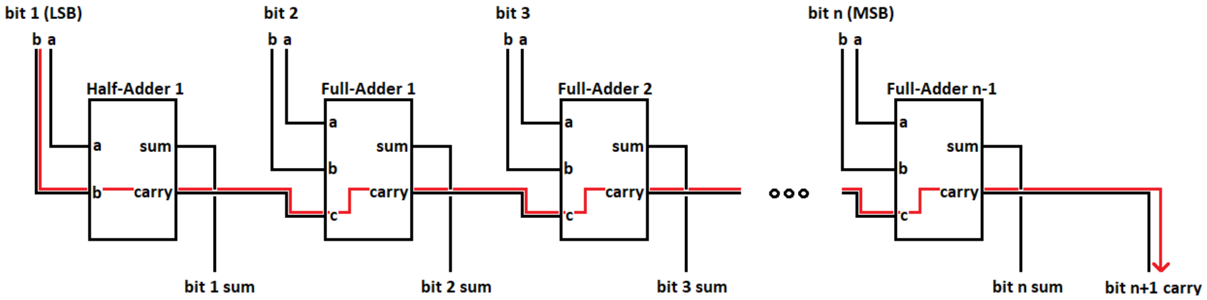


Figure 25: Schematic overview of a n-bit Ripple-Carry adder, illustrating the critical path (red). This worst-case scenario is given under the assumption where input a and b is both 1 for the Half-adder, but 0 and 1 for the Full-Adders.

By observing the critical path, one should be able to give an estimation of how much total delay a 16by12-bit adder will add for a worst-case scenario, if the delays for the different logic gates are known. Critical path is chosen here as simulating every possible transition is unrealistic for this logic gate, with its 2^{192} possible input combinations before even considering the previous state. As one can see from the critical-path figure, it goes via the *Carry* signal. By looking at the schematics for the Half- and Full-Adders in the theory section, one can see the logic gates which the signal passes through. The estimated gate delay for the the Half- and Full-Adders can be seen in equation 18 and 19, respectively.

$$Half\_Adder_{Carry\ signal\ delay} = delay_{NAND} + delay_{Inverter} \tag{18}$$

$$Full\_Adder_{Carry\ signal\ delay} = delay_{Minority-3} + delay_{Inverter} \tag{19}$$

These equations can be further inserted into equations 20 and 21 to calculate delay of more complex adders. The first equation, 20, gives the delay of a Ripple-Carry adder able to add n bits binary numbers. The second equation, 21 gives the delay of adders built up by smaller Ripple-Carry adders.

$$delay_{n\_bit\_adder} = Half\_Adder_{Carry\ signal\ delay} + (n-1) * Full\_Adder_{Carry\ signal\ delay} \tag{20}$$

$$delay_{m\_by\_n\_bit\_adder} = \sum_{i=n}^{m} delay_{n\_bit\_adder} \tag{21}$$

To estimate the delay for for a 16by12-bit adder given in this project, one can modify the equation 21, as seen in the following equation 22.

$$
\begin{aligned}
delay_{\boldsymbol{16\_by\_12\_bit\_adder}} &= \sum_{i=12}^{16} delay_{\boldsymbol{n\_bit\_adder}} \\
&= delay_{\boldsymbol{12\_bit\_adder}} + delay_{\boldsymbol{13\_bit\_adder}} \\
&\quad + delay_{\boldsymbol{14\_bit\_adder}} + delay_{\boldsymbol{15\_bit\_adder}} + delay_{\boldsymbol{16\_bit\_adder}} \\
&= 70 * delay_{Inverter} + 5 * delay_{NAND} + 65 * delay_{Minority-3}
\end{aligned}
\tag{22}
$$

From equation 22 the number of gate delays to be multiplied with the respective delays can be seen. It is then necessary to know the worst-case delays for the three gates included in the critical path, and has earlier been analyzed in [41], and can be seen in table 10. It should be noted that this will result in a rough estimate, as it may not be the carry-signal (Which is a part of the critical path) which yields the worst-case delay given in the table.

Table 10: Time delays for different logic gates by different transistor types, at different supply voltages.

		Worst-case delay		
Logic gate	**Transistor type**	**V_{DD}=300 mV**	**V_{DD}=350 mV**	**V_{DD}=400 mV**
Inverter	mosfet_high	3.5 ns	920 ps	286.4 ps
	mosfet_low	24.2 ps	14.2 ps	9.0 ps
NAND	mosfet_high	5.4 ns	1.4 ns	435.6 ps
	mosfet_low	53.2 ps	31.5 ps	20.2 ps
Minority-3	mosfet_high	10.4 ns	2.8 ns	896.8 ps
	mosfet_low	77.9 ps	44.5 ps	27.6 ps

From table 10, it seems that the combination of ***mosfet_low*** transistors at V_{DD}=350 mV seems to give a total delay of 4.04 ns (248 MHz) according to equation 22. This comply with the requirement specifications, with a safety margin relation of 6.1 relative to the 25 ns requirement. A safety margin is important to maintain, as layout design and parasitic capacitances have not yet been considered.

As a supply voltage of V_{DD}=350 mV seems to be an appropriate voltage according to the calculations in this section, the ***mosfet_low*** transistor adder will be tested for delay in the results section, at this given supply voltage. The ***mosfet_high*** transistors will not be further analyzed, as its delay seems to large to justify further research of it.

4.5 Logic gates sizing

The logic gates in this report were initially optimized for 250 mV supply voltage. However, this supply voltage turned out to be insufficient for the requirement specifications given in this project. The logic gates in this table also had some significant imbalance in transistor dimensions, which could introduce unnecessary challenges in later layout design. For these reasons, sizing and body-biasing were updated and is shown in table 18, with much better balance in terms of NMOS- and PMOS widths. Two libraries were made for the ***mosfet_low*** devices, where one adopts body-bias, while the other does not.

Sizing of the logic gates were performed by short-circuiting the input nodes of the logic gate together, and sweeping the input voltage from 0 V to V_{DD} voltage. The width of the transistors were then balanced so that $V_{in}=V_{out}=V_{DD}/2$. Lengths were always kept at 20 nm for all circuits, as seen in table 18. The more complex gates, like the Half-, Full- and Ripple-Carry Adder were not further optimized after being assembled by the basic gates according to the given schematic. This should not be necessary as the basic gates already are optimized.

4.6 Circuit and layout design

For this project, *Cadence Virtuoso* will be used for both circuit and layout design. *Cadence Virtuoso*'s *ADE L* simulator will be applied for circuit simulations, both with and without parasitic capacitances included. For schematic design, the *Schematic L* tool will be used, and for the layout design the *Layout XL* tool will be used. *Cadence QRC* will be used to extract the parasitic capacitances from layout design, while parasitic resistances will not be considered.

For reliable design of the layout schematics, design will be tested through the tools *Design Rule Check* (DRC) and *Layer Versus Schematic* (LVS). These tools will respectively check if the design is possible to be manufactured reliable in real world conditions, and if the wiring corresponds to the schematics design initially. All layout designs will be optimized until no DRC- or LVS errors occur, unless otherwise is stated.

5 Results

5.1 Transistor properties

The following section will show and illustrate the results of the experiments methods discussed in the *Method* section.

5.1.1 Threshold voltage

The threshold voltage, V_{th}, was initially mapped for both the ***mosfet_high-*** and ***mosfet_low*** transistors, and can be seen in figure 26.

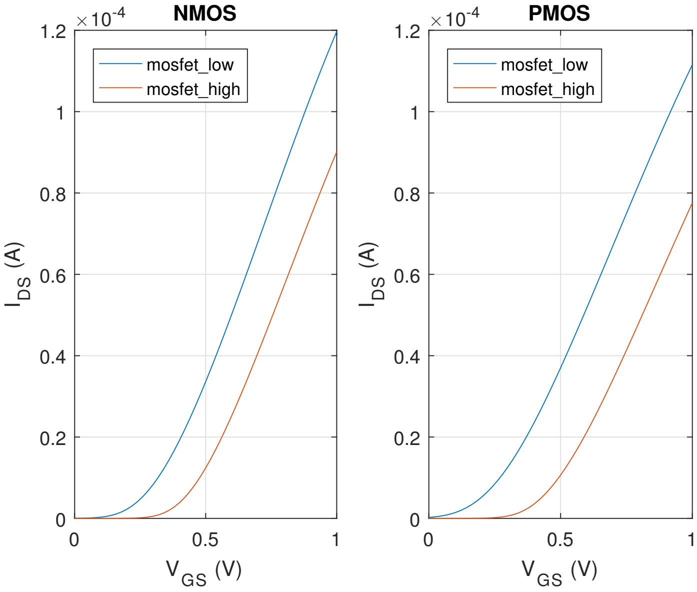


Figure 26: Threshold voltage, V_{th}, extraction by applying the extrapolation method. The current I_{DS} mapped as a function of V_{GS} is the current flowing from drain to source node, and the other way around for the PMOS. W_p=W_n=80 nm, L_p=L_n=20 nm, V_{DD}=1 V and V_{GS} is swept from 0 V to 1 V.

From the plot in figure 26, a tangential line may be drawn from the curve in the triode region and extract the threshold value from where the line intersect ≈ 0 A drawn from the power supply. The threshold values can be seen in table 11.

Table 11: Threshold voltages for the *mosfet_high* and *mosfet_low* transistors, at V_{DD}=1 V. No body-bias were applied.

	Threshold voltage [|mV|]	
Transistor	**NMOS**	**PMOS**
mosfet_high	456	461
mosfet_low	308	264

The threshold voltage will admittedly change when reducing the supply voltage down to 350 mV, but this should not change drastically, and the simulations were mainly done to indicate the threshold values.

5.1.2 Transconductance

The transconductance, g_m, was extracted at a supply voltage of V_{DD}=350 mV with no body-bias, as given in table 18. A supply voltage of 350 mV was earlier found to be a reasonable supply voltage for the requirements of this project, and therefore chosen for these simulations. The parameter was extracted from DC-sweep simulation of V_{GS} in ***Cadence Virtuoso***, and can be seen in figure 27.

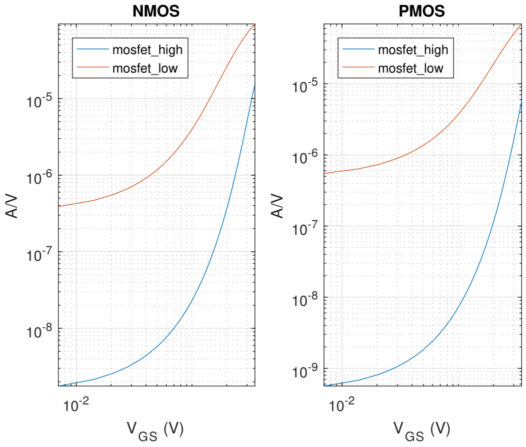


Figure 27: Transconductance given by DC-sweep of V_{GS} from 0 V to V_{GS}=V_{DD}=350 mV. No body-bias were applied. Transistor dimensions were given by W_p=W_n=80 nm, L_p=L_n=20 nm

5.1.3 Transresistance

The transresistance, g_{ds}, was extracted at a supply voltage of V_{DD}=350 mV, with no body-bias as given in table 18. As with the transconductance simulations, a supply voltage of 350 mV was earlier found to be a reasonable supply voltage for the requirements of this project, and therefore chosen for these simulations. The parameter was extracted from DC-sweep simulation of V_{GS} in ***Cadence Virtuoso***, and can be seen in figure 28.

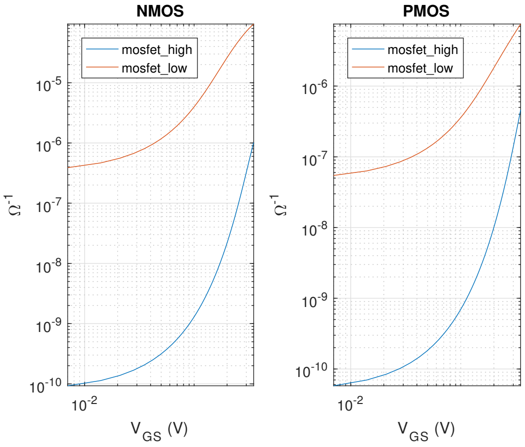


Figure 28: Transresistance given by DC-sweep of V_{GS} from 0 V to V_{GS}=V_{DD}=350 mV. No body-bias were applied. Transistor dimensions were given by W_p=W_n=80 nm, L_p=L_n=20 nm

5.2 Worst-case scenario input

Worst-case switching combination with respect to energy consumption was found for the simpler gates, and is given in table 12. The energy consumption was extracted by integrating the current drawn from the isolated voltage supply, as shown in figure 29. To avoid integrating the static currents before and after switching, boundaries were defined as where the current consumption exceeded ±10% of the average leakage current, for the two different input states respectively.

As no method of automating the process of mapping switching energy was found or developed, time delay was chosen as basis for the worst-case inputs for the more complex devices investigated in this report, both when extracting delay and switching energy. The simpler gates have different worst-case transitions for delay and switching energy, as seen in table 12. Since the project requirements only specify time requirements and no power consumption restraints, delay is the most important aspect to consider.

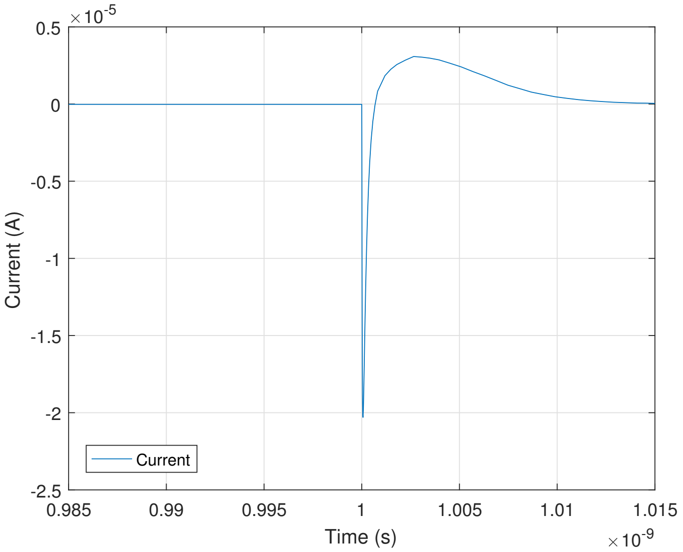


Figure 29: Typical transient analysis of the energy consumed while switching, for an inverter of type ***mosfet_low*** transistors at V_{DD}=350 mV. The switching here initiates at 1 ns, and stops at approximately 1.012 ns.

Some irregularities occurred in the results, as the waveforms of power consumption varied in shape for different transistor structures, sizes, biasing and supply voltages. Sometimes the current consumption would never reach 110% (or 90%) of the leakage current after switching, and therefore an approximation was made. The behaviour of the waveform proved to vary greatly throughout simulations of different logic gates, and waveforms sometimes oscillated several times over and below the final value before stabilizing within the ±10% limit. Some times the waveform went below 0 A, as seen in figure 29, supplying considerable amounts of current back into the supply source.

Table 12: An overview of the different states for worst-case input for different transistors and logic gates. The worst-case for delay are given by the transition from one state to another. A worst-case given as 10 → 00 means that port *a* is initially fed with a high input signal followed by low input signal. Port *b* has initially a low input signal, which also remains low after the switching. The energy or delay is extracted from the transition between the two input combinations.

		Worst-case transition (Port a, b, ...)	
Logic gate	**Transistor type**	**Energy**	**Delay**
Inverter	mosfet_high	1 → 0	1 → 0
	mosfet_low	1 → 0	1 → 0
	mosfet_low_bb	1 → 0	1 → 0
NAND	mosfet_high	11 → 00	00 → 11
	mosfet_low	11 → 00	00 → 11
	mosfet_low_bb	11 → 00	00 → 11
NOR	mosfet_high	11 → 00	11 → 00
	mosfet_low	11 → 00	00 → 10
	mosfet_low_bb	11 → 00	00 → 10
XOR	mosfet_low	11 → 10	00 → 10
XNOR	mosfet_high	00 → 10	01 → 11
	mosfet_low	00 → 10	10 → 11
	mosfet_low_bb	00 → 10	10 → 11

The DUT was inserted in a test bench, given in figure 23, where all the different combinations were fed into the DUT in a transient analysis. The delays were then measured in a similar manner as in figure 30, were the intersections of the inputs were subtracted from the intersections of the inputs. A typical plot result of different time delays can be observed in figure 31, which is several delays given for different Full-Adders.

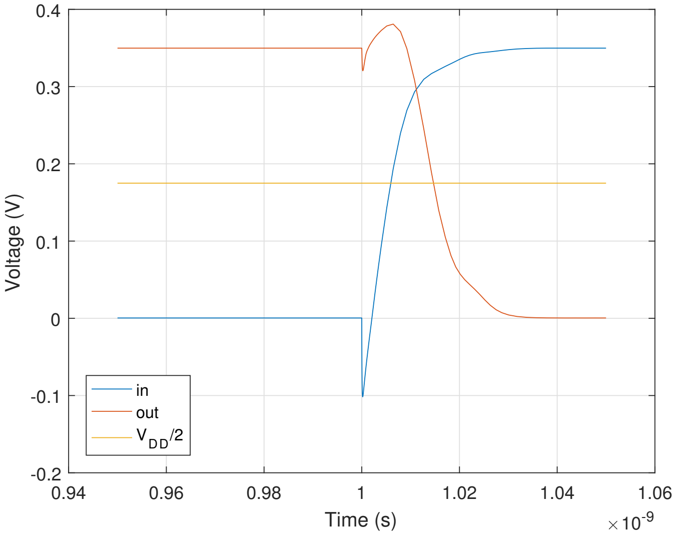
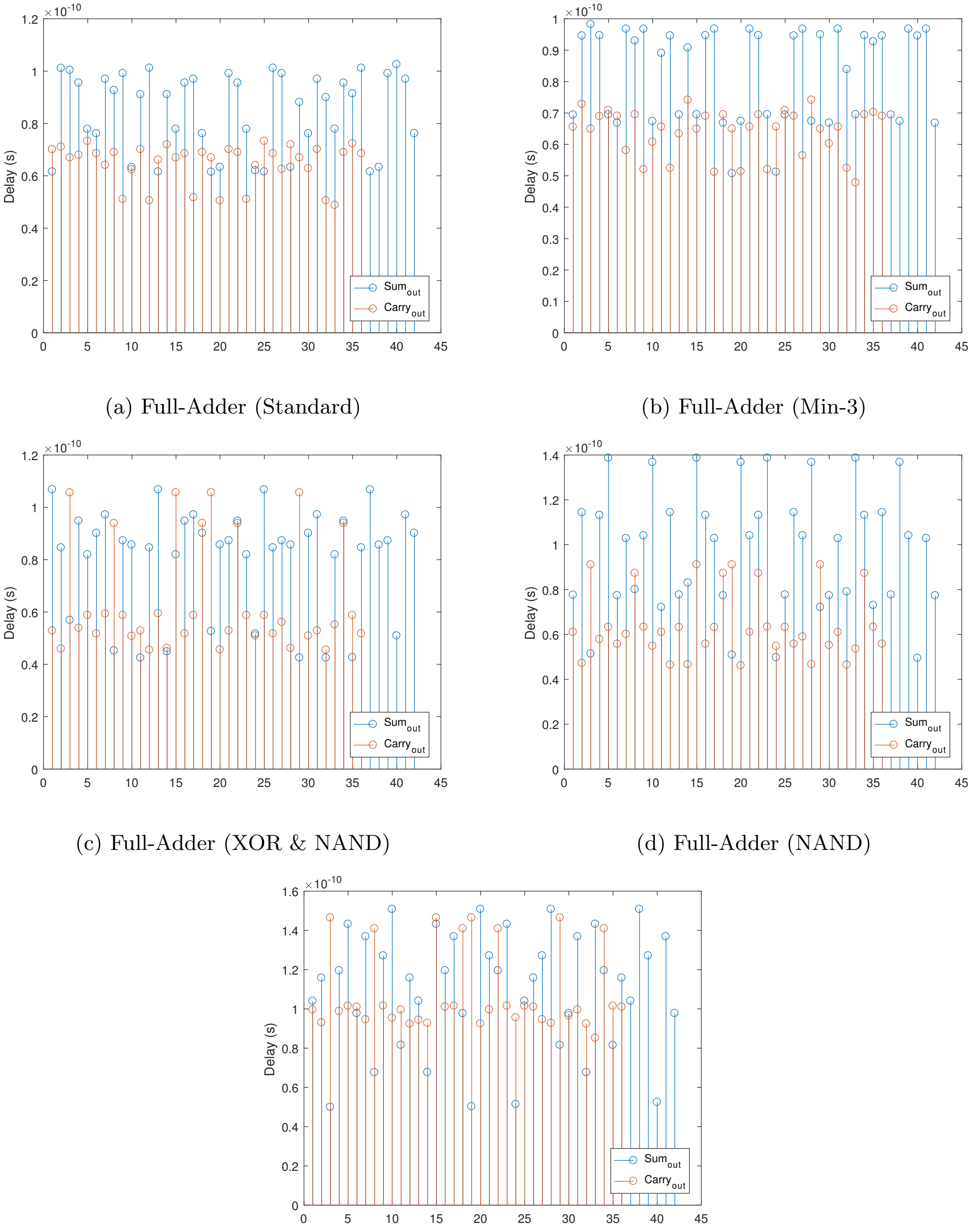


Figure 30: A typical transient analysis of time delay for devices, where the threshold values are given by the intersections of the $V_{DD}/2$ line (yellow). The simulated device here is an inverter of ***mosfet_low*** transistors at V_{DD}=350 mV, where the switching initiates at 1 ns. However, the delay is measured from approximately 1.005 ns to 1.015 ns, where the yellow line intersects the other lines.



(a) Full-Adder (Standard)

(b) Full-Adder (Min-3)

(c) Full-Adder (XOR & NAND)

(d) Full-Adder (NAND)

(e) Full-Adder (NOR)

Figure 31: Worst-case delay plots of different Full-Adders by ***mosfet_low*** transistors at V_{DD}=350 mV, showing the different delays for both outputs, $\text{carry}_{\text{out}}$ and sum_{out}. A Full-Adder has 64 possible different transitions, but not all combinations result in a changed output. Therefore, there are only 42 different Sum_{out}- and 36 different $\text{Carry}_{\text{out}}$ delays in this plot. The numbers on the X-axis does not directly indicate the input combination, but which number in the line of different pulses the combination was.

It can be observed from figure 31 that the different input combinations are of significance, as the differences sometimes differs by a factor of up to 3. It is also interesting to note that the delay for the $\text{Carry}_{\text{out}}$ are generally lower compared to Sum_{out} for all Full-Adders. However, it was observed during simulation that sometimes, either the carry- or sum signal would exceed the $\text{V}_{\text{DD}}/2$ threshold before stabilizing around the correct state. This could give false results when plotting the delay, and are therefore important to pay attention to. A typical example of this can be seen in figure 32. The worst-case time delays extracted in figure 31 is shown as logic transitions given in table 13.

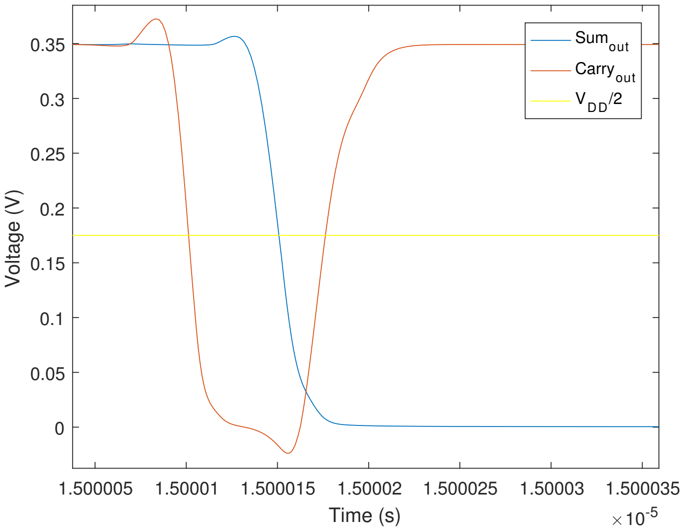


Figure 32: A glitch occurring when a NOR-based Full-Adder is switching state. In this case, the $\text{carry}_{\text{out}}$ signal is exceeding the $\text{V}_{\text{DD}}/2$ threshold line (yellow) when it should not, before returning to its final and correct state.

Table 13: An overview of the different states for worst-case input for different transistors and logic gates. The worst-case for delay are given by the transition from one state to another. A worst-case given as 10 → 00 means that port *a* is initially fed with a high input signal followed by low input signal. Port *b* has initially a low input signal, which also remains low after the switching. The energy or delay is extracted from the transition between the two input combinations.

		Worst-case transition (Port a, b, c)	
Logic gate	**Transistor type**	**Energy**	**Delay**
Minority-3	mosfet_high	n/a	110 → 010
	mosfet_low		000 → 110
	mosfet_low_bb		000 → 110
Half-Adder	mosfet_high	n/a	11 → 10
	mosfet_low		01 → 11
	mosfet_low_bb		01 → 11
Full-Adder (Std.)	mosfet_low	n/a	001 → 110
Full-Adder (Min-3)	mosfet_high	n/a	011 → 010
	mosfet_low		110 → 001
	mosfet_low_bb		011 → 010
Full-Adder (XOR & NAND)	mosfet_low	n/a	000 → 100
Full-Adder (NAND)	mosfet_low	n/a	011 → 111
Full-Adder (NOR)	mosfet_low	n/a	111 → 011

One can see from table 12 and 13 that the worst-case scenarios also are different relative to which type of transistor that is applied, and if body-bias is enabled. To derive the worst-case input for the D Flip-Flop, it is especially important to consider the previous state, as the final state may differ for the same input combination if the initial state is different. The worst-case scenarios for switching delay are shown in table 14.

Table 14: An overview of the different states for worst-case input for the D Flip-Flop. The worst-case input combinations for delay are given by the transition from one state to another. A worst-case given as 01,10 → 00,01 means that port *a* remains low during the switching, while *clk* goes from high to low. The two outputs, Q and $\overline{Q}$ then also switches. The delay is extracted from the transition between the two input combinations.

		Worst-case transition (Port a, clk, Q, $\overline{\mathbf{Q}}$)	
Logic gate	**Transistor type**	**Energy**	**Delay**
D Flip-Flop	mosfet_high	n/a	01,10 → 00,01
	mosfet_low		01,10 → 00,01
	mosfet_low_bb		01,10 → 00,01

The worst-case input combination for static power consumption is far easier to map, as it is not dependant on previous states (With exception for the D Flip-Flop). An overview of the worst-case scenarios can be seen in table 15.

Table 15: Worst-cse input combinations with respect to static power consumption. The first digit represent port a, the next port b and so on. For the D Flip-Flop, the two first digits represent the input port a and clk respectively. The two digits in parenthesis behind represents the state of the output ports, Q and $\overline{Q}$, as these states are dependant on the earlier input combination due to the memory property.

Logic gate	**Transistor type**	**Worst-case input (Port a, b ,c)**
Inverter	mosfet_high	1
	mosfet_low	1
	mosfet_low_bb	1
NAND	mosfet_high	01
	mosfet_low	11
	mosfet_low_bb	11
NOR	mosfet_high	01
	mosfet_low	01
	mosfet_low_bb	01
XNOR	mosfet_high	10
	mosfet_low	10
	mosfet_low_bb	10
XOR	mosfet_low	11
Minority-3	mosfet_high	011
	mosfet_low	110
	mosfet_low_bb	110
Half-Adder	mosfet_high	01
	mosfet_low	11
	mosfet_low_bb	11
Full-Adder (std.)	mosfet_low	111
Full-Adder (Min-3)	mosfet_high	101
	mosfet_low	110
	mosfet_low_bb	110
Full-Adder (NAND-XOR)	mosfet_low	011
Full-Adder (NAND)	mosfet_low	011
Full-Adder (NOR)	mosfet_low	100
D Flip-Flop	mosfet_high	11 (01)
	mosfet_low	01 (01)
	mosfet_low_bb	01 (01)

5.3 Deriving a supply voltage

As discussed in the Methods sections, a supply voltage of 350 mV was estimated to be an appropriate voltage for the given requirement specifications in this project. To verify the total delay of a 16by12bit-adder, a test bench similar to figure 23 was used to simulate realistic environments. Input combinations were then fed into the 16by12-bit adder according to the critical path given in the method chapter. A visual representation of the time delay can be seen in figure 33, which shows a typical transient analysis of a Ripple-Carry adder.

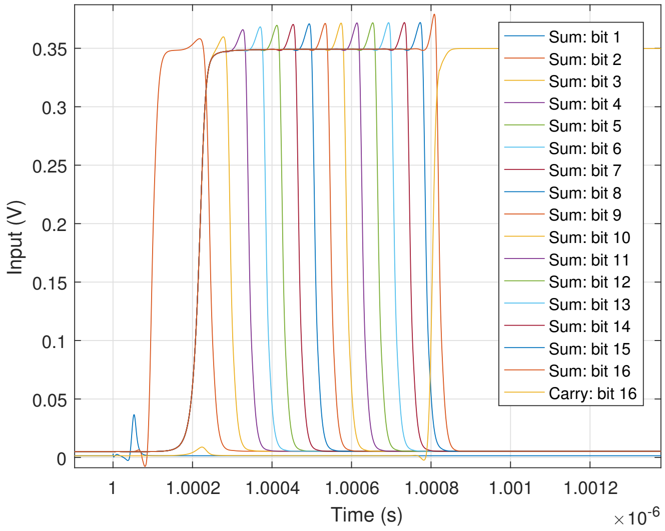


Figure 33: A transient analysis of time delay for a 16-bit Ripple-Carry adder under worst-case scenario input combination. The propagation of the *carry* signal time delays can clearly be seen by the adders coupled in series. The adder shown is simulated at a V_{DD}=350 mV and built up by ***mosfet_low*** transistors.

The total delays for the 5 different Ripple-Carry adders can be seen in table 16, and added together in equation 22. Table 17 gives the total delay for the 16by12-bit adder built up by different transistor types.

Table 16: Delay simulated by the critical path through n-bit Ripple-Carry adders. V_{DD}=350 mV.

	Ripple-Carry adder delay				
Transistor type	**12-bit adder**	**13-bit adder**	**14-bit adder**	**15-bit adder**	**16-bit adder**
mosfet_low	579.0 ps	632.7 ps	686.5 ps	740.2 ps	794.1 ps

Table 17: Total delay simulated by the critical path through a 16by12-bit Ripple-Carry adder. V_{DD}=350 mV.

Transistor type	**16by12-bit Ripple-Carry adder delay**
mosfet_low	3.4 ns

Comparing table 17 with the estimated delay in equation 22 seems to correspond reasonable. A calculated delay of 4.04 ns was calculated for the ***mosfet_low*** transistor 16by12-bit adder at V_{DD}=350 mV, compared to a simulated delay of 3.4 ns at the same supply voltage.

A simulated delay of 3.4 ns equals to a maximum operating speed of 294.1 MHz, which is more than sufficient for the 40 MHz limit given in the requirement specifications. However, these results are given under perfect theoretical conditions, and does not take parasitic resistances and capacitances into account, which should decrease performance significantly. When including parasitic capacitances introduced by layout design, a delay performance degradation was found to be reduced by a factor of up to 3.5 for the ***mosfet_low*** transistor in [41]. Considering that PVT-variations also will affect the performance, further performance degradation will occur. The delay measurements according to this experiment gives a safety margin with a factor of 7.4, which is considered enough for a supply voltage of V_{DD}=350 mV. Therefore, a supply voltage of V_{DD}=350 mV will be applied throughout the project for the ***mosfet_low*** transistor devices, as well as the ***mosfet_low*** with body-bias enabled.

5.4 Logic gates sizing & body-biasing

After a supply voltage was derived, logic gates were balanced for V_{DD}=350 mV, and can be seen in table 18. The more complex logic circuits like Half-Adders, Full-Adders and Flip-Flops are not shown in this table, as they built by the simpler ports given in the table.

Table 18: Transistor sizes and bulk-biasing. All devices in the table are optimized for a supply voltage of 350 mV.

		NMOS			**PMOS**		
Logic gate	**Transistor**	**Width**	**Length**	**Body-bias**	**Width**	**Length**	**Body-bias**
Inverter	mosfet_high	105 nm	20 nm	0 mV	100 nm	20 nm	-750 mV
	mosfet_low	105 nm	20 nm	0 mV	100 nm	20 nm	0 mV
	mosfet_low_bb	100 nm	20 nm	2000 mV	110 nm	20 nm	-2000 mV
NAND	mosfet_high	120 nm	20 nm	0 mV	100 nm	20 nm	-750 mV
	mosfet_low	125 nm	20 nm	0 mV	100 nm	20 nm	0 mV
	mosfet_low_bb	115 nm	20 nm	2000 mV	100 nm	20 nm	-2000 mV
NOR	mosfet_high	100 nm	20 nm	0 mV	175 nm	20 nm	-750 mV
	mosfet_low	100 nm	20 nm	0 mV	100 nm	20 nm	0 mV
	mosfet_low_bb	100 nm	20 nm	2000 mV	125 nm	20 nm	-2000 mV
XNOR	mosfet_high	100 nm	20 nm	0 mV	105 nm	20 nm	-750 mV
	mosfet_low	110 nm	20 nm	0 mV	100 nm	20 nm	0 mV
	mosfet_low_bb	100 nm	20 nm	2000 mV	110 nm	20 nm	-2000 mV
XOR	mosfet_high	n/a	n/a	n/a	n/a	n/a	n/a
	mosfet_low	110 nm	20 nm	0 mV	100 nm	20 nm	0 mV
	mosfet_low_bb	n/a	n/a	n/a	n/a	n/a	n/a
MIN-3	mosfet_high	105 nm	20 nm	0 mV	100 nm	20 nm	-750 mV
	mosfet_low	110 nm	20 nm	0 mV	100 nm	20 nm	0 mV
	mosfet_low_bb	100 nm	20 nm	2000 mV	105 nm	20 nm	-2000 mV
Full-Adder (Std.)	mosfet_high	n/a	n/a	n/a	n/a	n/a	n/a
	mosfet_low	100 nm	20 nm	0 mV	105 nm	20 nm	0 mV
	mosfet_low_bb	n/a	n/a	n/a	n/a	n/a	n/a

Each logic gate introduced in the theory chapter was designed and verified for their function after balancing dimensions and body-bias. For the final version of transistor sizes, forward body-bias was applied to the logic gates referred to as ***mosfet_low_bb***. FBB will admittedly lower threshold voltage and thus increase leakage currents, but reverse body-bias would on the other hand result in highly asymmetric layout design, and was therefor avoided. RBB is however not enabled for the flipped-wells structures in this library. FBB enables lower voltage operation without speed loss, which accounts for a more significant power reduction compared to leakage currents.

Transient analyzes were performed to verify the logic circuit's functionality, where every input combination was entered and compared the output with the respective truth tables. The simulations were performed with the test bench shown in figure 23. The schematics for each of the logic gates can be seen in the Appendix section. The different bit sequences received for the different logic gates can be seen in the respective figures, as the transient responses on the left (red).

5.4.1 Inverter

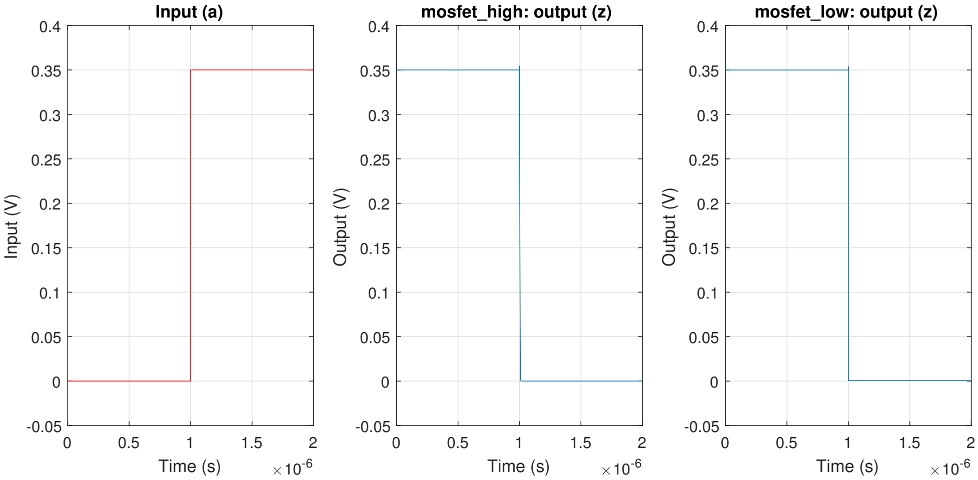


Figure 34: Transient simulation for an inverter composed of respectively ***mosfet_high*** and ***mosfet_low*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signal is shown on the plot to the left (red). The output signal is shown in the right side plots (blue).

5.4.2 NAND

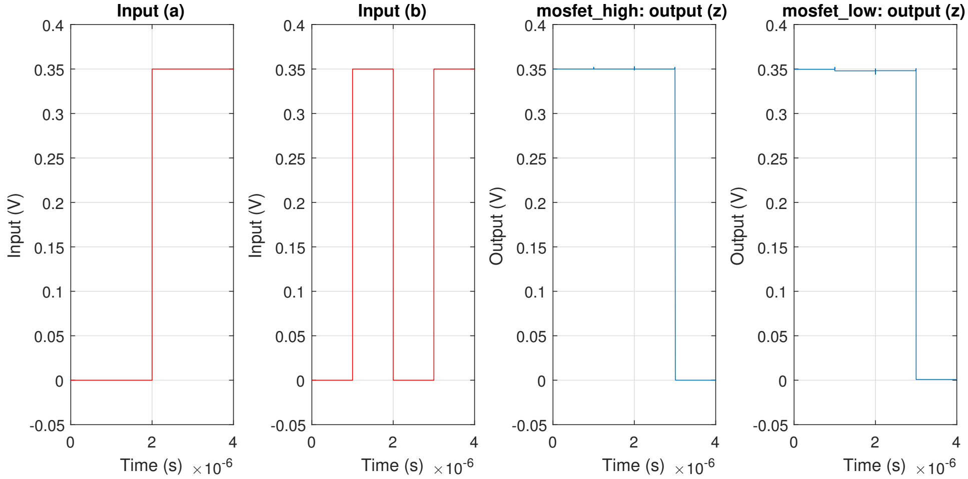


Figure 35: Transient simulation for a NAND-gate composed of respectively ***mosfet_high*** and ***mosfet_low*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

5.4.3 NOR

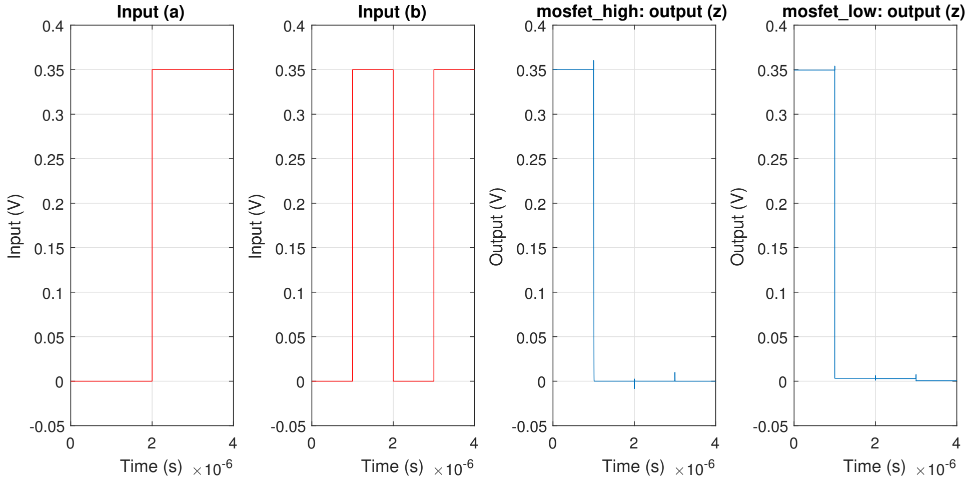


Figure 36: Transient simulation for a NOR-gate composed of respectively ***mosfet_high*** and ***mosfet_low*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

5.4.4 XNOR

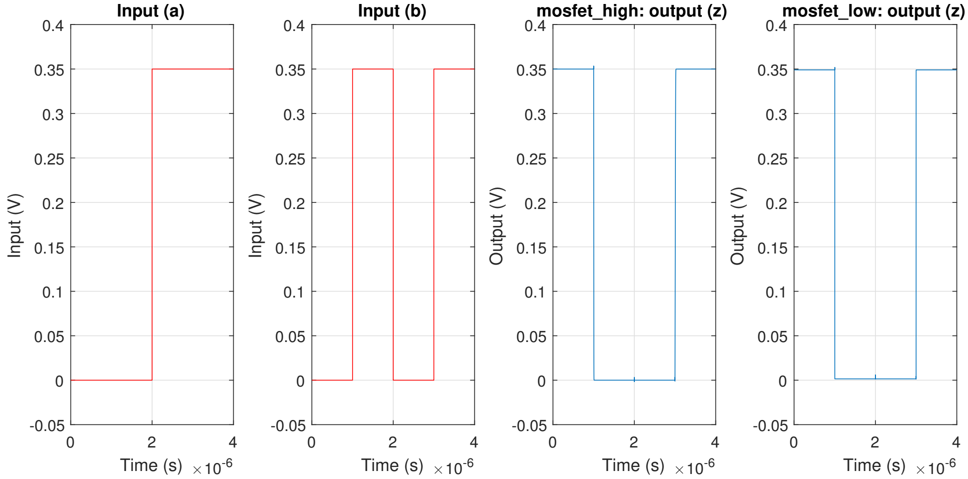


Figure 37: Transient simulation for a XNOR-gate composed of respectively ***mosfet_high*** and ***mosfet_low*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

5.4.5 XOR

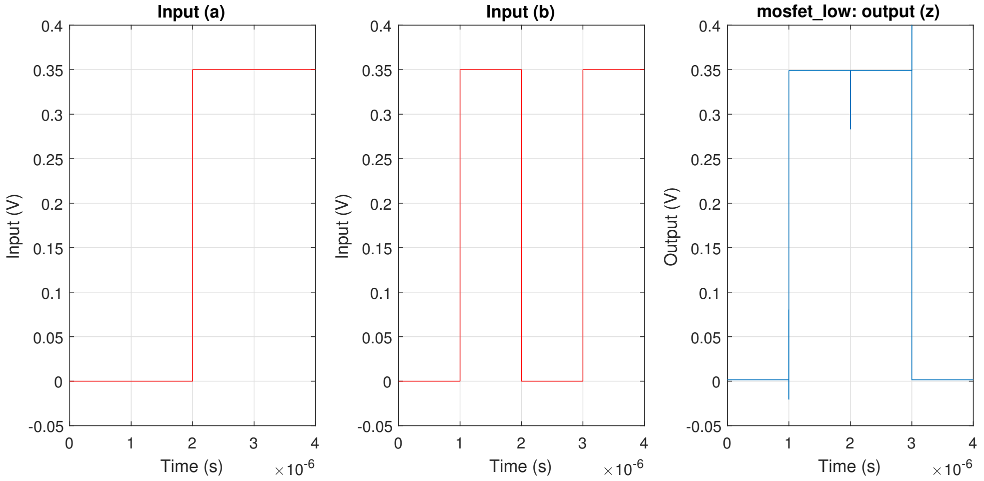


Figure 38: Transient simulation for a XOR-gate composed of ***mosfet_low*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

5.4.6 Minority-3

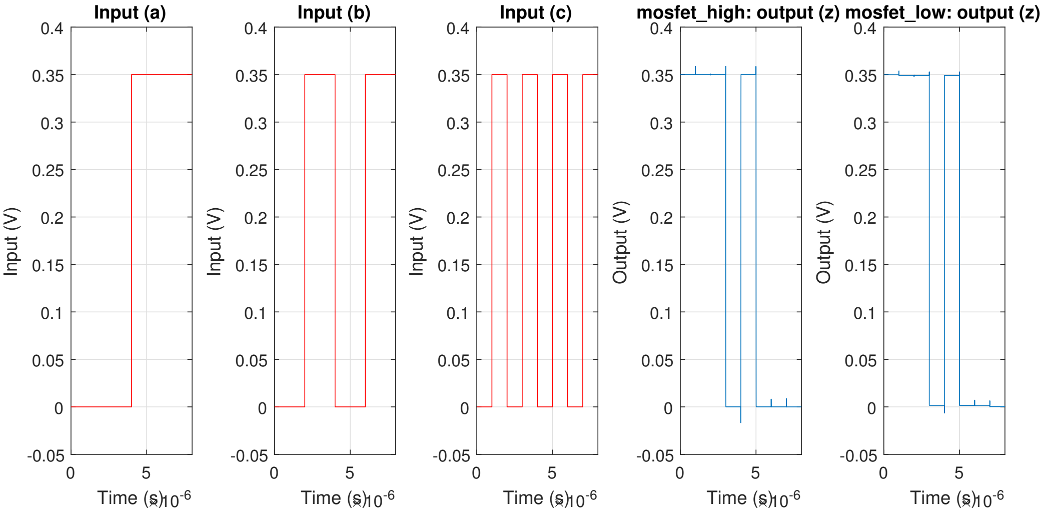


Figure 39: Transient simulation for a Minority-3 gate composed of respectively ***mosfet_high*** and ***mosfet_low*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

5.4.7 Half-Adder

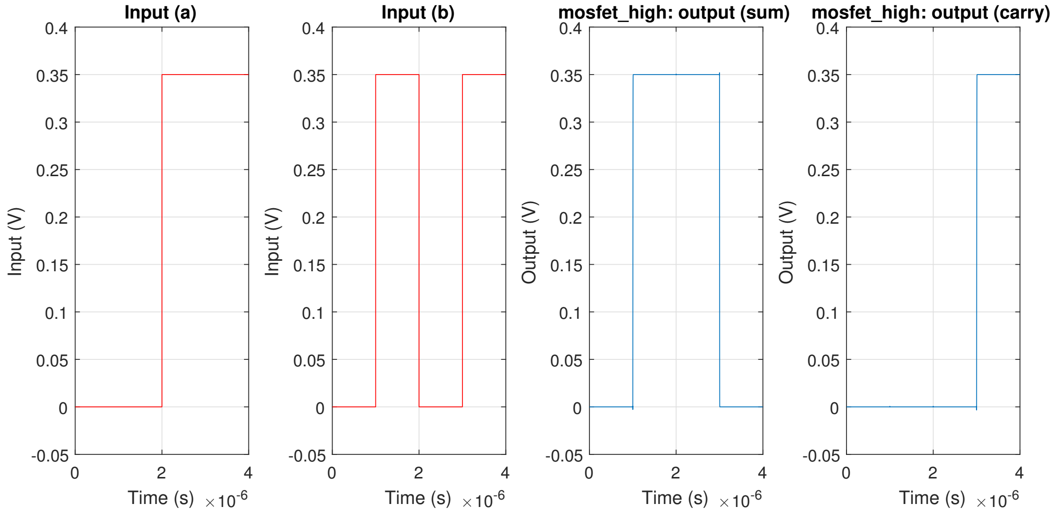


Figure 40: Transient simulation for a Half-Adder gate composed of ***mosfet_high*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

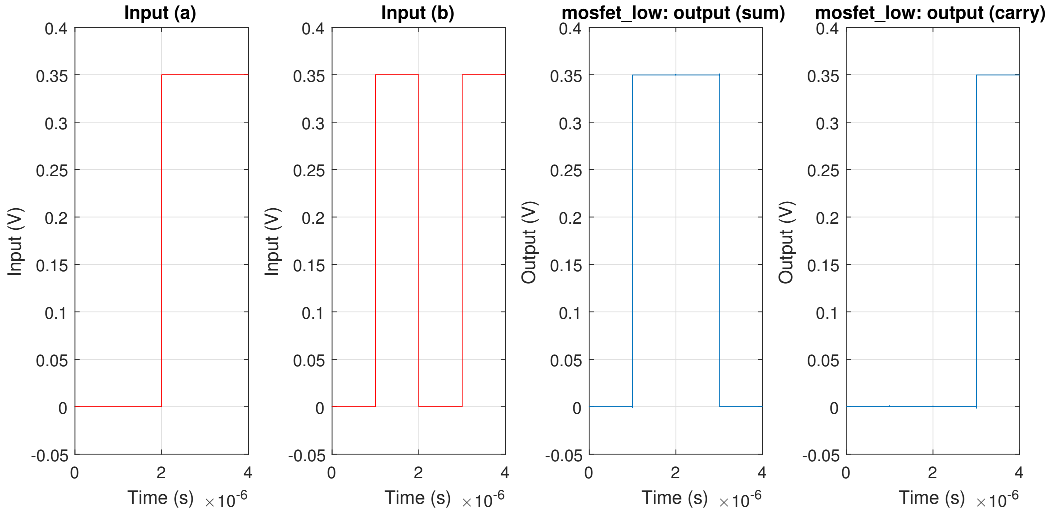


Figure 41: Transient simulation for a Half-Adder gate composed of ***mosfet_low*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

5.4.8 Full-Adder (Std.)

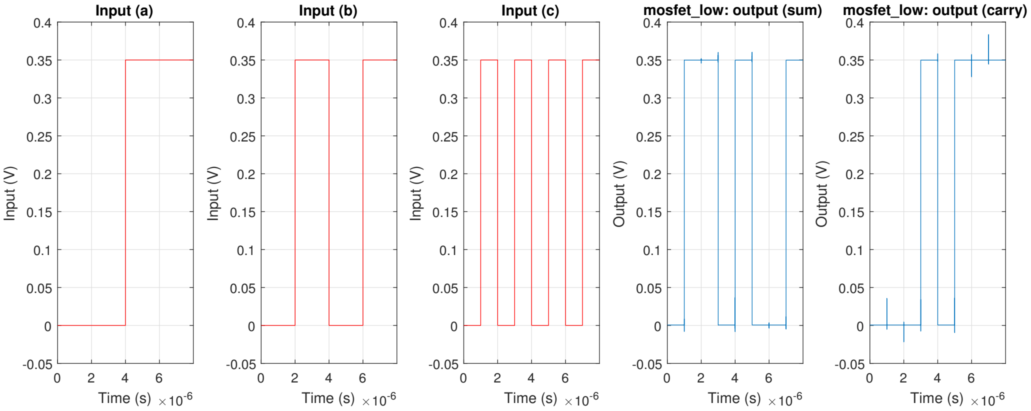


Figure 42: Transient simulation for a standard Full-Adder gate composed of ***mosfet_low*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

5.4.9 Full-Adder (Min-3)

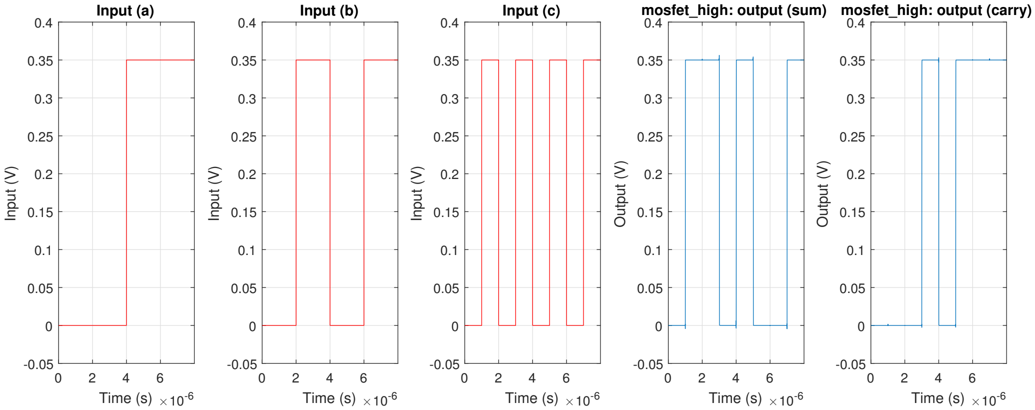


Figure 43: Transient simulation for a Minority-3 based Full-Adder gate composed of ***mosfet_high*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

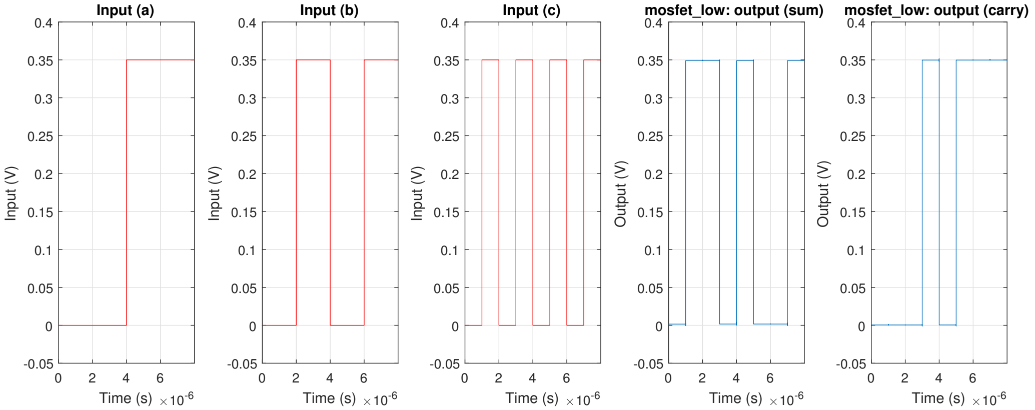


Figure 44: Transient simulation for a Full-Adder gate composed of ***mosfet_low*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

5.4.10 Full-Adder (NAND & XOR)

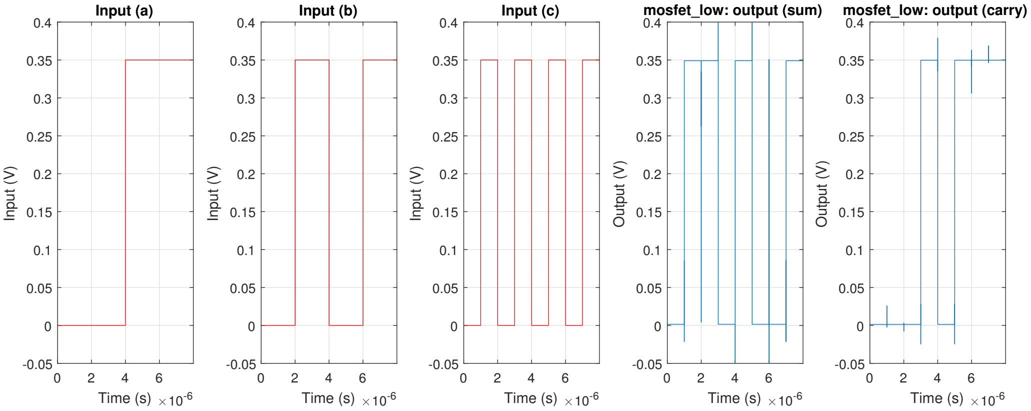


Figure 45: Transient simulation for a XOR & NAND-based Full-Adder gate composed of ***mosfet_low*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

5.4.11 Full-Adder (NAND)

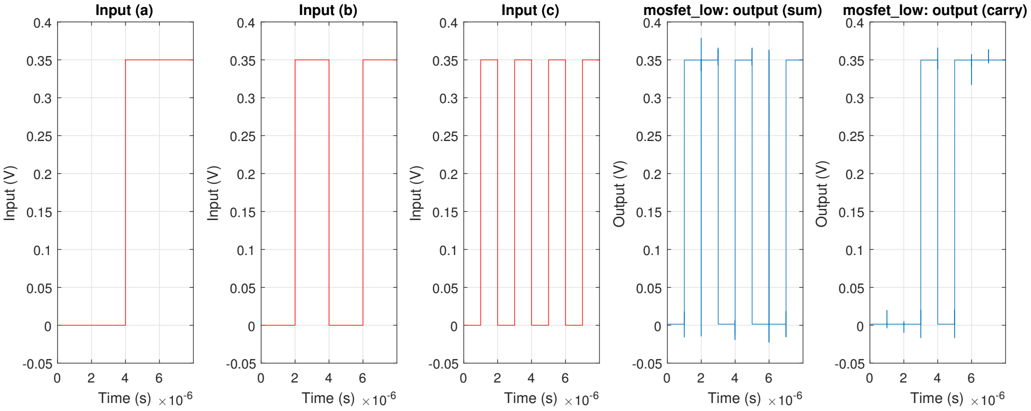


Figure 46: Transient simulation for a NAND-based Full-Adder gate composed of ***mosfet_low*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

5.4.12 Full-Adder (NOR)

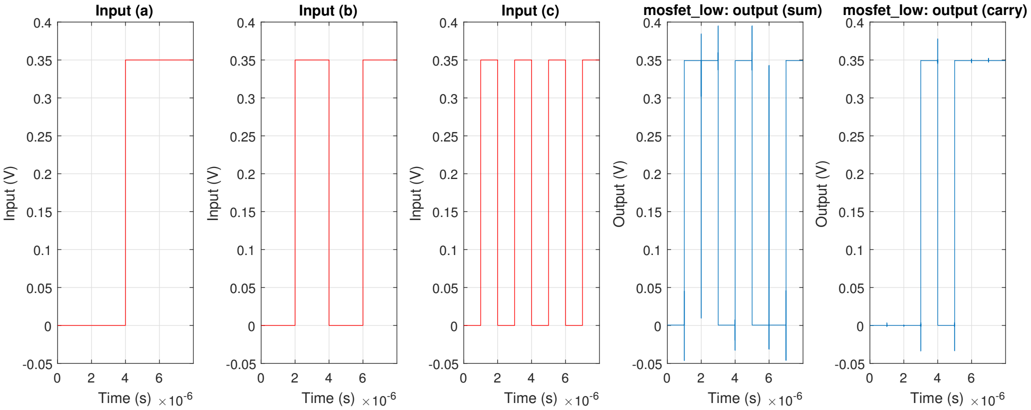


Figure 47: Transient simulation for a NOR-based Full-Adder gate composed of ***mosfet_low*** transistors, with a supply voltage of V_{DD}=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

5.4.13 D Flip-Flop

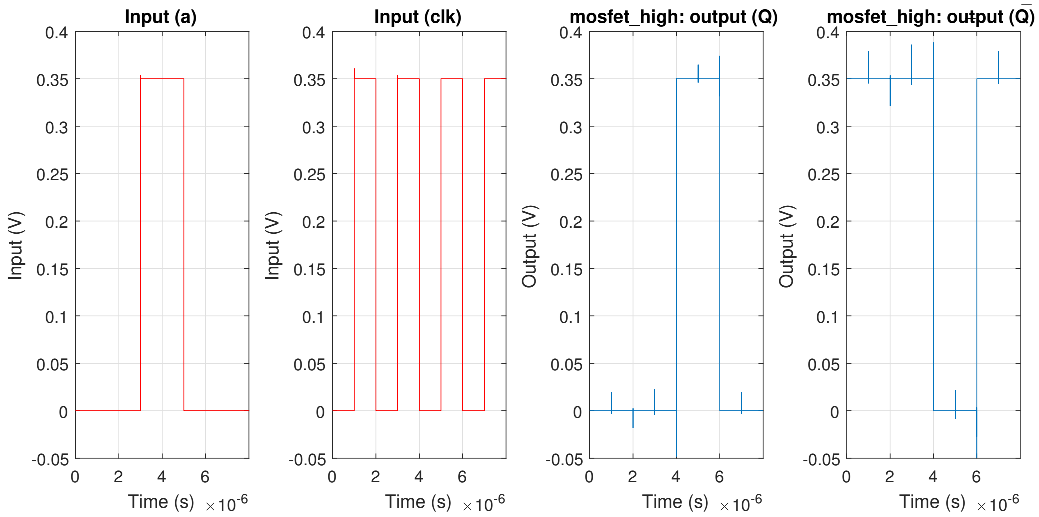


Figure 48: Transient simulation for a D Flip-Flop gate composed of ***mosfet_high*** transistors, with a supply voltage of V$_{\text{DD}}$=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

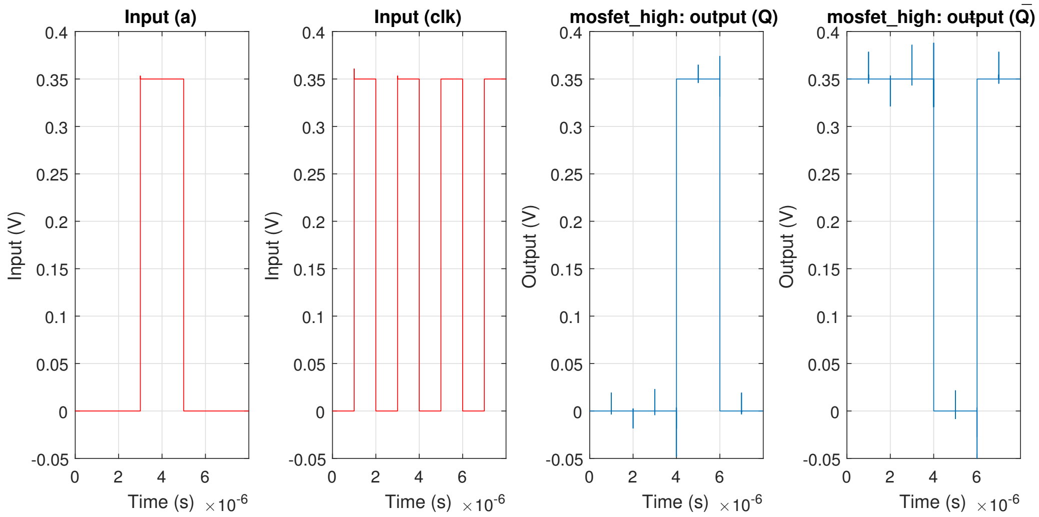


Figure 49: Transient simulation for a D Flip-Flop gate composed of ***mosfet_low*** transistors, with a supply voltage of V$_{\text{DD}}$=350 mV. Bit sequences were changed by intervals of 1 μs, and the received signals is shown on the plots to the left (red). The output signals is shown in the right side plots (blue).

5.5 Logic gate performance

The logic gates' average static power consumption was mapped, given by the worst-case of all the input combinations, as seen in table 19.

Table 19: Average static power consumption for all the different logic input combinations for each logic gate, by the worst-case input combinations for static power consumption seen in table 15. The simulations were performed with a supply voltage of V_{DD}=350 mV.

Logic gate	**Transistor type**	**Static power**
Inverter	mosfet_high	9.32 pW
	mosfet_low	16.33 nW
	mosfet_low_bb	498.85 nW
NAND	mosfet_high	15.11 pW
	mosfet_low	32.59 nW
	mosfet_low_bb	596.93 nW
NOR	mosfet_high	36.31 pW
	mosfet_low	16.12 nW
	mosfet_low_bb	613.34 nW
XNOR	mosfet_high	216.80 pW
	mosfet_low	54.08 nW
	mosfet_low_bb	1.69 μW
XOR	mosfet_low	63.00 nW
Minority-3	mosfet_high	149.90 pW
	mosfet_low	32.46 nW
	mosfet_low_bb	676.55 nW
Half-Adder	mosfet_high	505.3 pW
	mosfet_low	104.72 nW
	mosfet_low_bb	3.24 μW
Full-Adder (std.)	mosfet_low	80.22 nW
Full-Adder (Min-3)	mosfet_high	745.70 pW
	mosfet_low	104.13 nW
	mosfet_low_bb	2.89 μW
Full-Adder (NAND-XOR)	mosfet_low	155.56 nW
Full-Adder (NAND)	mosfet_low	156.95 nW
Full-Adder (NOR)	mosfet_low	145.18 nW
D Flip-Flop	mosfet_high	416.2 pW
	mosfet_low	88.47 nW
	mosfet_low_bb	3.97 μW

In addition, delay and switching energy consumption was also mapped for the logic gates designed in this report. This data is useful to estimate a complete circuit's performance and power dissipation. The results can be seen in table 20, where the simulations were performed with a worst-case scenario input combination given in table 12, 13 and 14.

Table 20: Energy consumption and time delay for switching by the different logic gates, under worst-case delay switchings given in table 12, 13 and 14. Supply voltage was V_{DD}=350 mV for all results, and ***mosfet_low_bb*** gates are ***mosfet_low*** gates with body-bias, as seen in table 18.

Logic gate	**Transistor type**	**Energy**	**Delay**
Inverter	mosfet_low	43.40 aJ	10.15 ps
	mosfet_low_bb	214.76 aJ	3.35 ps
NAND	mosfet_low	63.31 aJ	20.59 ps
	mosfet_low_bb	139.43 aJ	8.52 ps
NOR	mosfet_low	58.55 aJ	27.80 ps
	mosfet_low_bb	280.23 aJ	5.00 ps
XNOR	mosfet_low	48.04 aJ	36.77 ps
	mosfet_low_bb	688.40 aJ	10.38 ps
XOR	mosfet_low	57.94 aJ	34.01 ps
Minority-3	mosfet_low	105.39 aJ	32.90 ps
	mosfet_low_bb	229.92 aJ	14.03 ps
Half-Adder	mosfet_low	118.93 aJ	49.61 ps
	mosfet_low_bb	116.75 aJ	17.30 ps
Full-Adder (std.)	mosfet_low	116.52 aJ	83.16 ps
Full-Adder (Min-3)	mosfet_low	260.07 aJ	81.86 ps
	mosfet_low_bb	225.20 aJ	25.60 ps
Full-Adder (NAND-XOR)	mosfet_low	140.60 aJ	88.11 ps
Full-Adder (NAND)	mosfet_low	257.77 aJ	122.04 ps
Full-Adder (NOR)	mosfet_low	349.93 aJ	157.89 ps
D Flip-Flop	mosfet_low	130.42 aJ	75.6 ps
	mosfet_low_bb	322.61 aJ	23.15 ps

By looking at table 20, one can see that some of the switchings result in suspiciously low power consumption for a worst-case transition. This may occur because the worst-case switching combinations are mapped with respect to delay and not energy consumption. As seen in table 12, the worst-case input combinations are different for energy and delay, where only the inverter has the same worst-case transition for both energy and delay.

5.6 Layout design

The following figures in this section shows the layout design for the different logic gates presented in the theory chapter. All devices are designed in 22 nm FDSOI technology, and put in cells with a common height for later composition. All cells are without DRC and/or LVS errors, and are designed to be assembled into any combination with each other. Assembly requires a common height for all cells, with power supply- and body-bias rails lined up for overlap. The final library gives the possibility to create almost any system with the given logic gates. All layout designs, except the Ripple-Carry Adders, only use the two lowest layers of metal for routing in between the transistors. It is reported by the technology provider that synthesis is not enabled for routing with these metal layers, and therefore the levels from metal-3 and higher should be avoided as far as possible in the individual cells.

Layout design was made for both ***mosfet_low*** and ***mosfet_low_bb*** devices, with dimensions seen in table 18. Higher resolution images of layout designs can also be seen in the appendix section.

5.6.1 0 V body-bias

The technology supplier specifies that bulk connections can be placed as far 80 μm from a transistor, so therefore the bulk connections were made as individual cells and placed wherever they were required. The design can be seen in figure 50. The bulk connection includes connections for bulk nodes, as well as the dummy poly-silicon gates required at the edges of the regular transistor cells.

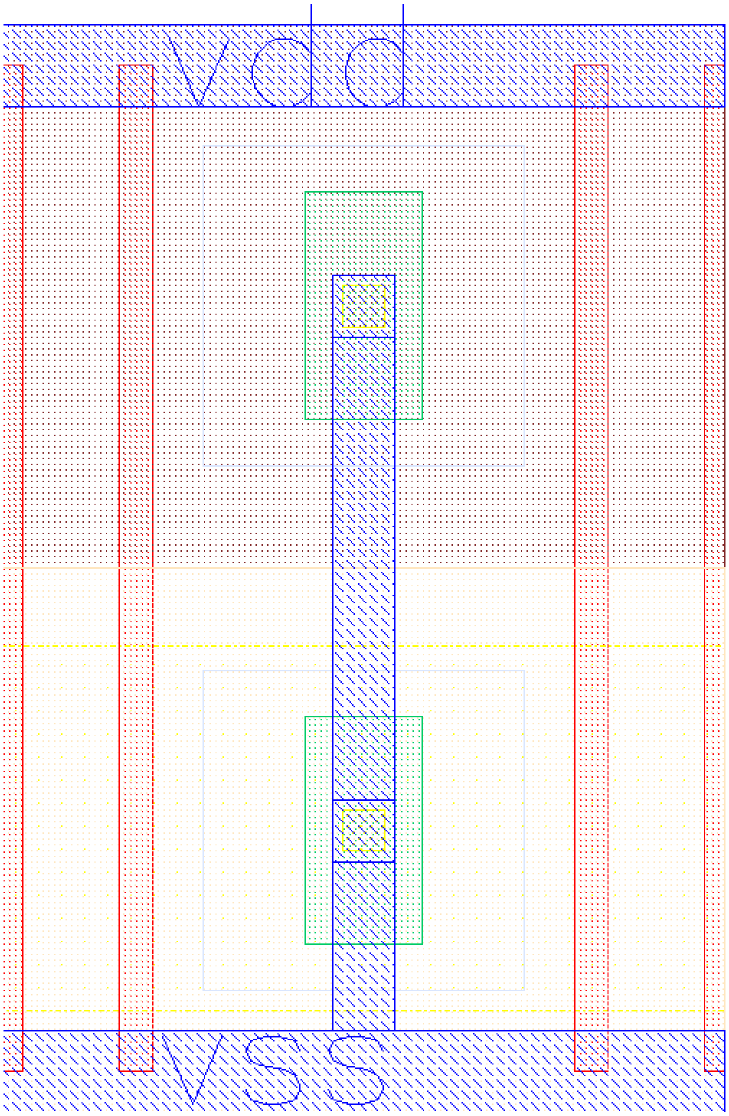


Figure 50: Bulk: 0 V body-bias without body-bias rails.

The height of the cells must be equal for all gates which are to be assembled later. Therefore, the height was derived from the gate which contains the largest transistor widths. By looking at table 18, the largest transistor width for the ***mosfet_low*** library is two 125 nm wide NMOS transistors in the NAND-gate. The p-doped well region must then be made large enough to contain a 125 nm wide transistor. This equals to a height of 531 nm for ***mosfet_low*** gates, including the supply rails. Since devices also will overlap each other later, the n-doped well area for the PMOS transistors must also have the same height to avoid DRC errors, and resulted in a total height of 1.062 μm.

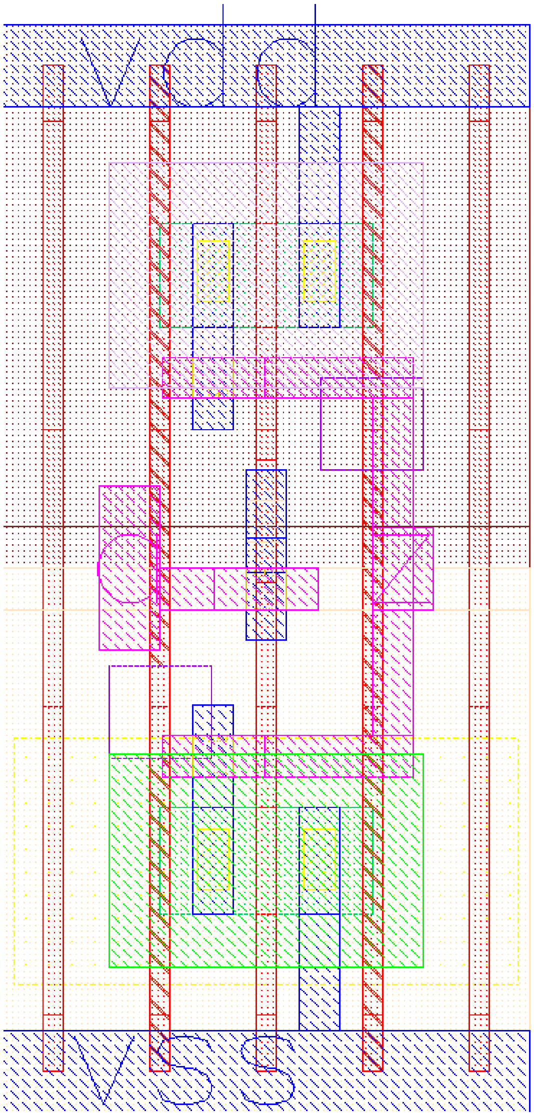


Figure 51: Inverter: 0 V body-bias, W x H = 0.416 μm x 1.062 μm.

The red vertical lines are the poly-silicon gate contacts, and though only one should be required for an inverter, DRC requires several dummy gates at each side. The two poly-silicon gates seen on each side of the inverter in figure 51 are dummy gates, and occupies a significant area of the layout design. However, the utmost dummy gates are meant to overlap when assembled with other gates. This also indicates that more complex gates of more transistors, have the potential for a higher density of transistors on a given area.

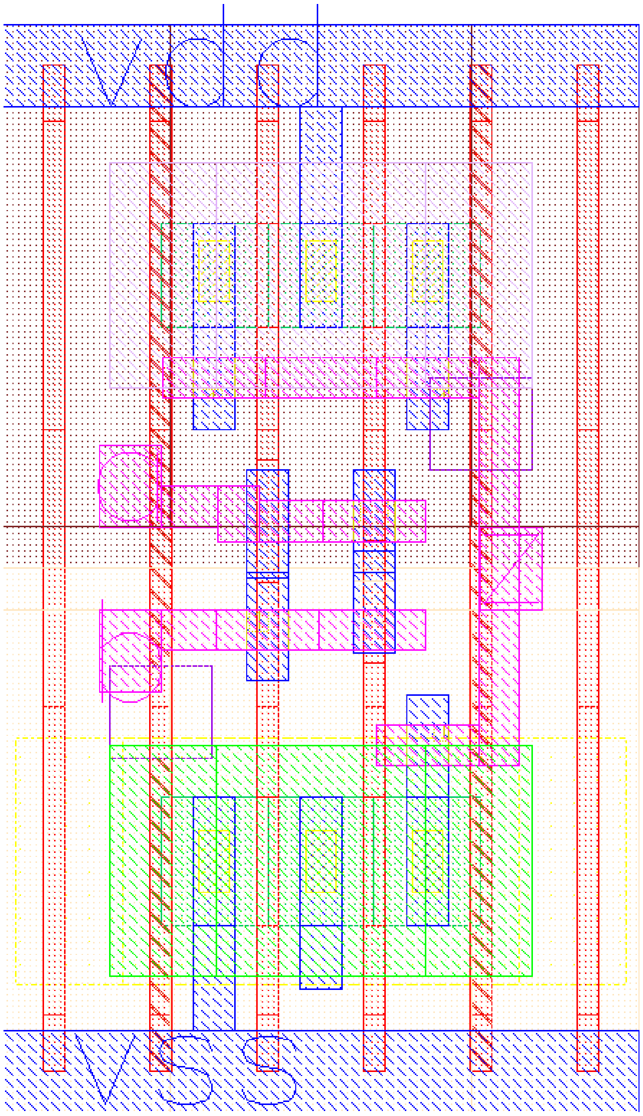


Figure 52: NAND: 0 V body-bias, W x H = 0.620 μm x 1.062 μm.

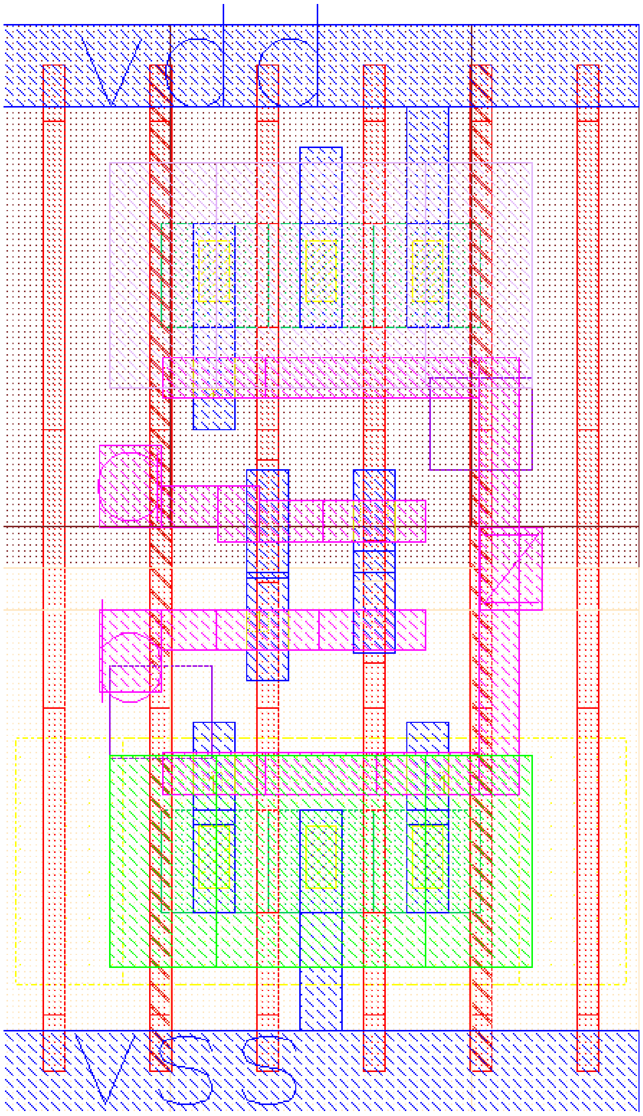


Figure 53: NOR: 0 V body-bias, W x H = 0.620 μm x 1.062 μm.

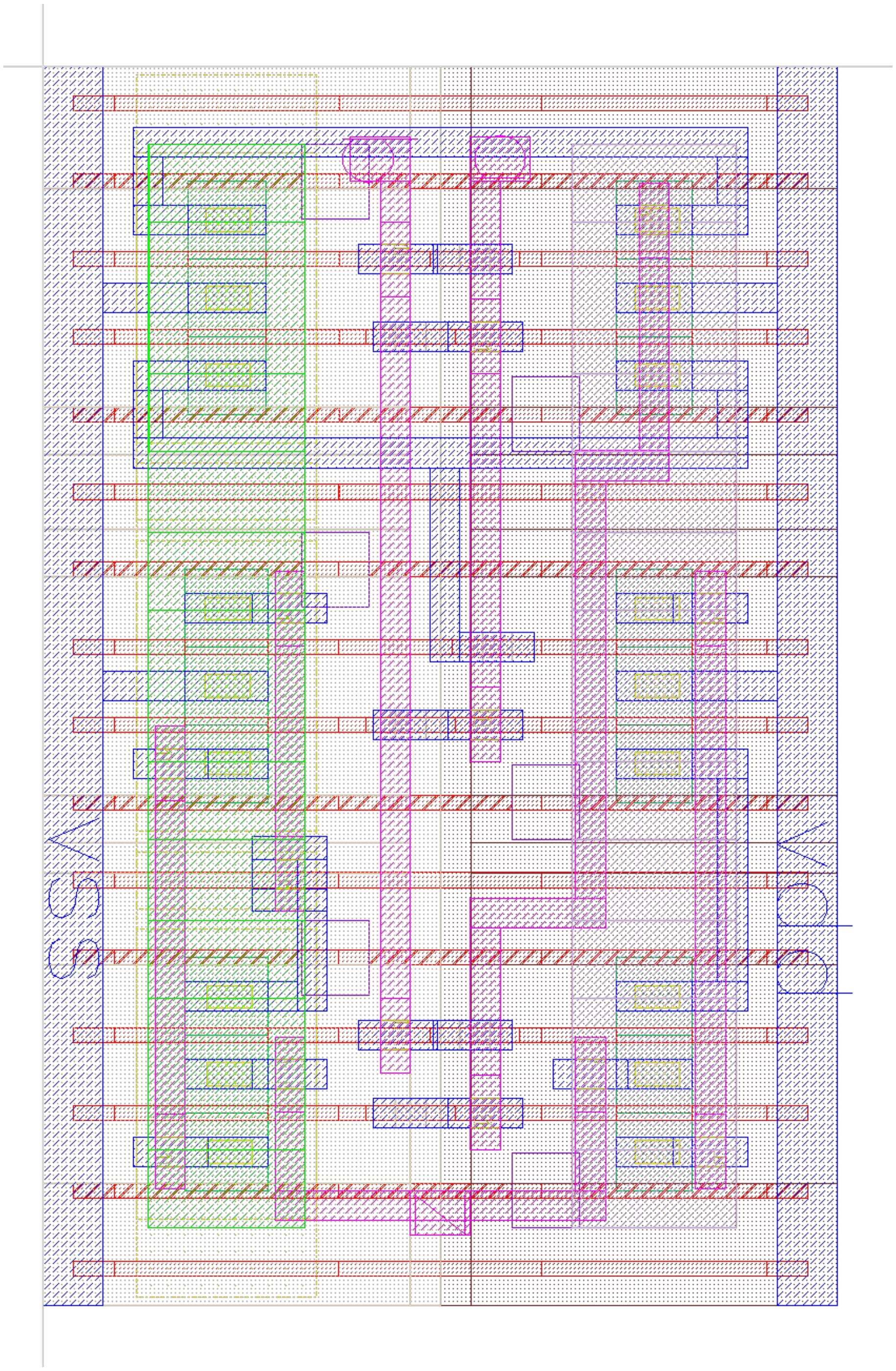


Figure 54: XNOR: 0 V body-bias, W x H = 1.560 μm x 1.062 μm.

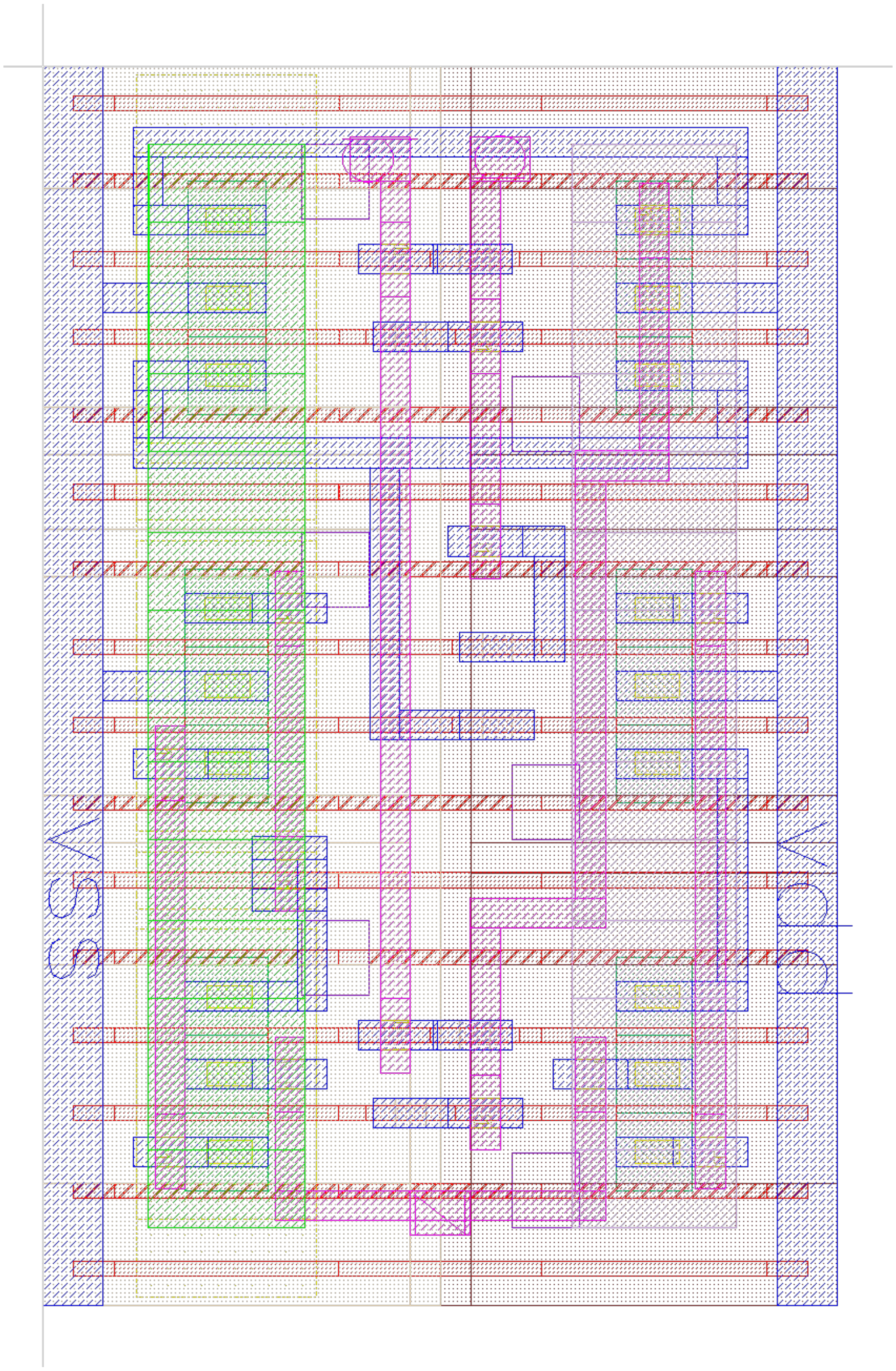


Figure 55: XOR: 0 V body-bias, W x H = 1.560 μm x 1.062 μm.

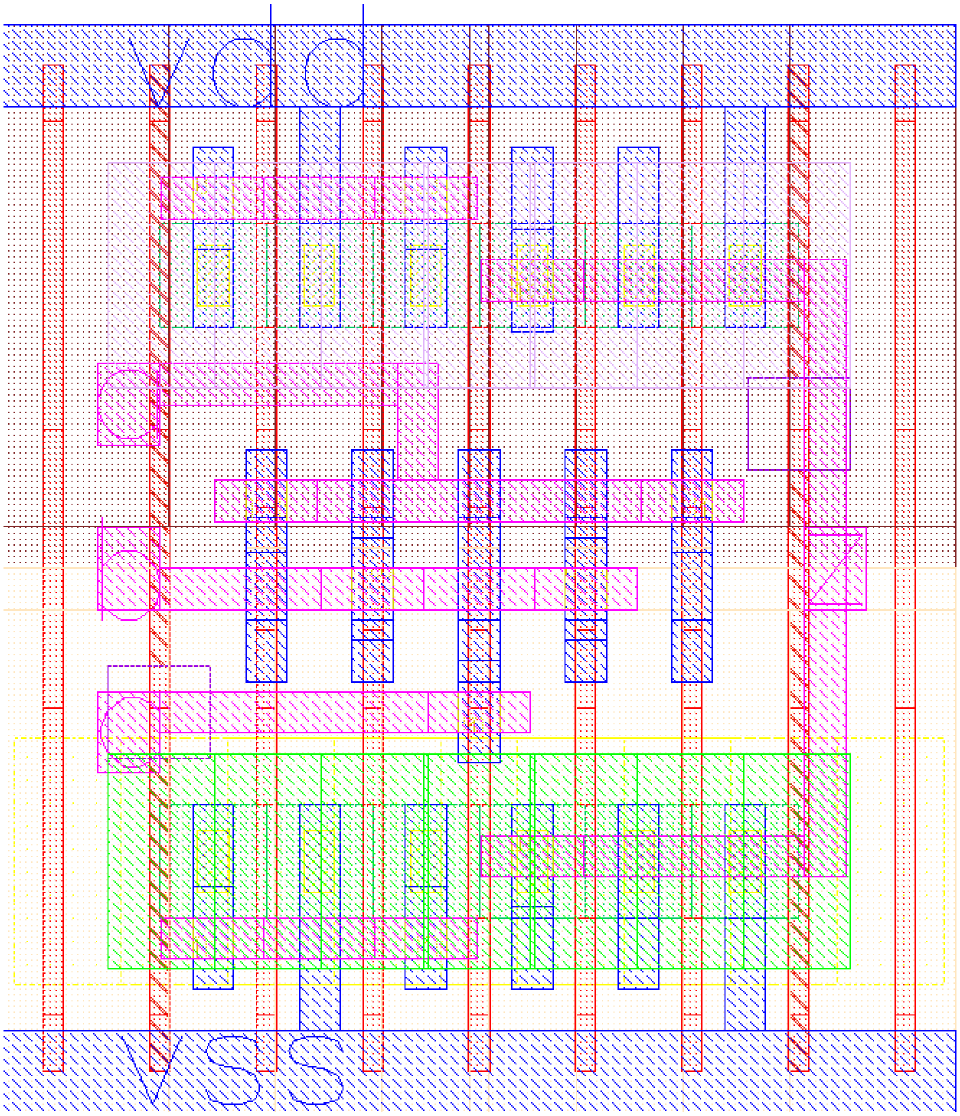


Figure 56: Minority-3: 0 V body-bias, W x H = 0.832 μm x 1.062 μm.

The Minority-3 gate by ***mosfet_low*** transistors shown in figure 56 achieves the highest density of transistors in a single cell, of all the basic logic gates in this report. 10 transistors on an area of 0.832 μm x 1.062 μm = 0.884 μm^2 gives an average layout area of 88.4 nm^2 per transistor.

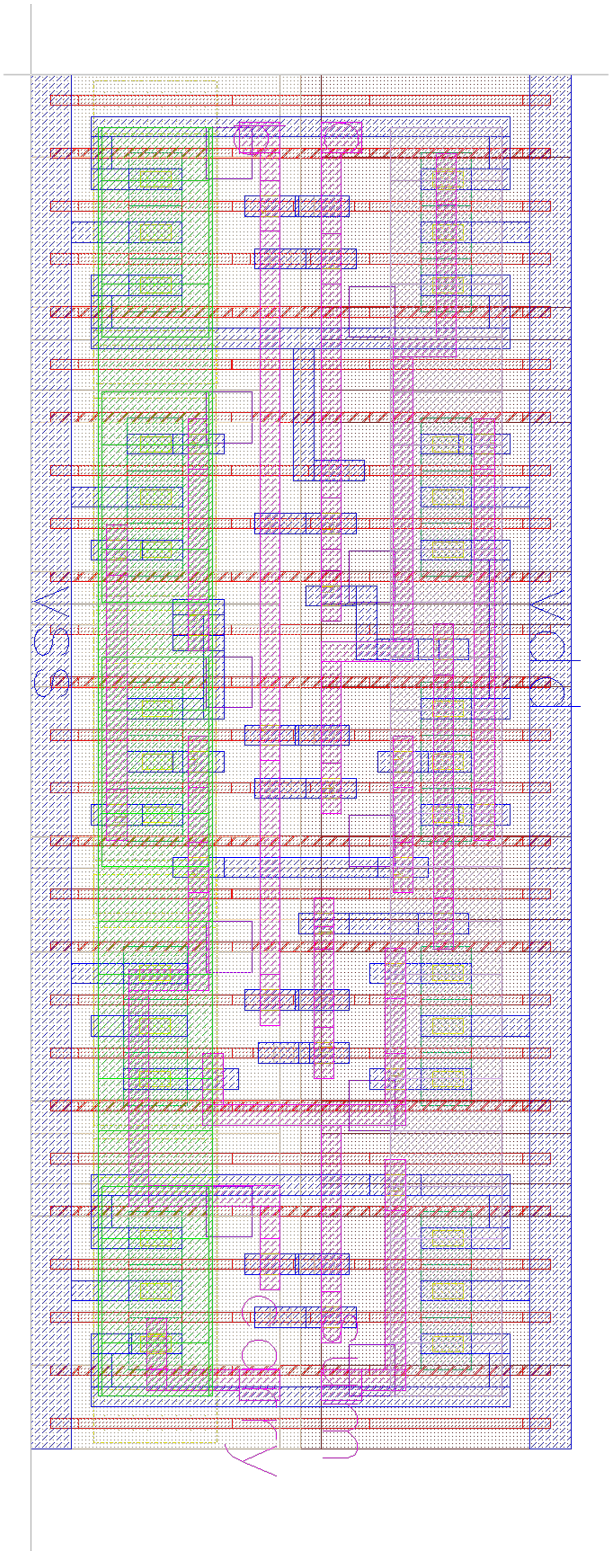


Figure 57: Half-Adder: 0 V body-bias, W x H = 2.600 μm x 1.062 μm.

The Half- and Full-Adders (Except the standard Full-Adder in figure 58) are composed of simpler logic gates like Inverters, NAND-, NOR-, XOR-, XNOR- and Minority-3 gates. The layout design for these adders were then assembled by the layout designs of these gates, though some modifications were done to optimize space occupation and routing.

The Minority-3 based Full-Adder is the basic logic gate which achieves the absolutely highest density of transistors, which is on average 84.5 nm^2 per transistor.

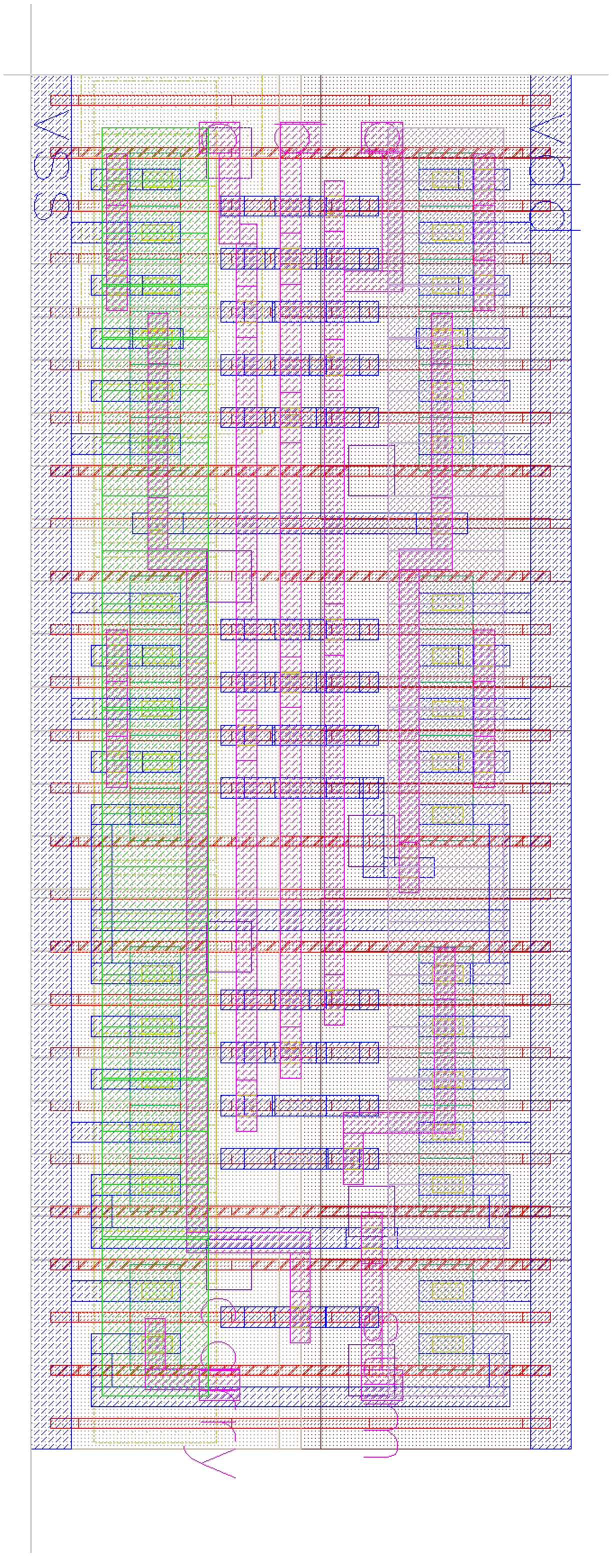


Figure 58: Full-Adder (Standard): 0 V body-bias, W x H = 2.600 μm x 1.062 μm.

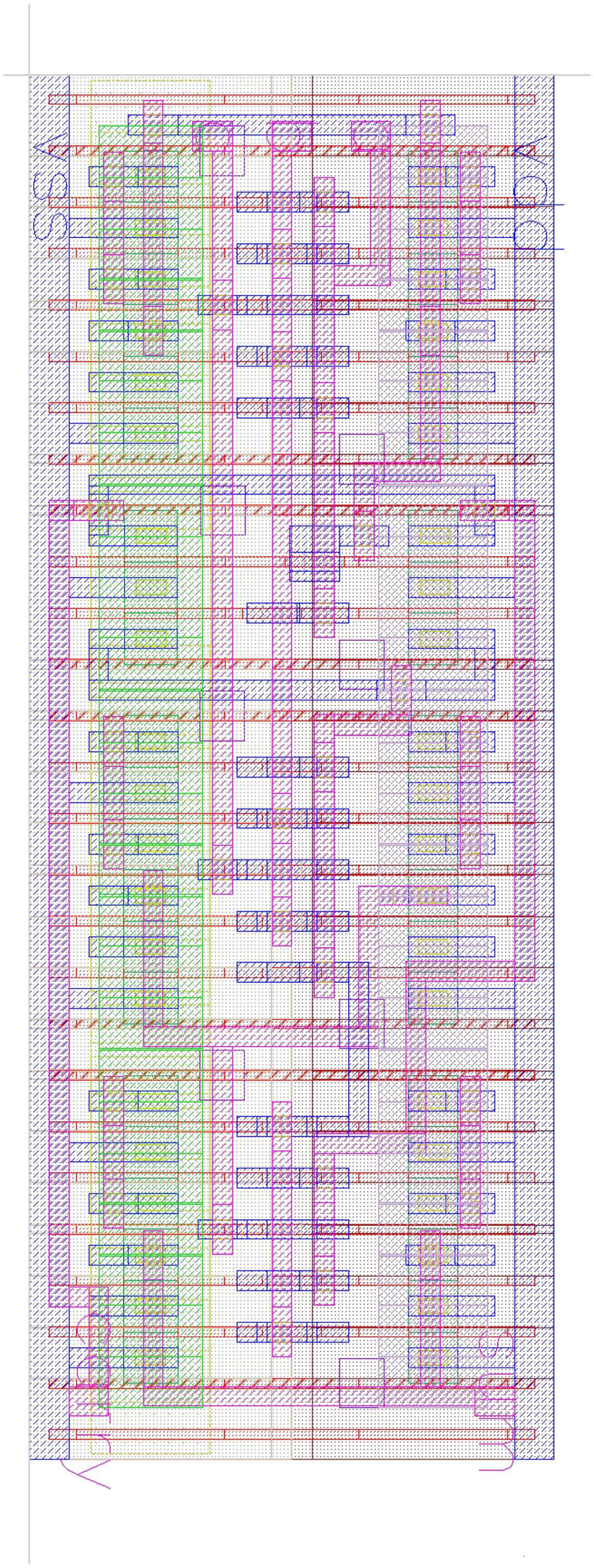
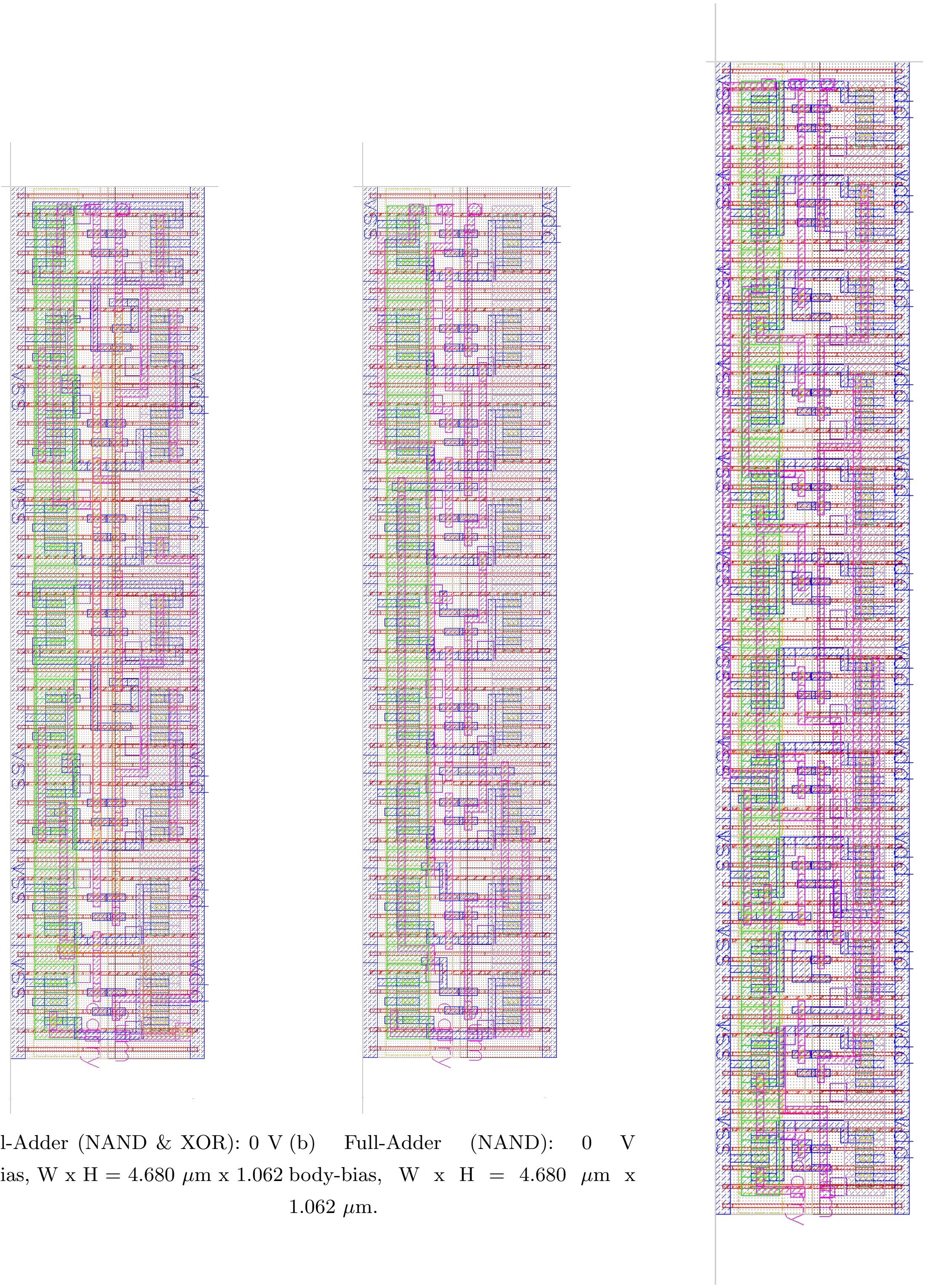


Figure 59: Full-Adder (Min-3): 0 V body-bias, W x H = 2.704 μm x 1.062 μm.



(a) Full-Adder (NAND & XOR): 0 V body-bias, W x H = 4.680 μm x 1.062 μm.

(b) Full-Adder (NAND): 0 V body-bias, W x H = 4.680 μm x 1.062 μm.

(c) Full-Adder (NOR): 0 V body-bias, W x H = 6.240 μm x 1.062 μm.

Figure 60: Layout designs of Full-Adders by NAND- & XOR-gates (left), NAND-gates (middle) and NOR-gates (right).

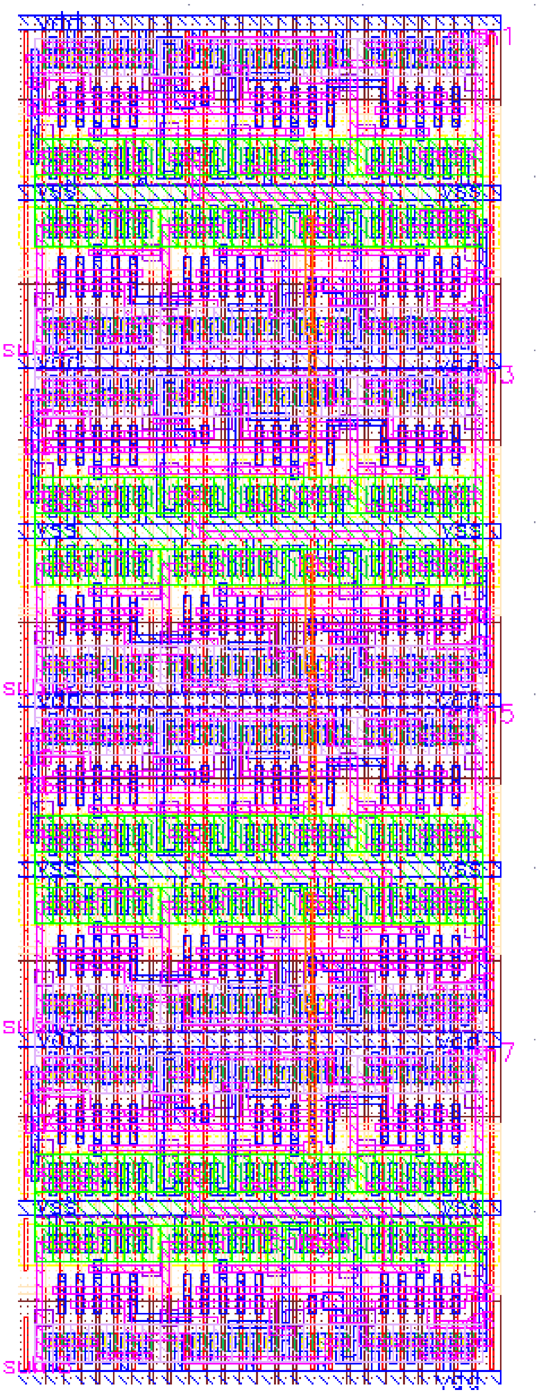
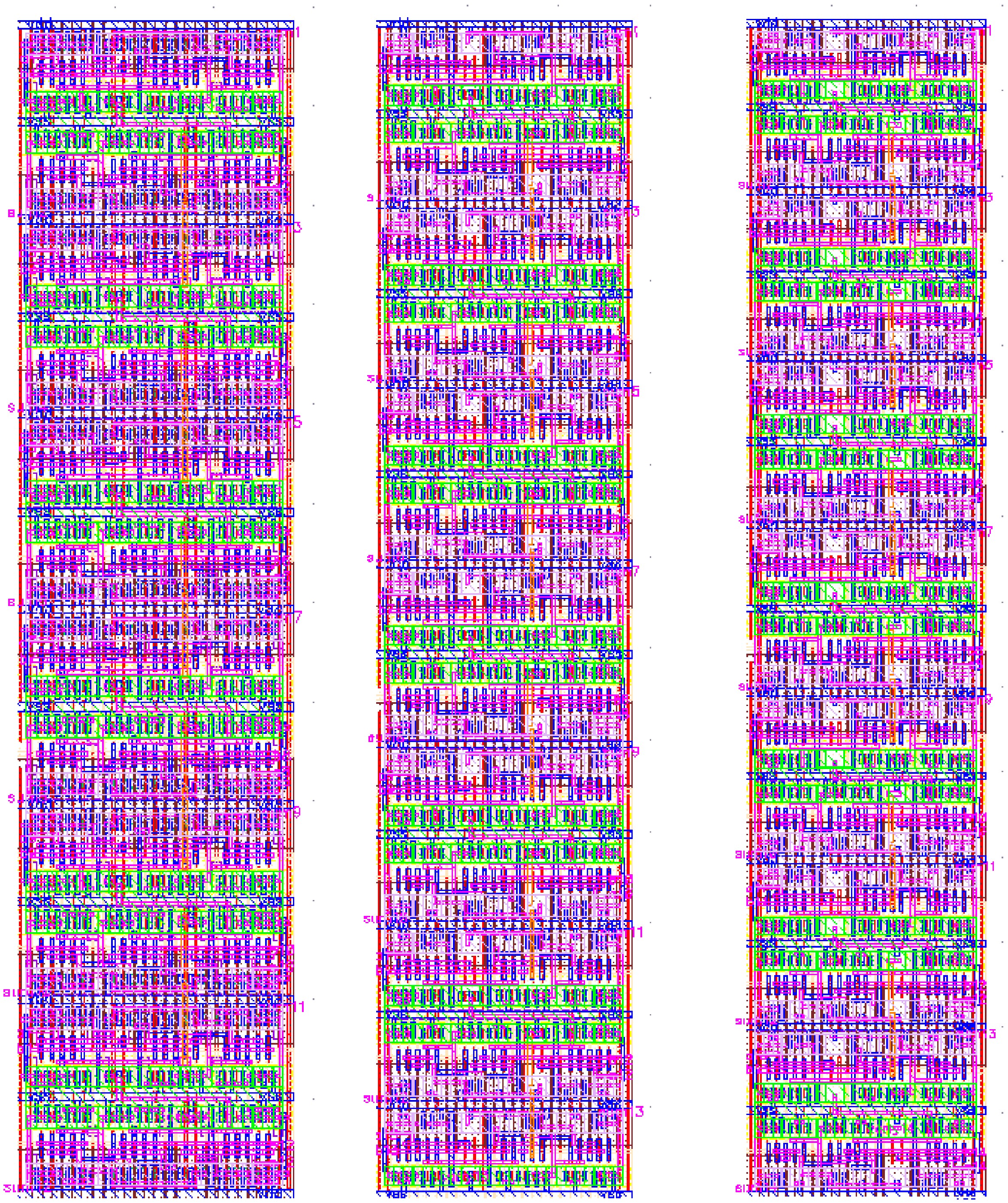


Figure 61: Layout design of an 8-bit Ripple-Carry Adder by ***mosfet_low*** transistors, given by the schematic in figure 104. Width and height of the cell is respectively 2.704 μm and 7.936 μm.

As seen in figure 61, 8 Minority-3 based Full-Adder cells are stacked vertically to form an 8-bit Ripple-Carry Adder. A total of 272 transistors are implemented into this design, and every other cell was flipped vertically so that power- or ground rail may be shared, and thus reduce layout area. This method would however be more problematic to apply for layout cells with body-bias rails.

The 12-, 13-, 14-, 15- and 16-bit Ripple-Carry adders seen in figures 62 and 63 are built in the similar way as the 8-bit Ripple-Carry adder. The Full-Adders are only stacked vertically, so width of the cell remains the same for all the Ripple-Carry adders. The 12-, 13- and 14-bit adders in figure 62 consists of 408, 442 and 476 transistors, respectively. The 15- and 16-bit adders requires 510 and 544 transistors respectively.

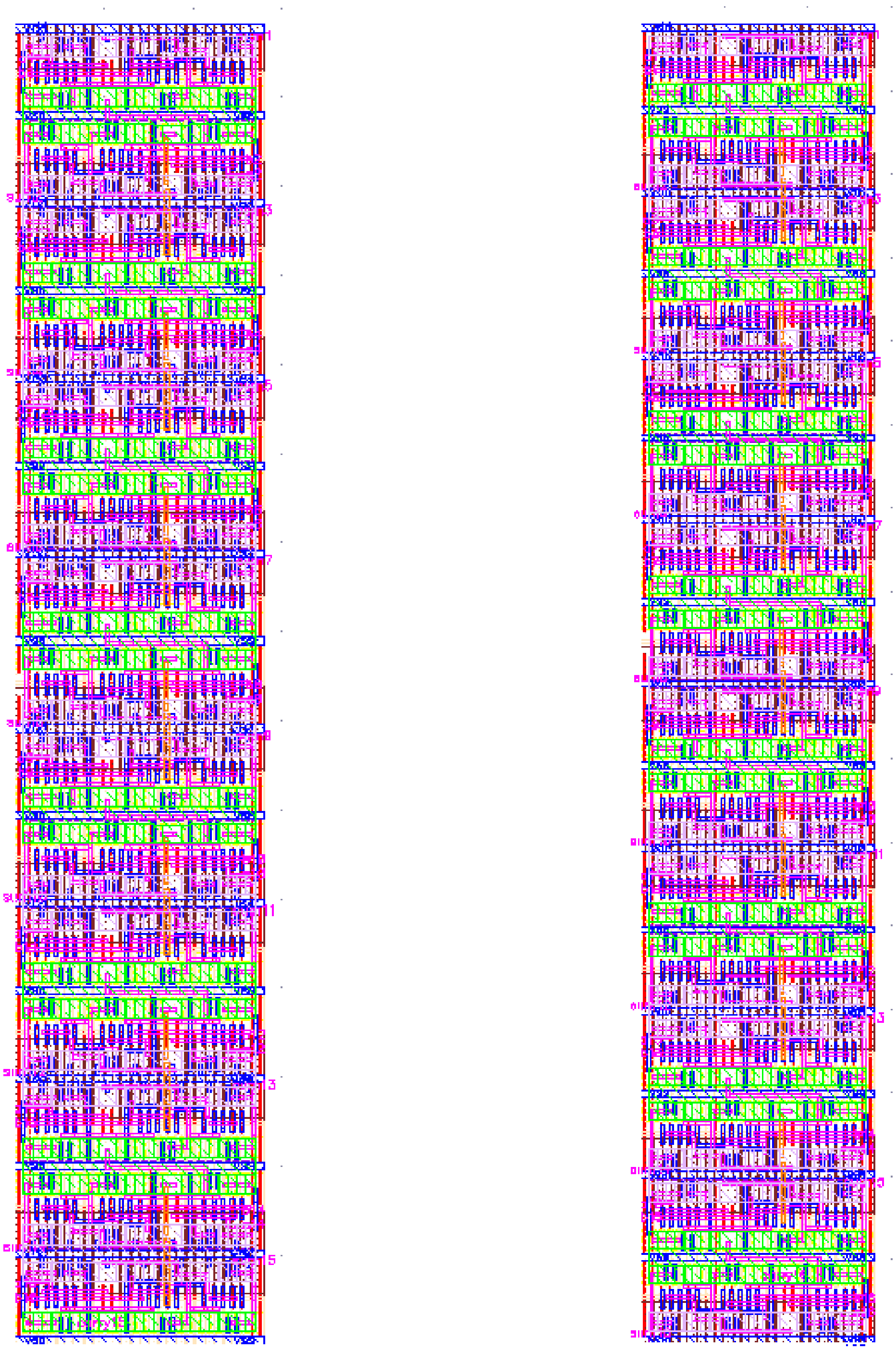


(a) Full-Adder (12-bit): 0 V body-bias, W x H = 2.704 μm x 11.864 μm.

(b) Full-Adder (13-bit): 0 V body-bias, W x H = 2.704 μm x 12.846 μm.

(c) Full-Adder (14-bit): 0 V body-bias, W x H = 2.704 μm x 13.828 μm.

Figure 62: Layout designs of 12-, 13-, and 14-bit Ripple-Carry Adders, by ***mosfet_low*** transistors.



(a) Full-Adder (15-bit): 0 V body-bias, W x H = 2.704 μm x 14.810 μm.

(b) Full-Adder (16-bit): 0 V body-bias, W x H = 2.704 μm x 15.792 μm.

Figure 63: Layout designs of 15- and 16-bit Ripple-Carry Adders, by ***mosfet_low*** transistors.

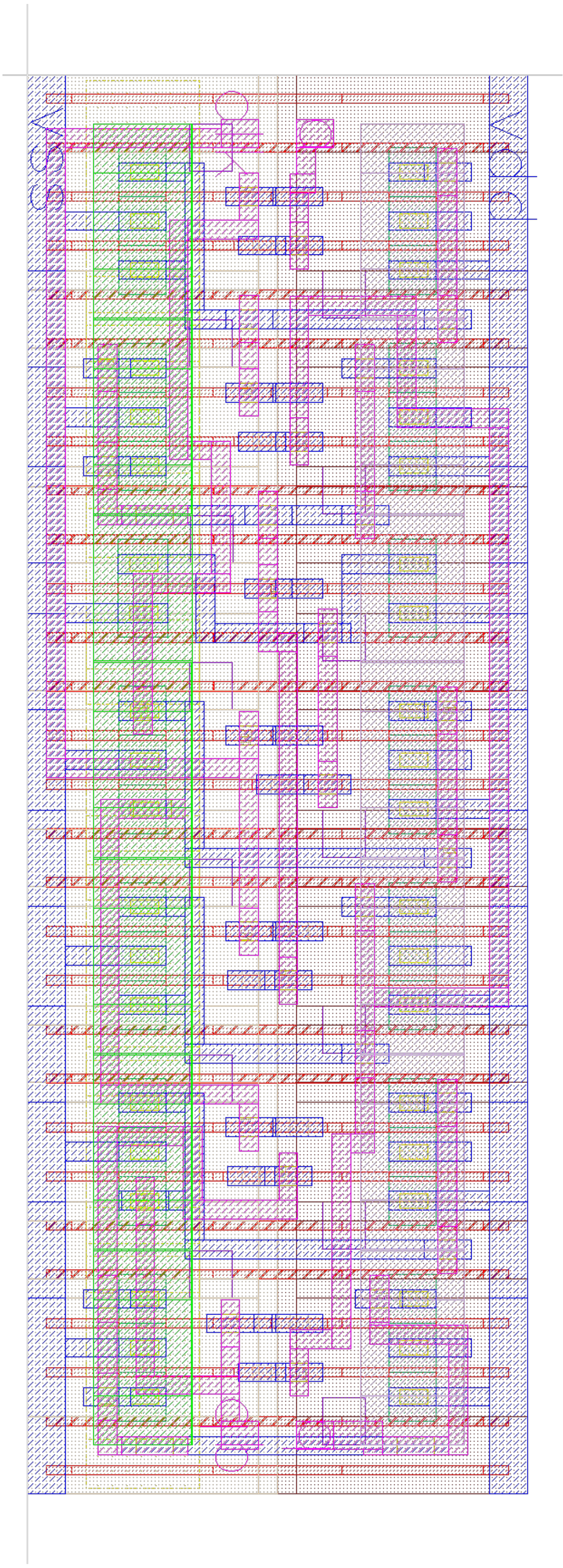


Figure 64: D Flip-Flop: 0 V body-bias, W x H = 2.912 μm x 1.062 μm.

5.6.2 ±2 V body-bias

As with the ***mosfet_low*** logic gates, the ***mosfet_low_bb*** must have a common height for later assembly. The same procedure was done for as for the ***mosfet_low*** gates, resulting in a standard height of 1.328 μm. An increased height was the result of introducing body-bias rails, which are labeled as ***bn*** for nmos body-bias, and ***bp*** for pmos body-bias. The body-bias rails have a width of 40 nm each, with an equal distance from the supply rails. This will lead to a reduced transistor density compared to the 0 V body-bias circuits, but otherwise the structures are mostly identical.

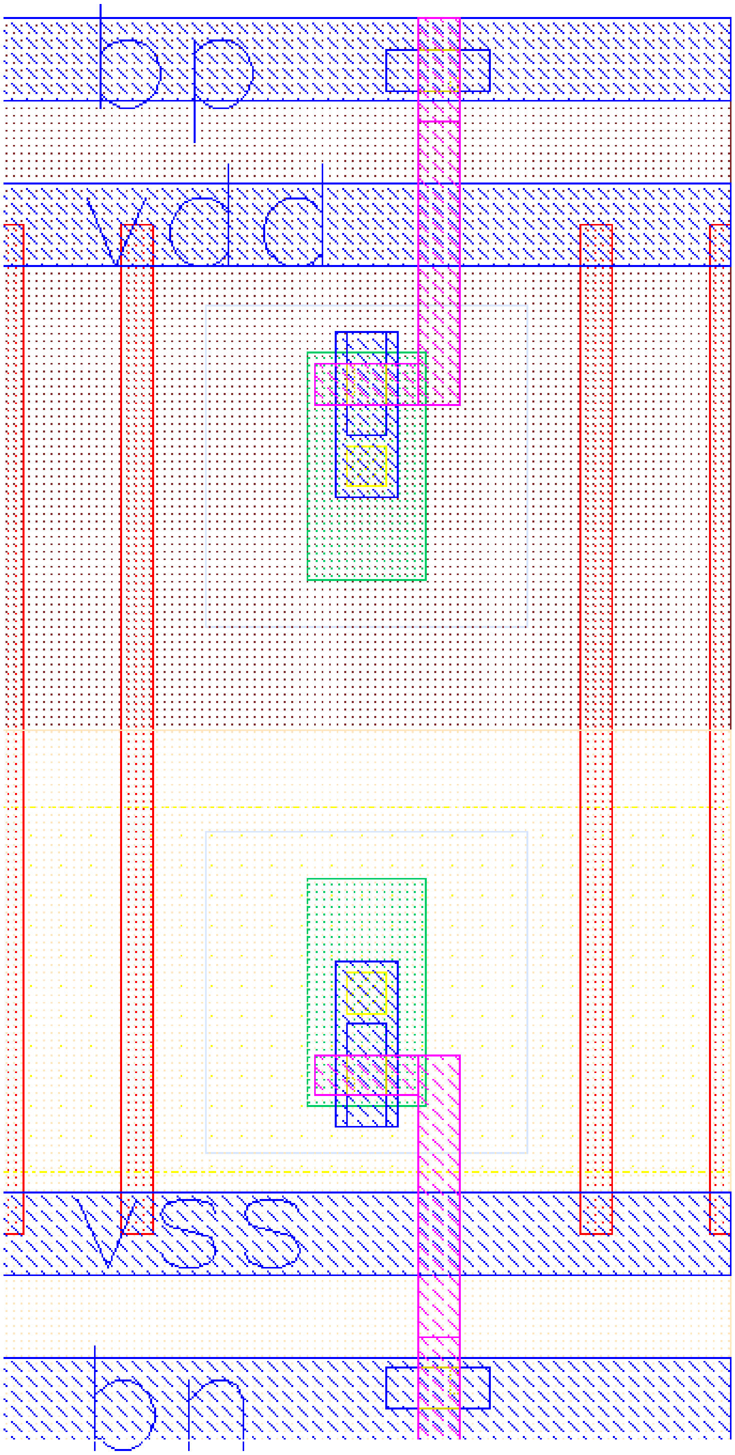


Figure 65: Bulk: 0 V body-bias without body-bias rails.

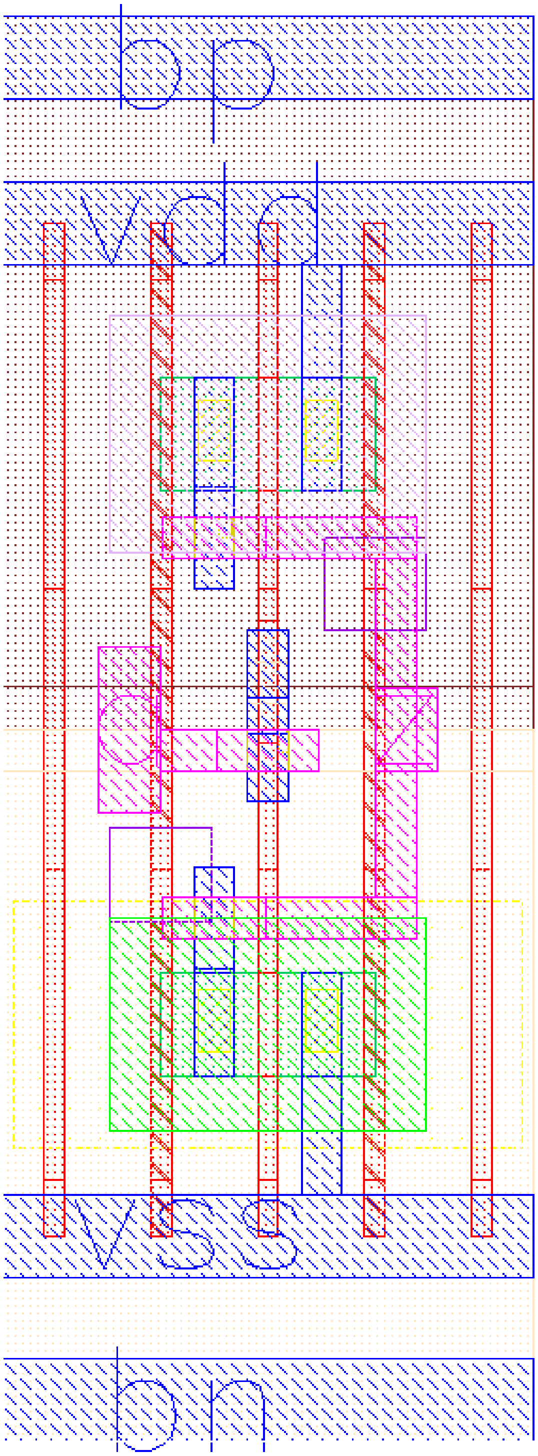


Figure 66: Inverter: 0 V body-bias, W x H = 0.416 μm x 1.382 μm.

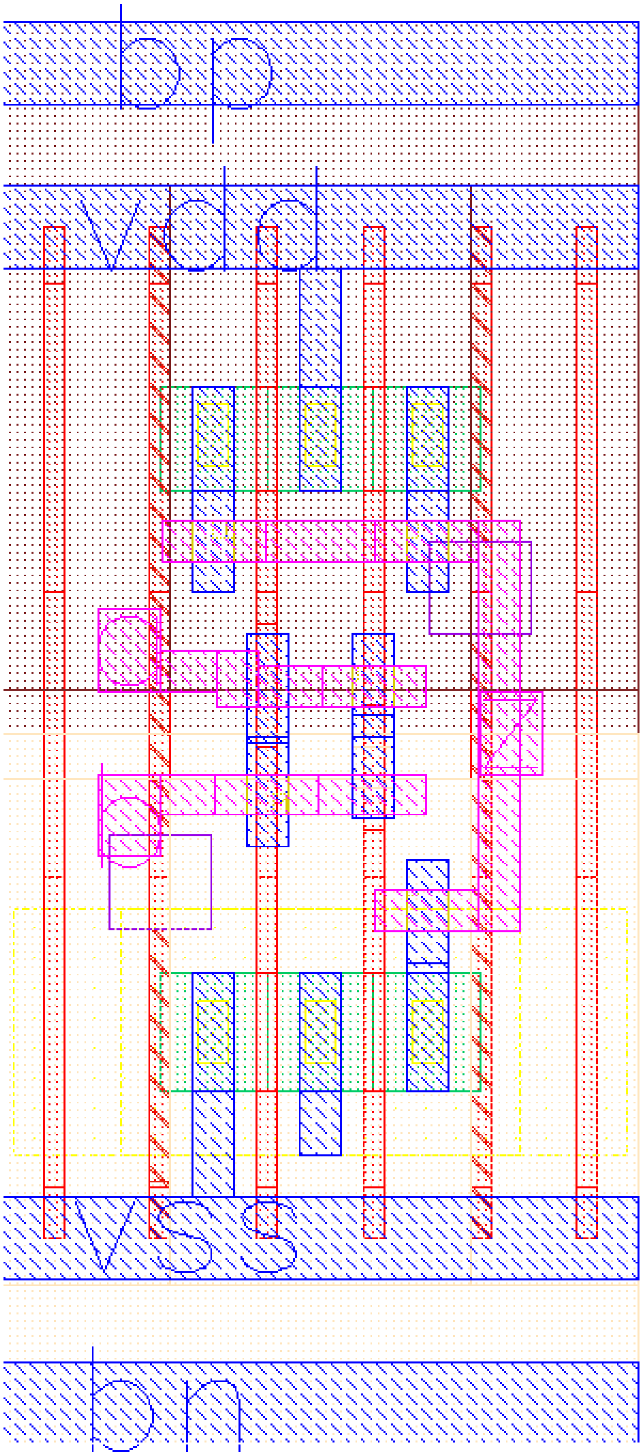


Figure 67: NAND: ±2 V body-bias, W x H = 0.620 μm x 1.382 μm.

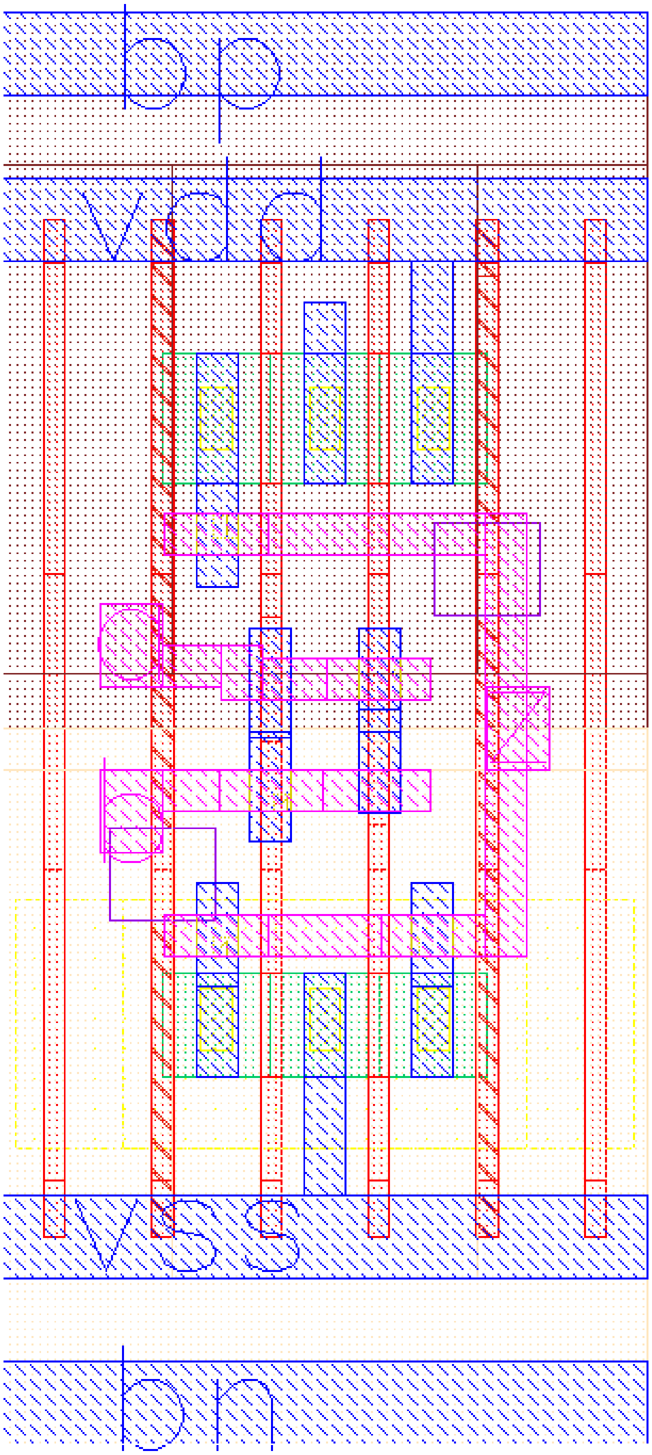


Figure 68: NOR: ±2 V body-bias, W x H = 0.620 μm x 1.382 μm.

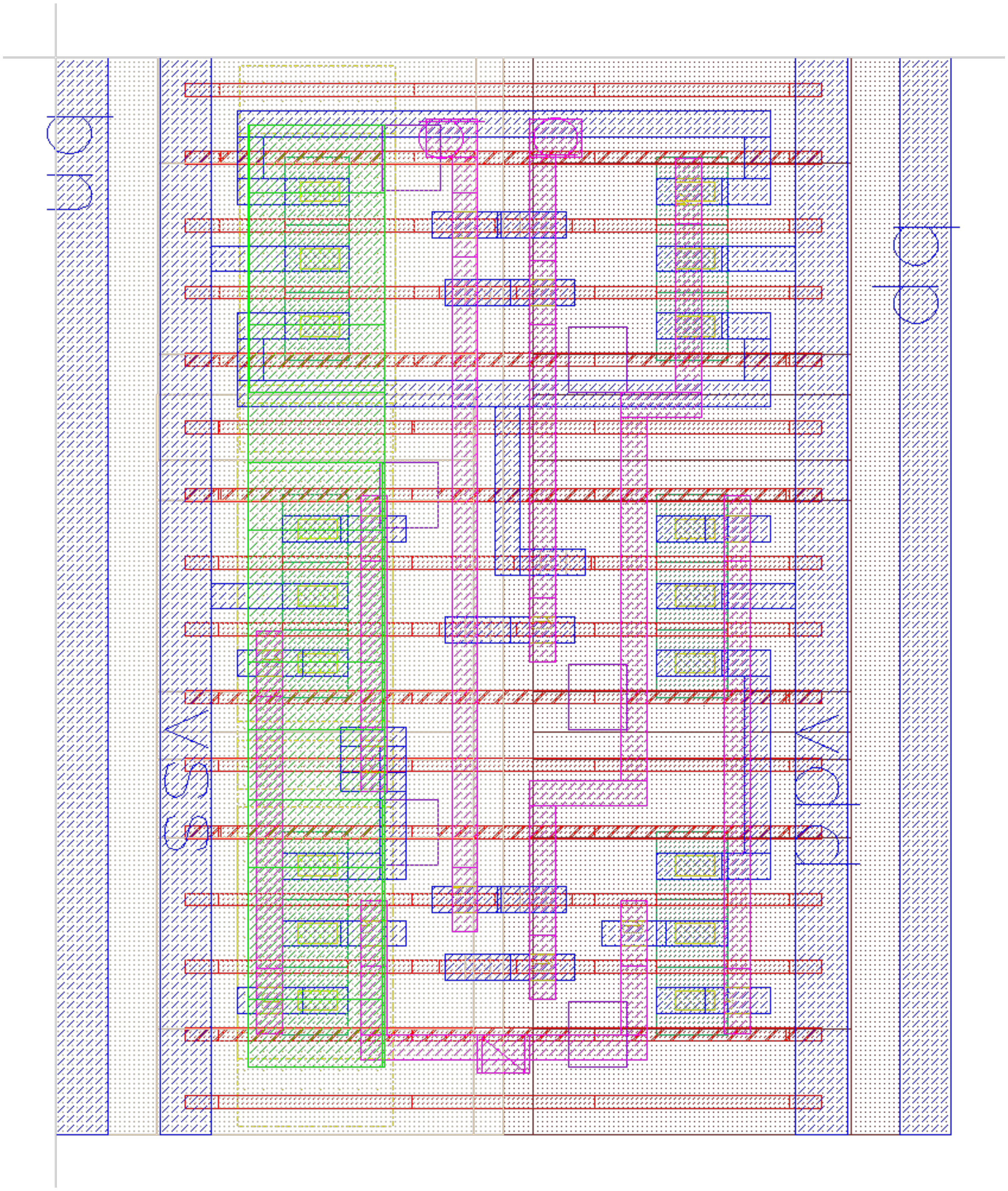


Figure 69: XNOR: ±2 V body-bias, W x H = 1.560 μm x 1.382 μm.

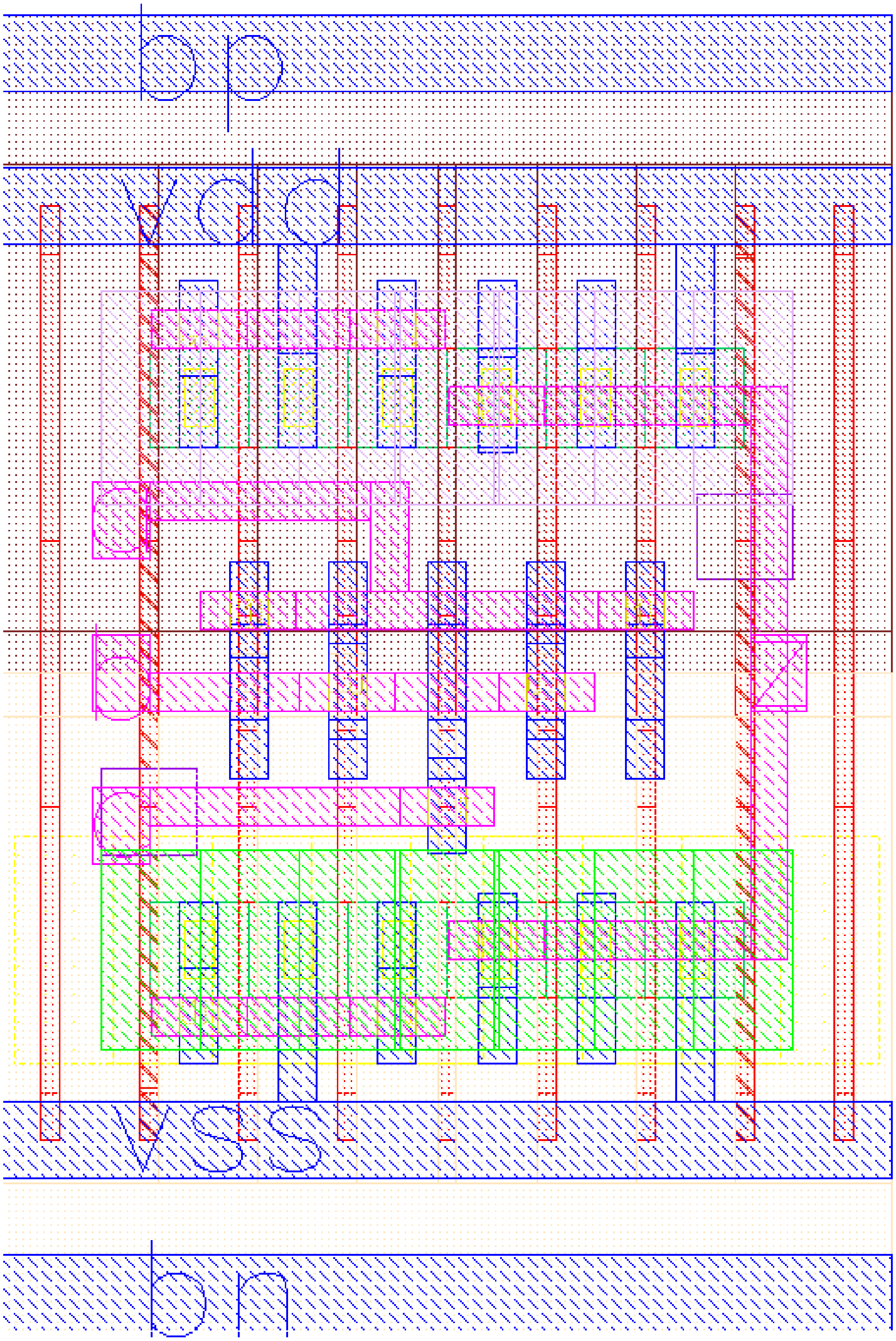


Figure 70: Minority-3: ±2 V body-bias, W x H = 0.832 μm x 1.382 μm.

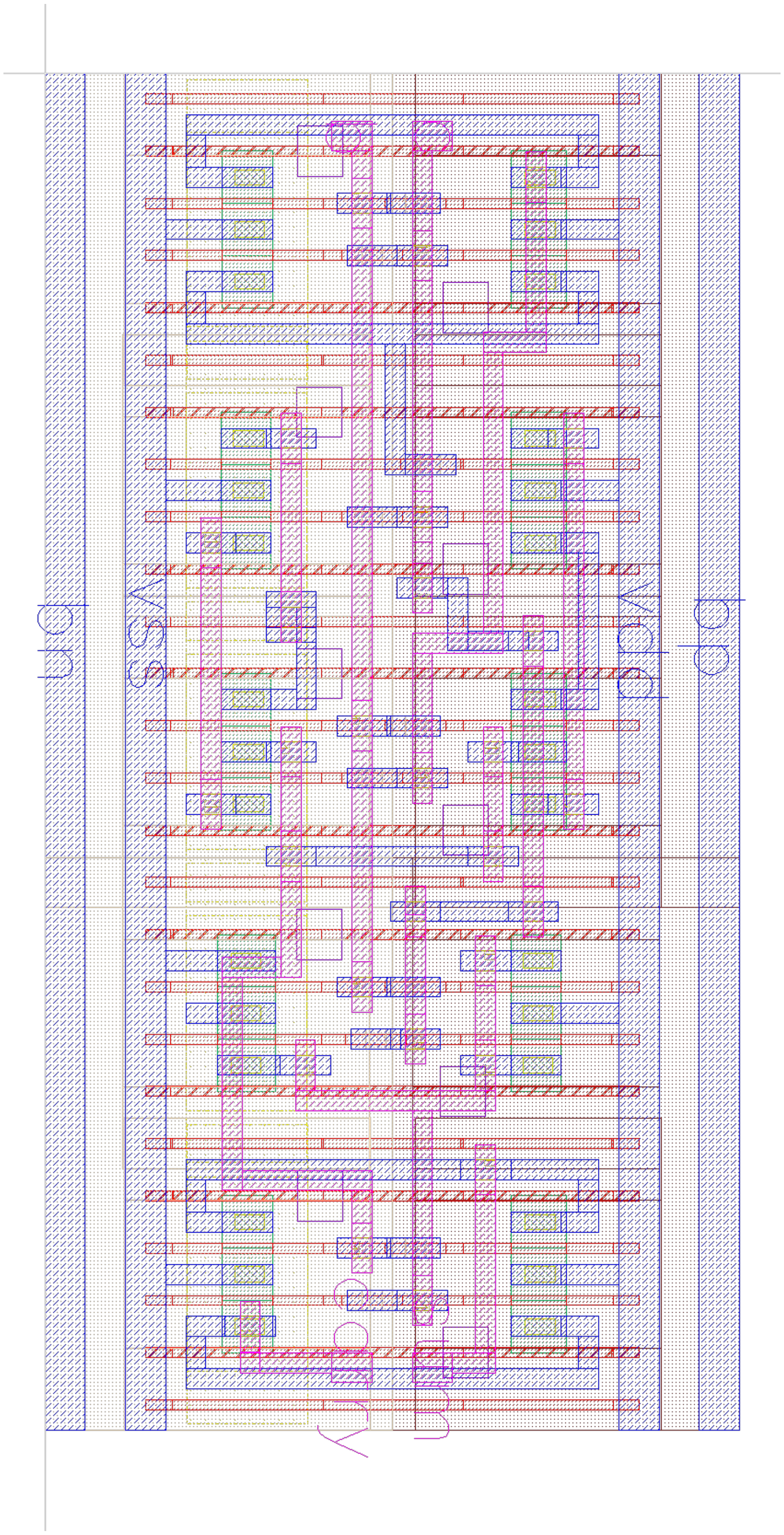


Figure 71: Half-Adder: ±2 V body-bias, W x H = 2.600 μm x 1.382 μm.

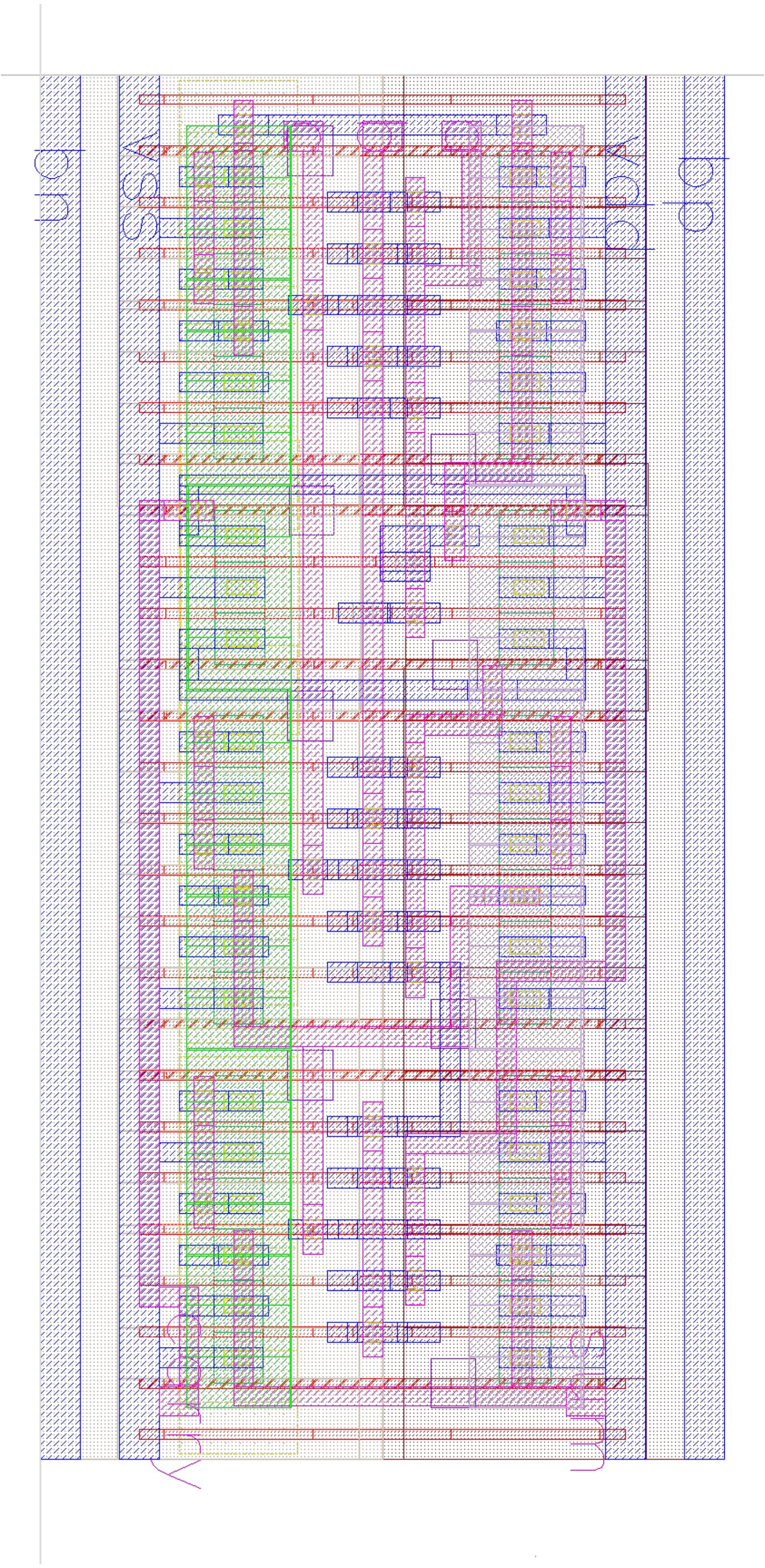


Figure 72: Full-Adder (Min-3): ±2 V body-bias, W x H = 2.704 μm x 1.382 μm.

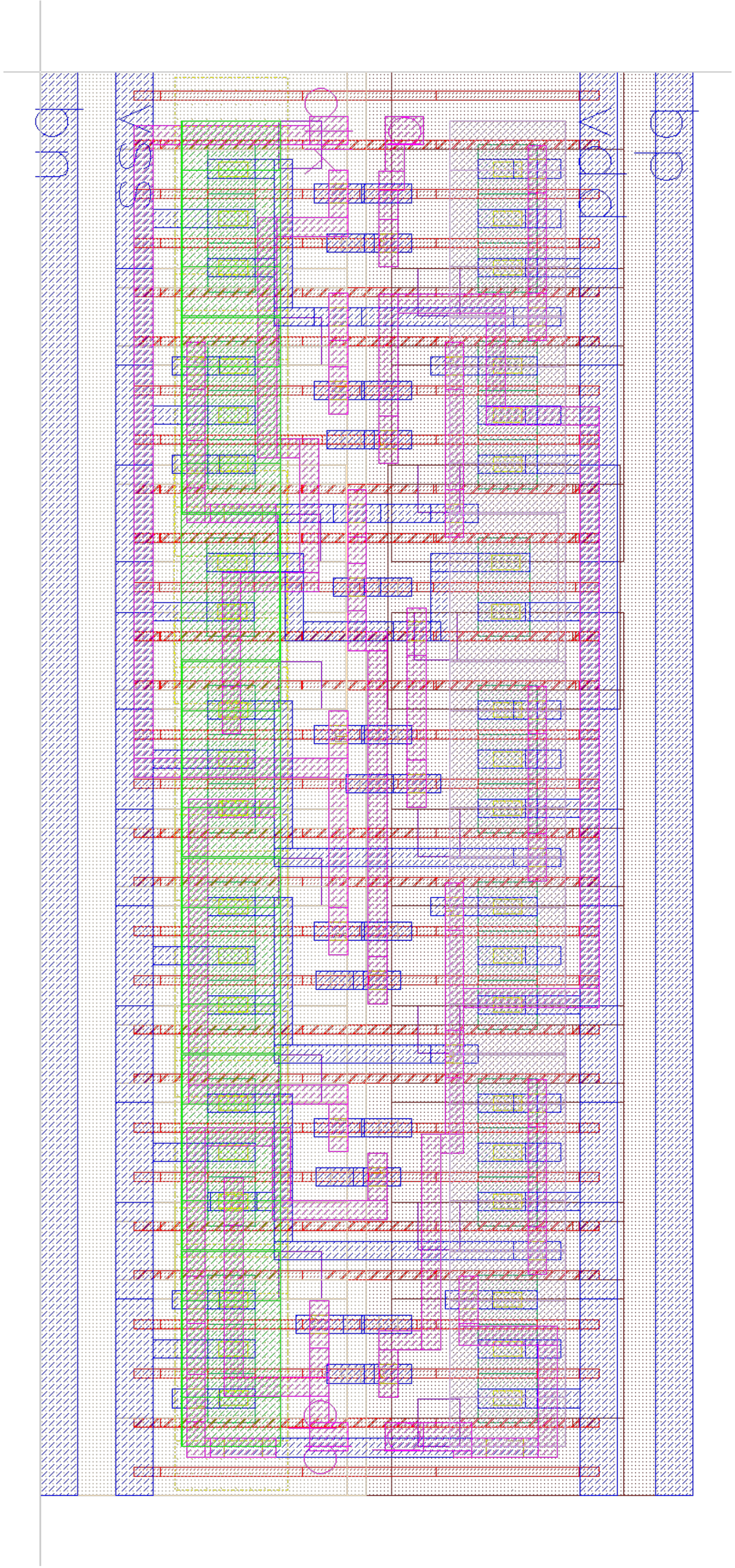


Figure 73: D Flip-Flop: ±2 V body-bias, W x H = 2.912 μm x 1.062 μm.

5.7 Parasitic capacitances

After completing layout design and removing DRC- and LVS errors, parasitic capacitances were extracted and added into the simulations as shown in tables 21, 22 and 23. The additional capacitances lead to a reduced performance compared to regular schematic design, and a degradation factor can be seen in the same tables. However, static power consumption was reduced for all circuits, as seen in table 21. All simulations were done with a temperature of 27 °C.

Table 21: Average static power consumption for all the different logic gates, with worst-case input combination as given by table 19. The table shows average static power consumption for schematics, as well as when including parasitic capacitances. A degradation factor shows how much static power consumption is increased when realising schematics to layout design. Supply voltage was V_{DD}=350 mV for all results, and ***mosfet_low_bb*** gates are ***mosfet_low*** gates with body-bias, as seen in table 18.

Logic gate	Transistor type	Static Power	Layout degradation factor
Inverter	mosfet_low	1.90 nW	0.1
	mosfet_low_bb	3.50 nW	0.01
NAND	mosfet_low	8.64 nW	0.3
	mosfet_low_bb	8.67 nW	0.01
NOR	mosfet_low	4.32 nW	0.3
	mosfet_low_bb	12.29 nW	0.02
XNOR	mosfet_low	14.60 nW	0.3
	mosfet_low_bb	22.93 nW	0.01
XOR	mosfet_low	15.77 nW	0.3
Minority-3	mosfet_low	22.74 nW	0.7
	mosfet_low_bb	27.11 nW	0.04
Half-Adder	mosfet_low	28.12 nW	0.3
	mosfet_low_bb	37.36 nW	0.01
Full-Adder (std.)	mosfet_low	40.92 nW	0.5
Full-Adder (Min-3)	mosfet_low	62.94 nW	0.6
	mosfet_low_bb	72.12 nW	0.02
Full-Adder (NAND-XOR)	mosfet_low	40.95 nW	0.3
Full-Adder (NAND)	mosfet_low	44.70 nW	0.3
Full-Adder (NOR)	mosfet_low	46.32 nW	0.3
D Flip-Flop	mosfet_low	27.28 nW	0.3
	mosfet_low_bb	51.00 nW	0.01

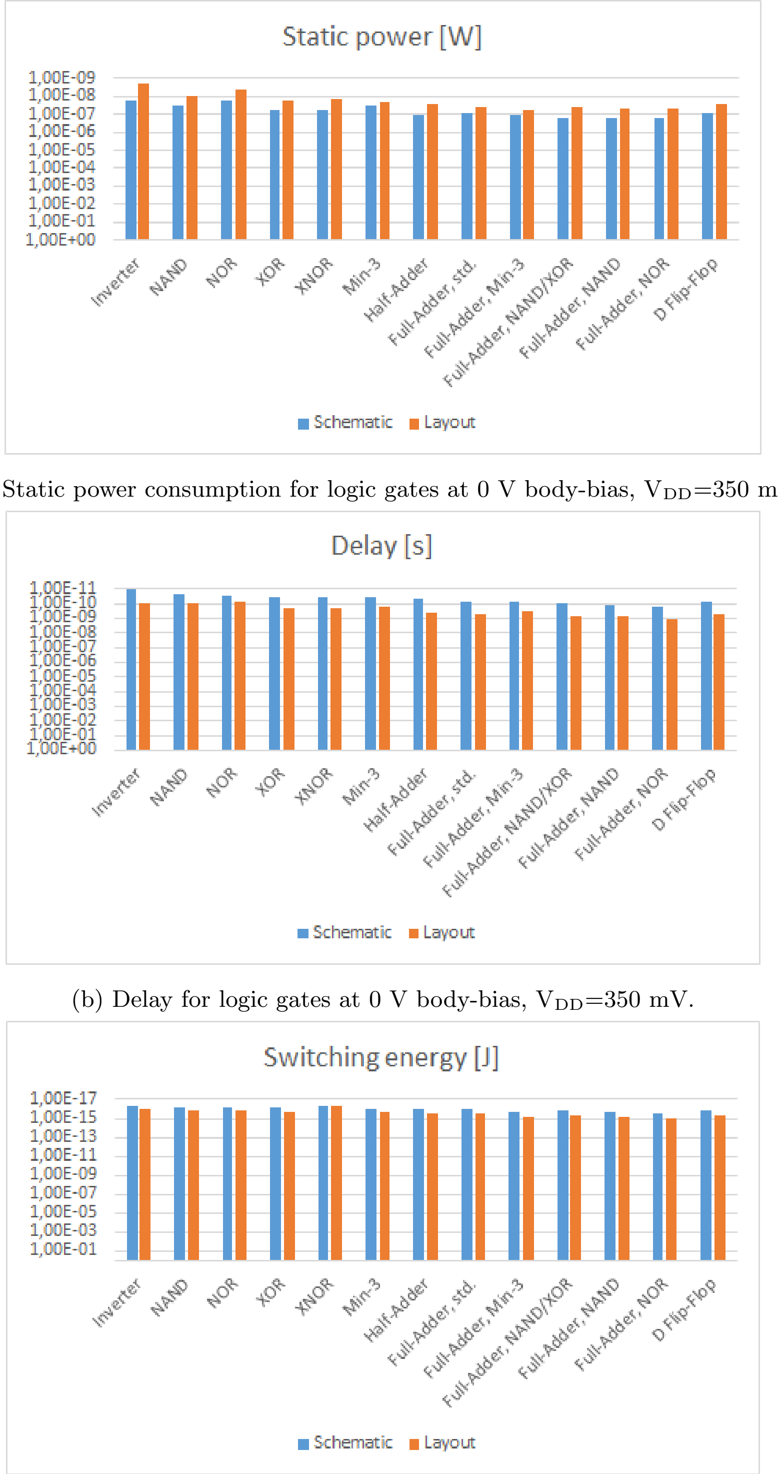
Table 22: Switching delay for the different logic gates, under worst-case conditions. Worst-case conditions are given in table 12, 13 and 14. In addition, parasitic capacitances are extracted from layout design and included in simulations. Supply voltage was V_{DD}=350 mV for all results, and ***mosfet_low_bb*** gates are ***mosfet_low*** gates with body-bias, as seen in table 18

Logic gate	**Transistor type**	**Delay**	**Layout degradation factor**
Inverter	mosfet_low	100.30 ps	9.9
	mosfet_low_bb	62.94 ps	18.8
NAND	mosfet_low	109.29 ps	5.3
	mosfet_low_bb	45.86 ps	5.4
NOR	mosfet_low	81.32 ps	2.9
	mosfet_low_bb	84.35 ps	16.9
XNOR	mosfet_low	221.29 ps	6.0
	mosfet_low_bb	192.16 ps	18.5
XOR	mosfet_low	227.02 ps	6.7
Minority-3	mosfet_low	150.07 ps	4.6
	mosfet_low_bb	162.43 ps	11.6
Half-Adder	mosfet_low	400.55 ps	8.1
	mosfet_low_bb	339.91 ps	19.6
Full-Adder (std.)	mosfet_low	494.82 ps	6.0
Full-Adder (Min-3)	mosfet_low	322.33 ps	3.9
	mosfet_low_bb	269.89 ps	19.2
Full-Adder (NAND-XOR)	mosfet_low	648.21 ps	7.4
Full-Adder (NAND)	mosfet_low	771.92 ps	6.3
Full-Adder (NOR)	mosfet_low	1.067 ns	6.8
D Flip-Flop	mosfet_low	570.93 ps	7.5
	mosfet_low_bb	377.50 ps	16.3

Table 23: Switching energy consumption for the different logic gates, under worst-case conditions. Worst-case conditions are given in table 12, 13 and 14. In addition, parasitic capacitances are extracted from layout design and included in simulations. Supply voltage was V_{DD}=350 mV for all results, and ***mosfet_low_bb*** gates are ***mosfet_low*** gates with body-bias, as seen in table 18

Logic gate	**Transistor type**	**Energy**	**Layout degradation factor**
Inverter	mosfet_low	105.91 aJ	2.4
	mosfet_low_bb	107.96 aJ	0.5
NAND	mosfet_low	139.37 aJ	2.2
	mosfet_low_bb	239.42 aJ	1.7
NOR	mosfet_low	135.19 aJ	2.3
	mosfet_low_bb	145.74 aJ	0.5
XNOR	mosfet_low	53.92 aJ	1.1
	mosfet_low_bb	67.71 aJ	0.1
XOR	mosfet_low	201.30 aJ	3.5
Minority-3	mosfet_low	201.70 aJ	1.9
	mosfet_low_bb	200.41 aJ	0.9
Half-Adder	mosfet_low	346.71 aJ	2.9
	mosfet_low_bb	322.61 aJ	2.8
Full-Adder (std.)	mosfet_low	352.35 aJ	3.0
Full-Adder (Min-3)	mosfet_low	664.06 aJ	2.6
	mosfet_low_bb	375.11 aJ	1.7
Full-Adder (NAND-XOR)	mosfet_low	401.05 aJ	2.9
Full-Adder (NAND)	mosfet_low	692.73 aJ	2.7
Full-Adder (NOR)	mosfet_low	977.16 aJ	2.8
D Flip-Flop	mosfet_low	396.91 aJ	3.0
	mosfet_low_bb	418.27 aJ	1.3

A graphical display of the layout degradation impact can be seen in figure 74 for the ***mosfet_low*** circuits, and figure 75 for the ***mosfet_low_bb*** circuits.

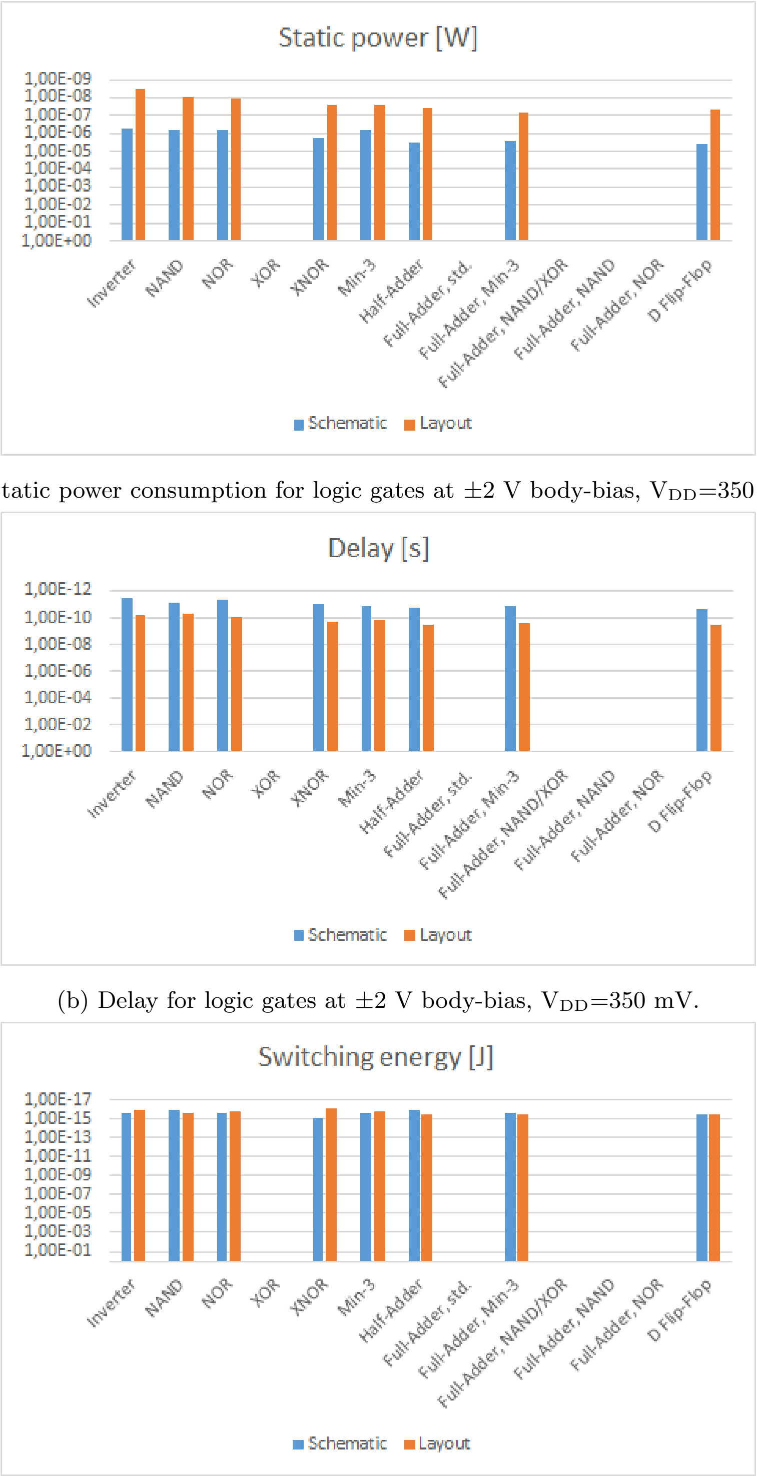


(a) Static power consumption for logic gates at 0 V body-bias, V_{DD}=350 mV.

(b) Delay for logic gates at 0 V body-bias, V_{DD}=350 mV.

(c) Switching energy for logic gates at 0 V body-bias, V_{DD}=350 mV.

Figure 74: Graphical display of the tables 21, 22 and 23, displaying the impact of the layout degradation factor.

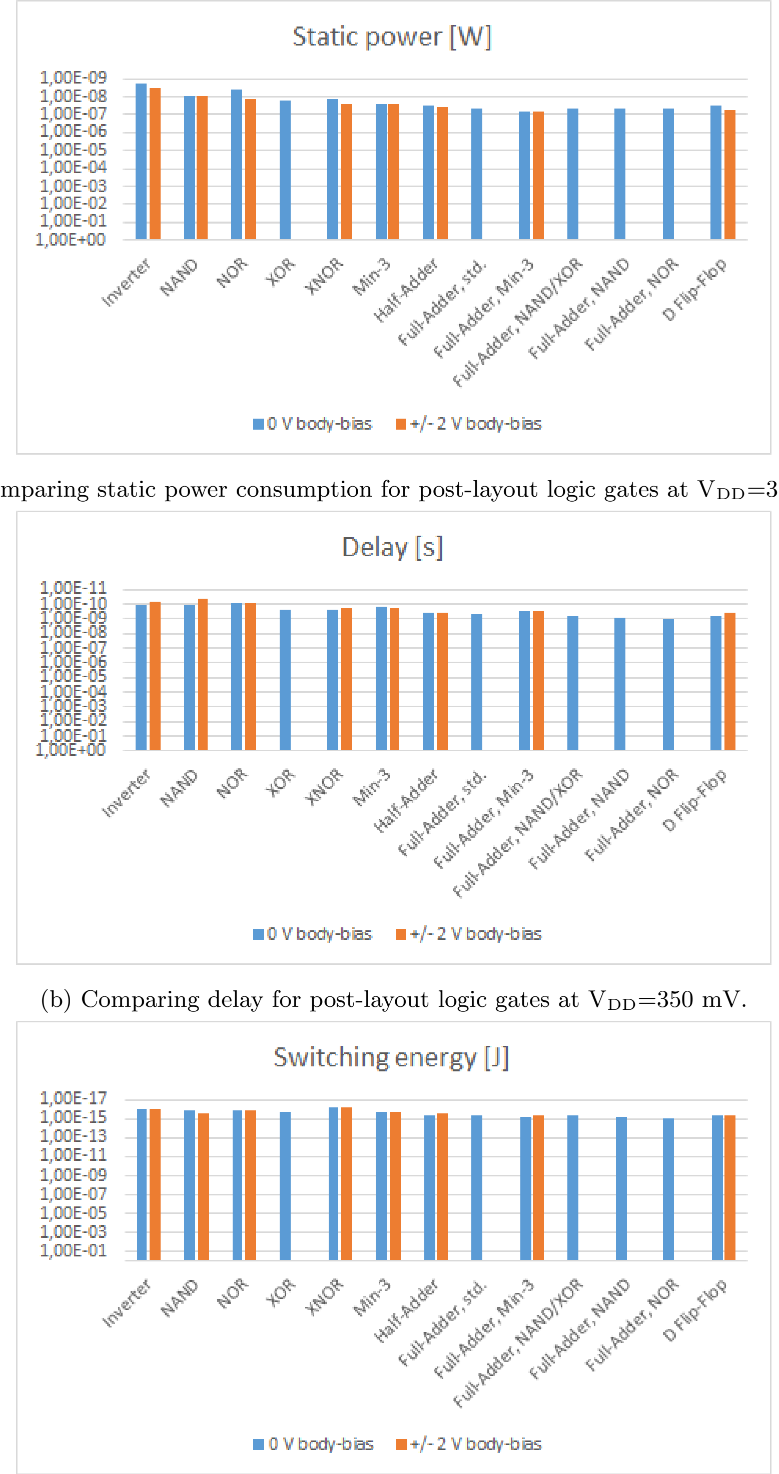


(a) Static power consumption for logic gates at ±2 V body-bias, V_{DD}=350 mV.

(b) Delay for logic gates at ±2 V body-bias, V_{DD}=350 mV.

(c) Switching energy for logic gates at ±2 V body-bias, V_{DD}=350 mV.

Figure 75: Graphical display of the tables 21, 22 and 23, displaying the impact of the layout degradation factor.



(a) Comparing static power consumption for post-layout logic gates at V_{DD}=350 mV.

(b) Comparing delay for post-layout logic gates at V_{DD}=350 mV.

(c) Comparing switching energy for post-layout logic gates at V_{DD}=350 mV.

Figure 76: Graphical display of comparing ***mosfet_low*** with 0 V body-bias and ***mosfet_low_bb*** with ±2 V body-bias transistor circuits after layout design and extracting parasitic capacitances.

5.7.1 Threshold voltage

As seen in table 21, 22 and 23, leakage seems to be reduced when introducing parasitic capacitances, while switching energy and delay increase. This could occur due to a change in the threshold voltage, and the property was therefore mapped again with respect to layout design. The new threshold voltage, compared with the schematic design, can be seen in figure 77

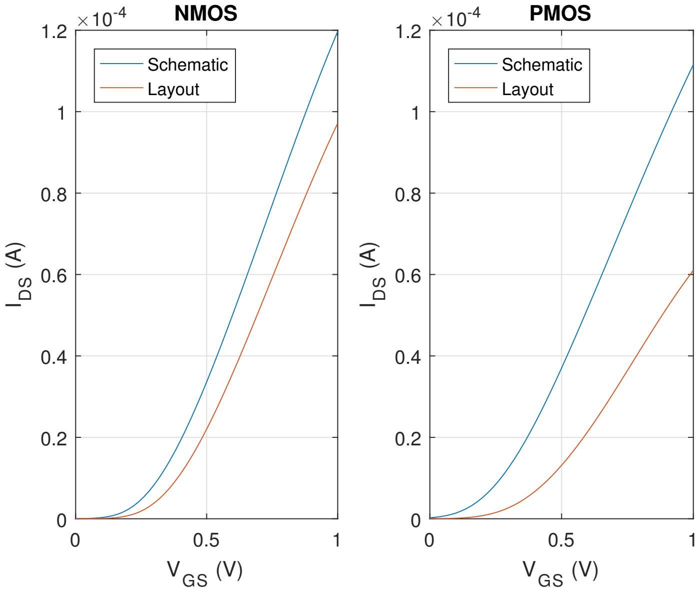


Figure 77: Threshold voltage, V_{th}, extraction by applying the extrapolation method. The current I_{DS} mapped as a function of V_{GS} is the current flowing from drain to source node, and the other way around for the PMOS. W_p=W_n=80 nm, L_p=L_n=20 nm, V_{DD}=1 V and V_{GS} is swept from 0 V to 1 V.

It can be clearly seen in the figure that the threshold voltage has increased, and the new threshold values can be seen in table 24.

Table 24: Threshold voltages for the ***mosfet_low*** transistors, at V_{DD}=1 V with parasitic capacitances included. No body-bias were applied.

	Threshold voltage [mV]	
Transistor	**NMOS**	**PMOS**
mosfet_low	353 mV	391 mV

5.8 Monte Carlo simulation

To ensure a properly function layout design, Monte Carlo simulations were run for all logic gates to ensure that chip manufacturing is feasible for the different designs. As mentioned earlier, subthreshold- and near-subtreshold circuits are significantly more vulnerable to process variation and mismatch, due to the exponential relations. 100 Monte Carlo simulations for mismatch and process variation were performed for all circuits, verifying logic function. All the circuits had a yield of 100% by verifying the logic function.

5.9 16by12-bit Adder

A new 16by12-bit Adder was made, where the initial Half-Adders were replaced by Minority-3 based Full-Adders, as they proved to be faster. The topology of the 16by12-bit Adder however, is the same as shown in figure 24. The Ripple-Carry Adder schematics can be seen in the appendix section, figures 105, 106, 107, 108 and 108. The simulations will now also include layout design and parasitic capacitances, and delay will be measured by the critical path as described earlier.

Table 25: Static power, delay and switching energy simulated by the critical path through a *n*-bit Ripple-Carry adder for different transistor types, with respect to layout design. V_{DD}=350 mV.

		***n*-bit Ripple-Carry Adder**				
Transistor type	**Property**	**12-bit**	**13-bit**	**14-bit**	**15-bit**	**16-bit**
mosfet_low	Static power	801.15 nW	869.40 nW	938.00 nW	1.00 μW	1.07 μW
	Delay	3.87 ns	4.19 ns	4.52 ns	4.84 ns	5.17 ns
	Switching energy	6.07 fJ	6.93 fJ	7.37 fJ	8.31 fJ	9.29 fJ

Table 26: Total static power, delay and switching simulated by the critical path through a 16by12-bit Ripple-Carry adder for different transistor types, with respect to layout design and parasitic capacitances. V_{DD}=350 mV.

Transistor type	**Property**	**16by12-bit Ripple-Carry Adder**
mosfet_low	Static power	26.60 μW
	Delay	22.59 ns
	Switching energy	207.95 fJ

From table 26, a delay of 22.59 ns is observed for the total 16by12-bit adder. A delay of 22.59 ns equals to a speed of 44.26 MHz. Static power and switching energy consumption for the circuit can be seen in the same table, where it was measured through the critical path.

6 Discussion

Two different transistors were initially investigated in this projecty, the ***mosfet_high*** and ***mosfet_low*** transistor. By adopting the ***mosfet_low*** transistor, two libraries were designed where one use a ±2 V forward body-bias. The other library have bulk connections connected to ground, giving a 0 V body-bias. The ***mosfet_high*** transistor library was discontinued during the work, as it did not show satisfying results regarding the project requirements. The cell library was initially optimized for a near-subthreshold operation at V_{DD}=350 mV. However, table 24 shows that threshold voltage increases after layout design. Therefore the cell library is now optimized for subthreshold operation, just below the threshold voltage.

During schematic design and simulation, the ***mosfet_low*** transistor library with a ±2 V body-bias was put on hold as it had some worrying high static power consumption compared to 0 V body-bias. For some circuits, like the NOR-gate seen in table 19, the static power consumption was increased by a factor of 38 when applying a ±2 V forward body-bias. Therefore, the 0 V body-bias library holds a few more logic gates with layout designs. However, the ±2 V body-bias library properties improved significantly after layout design and extracting parasitic capacitances, were leakage currents were reduced considerably by roughly two orders of magnitude, as seen in table 21.

6.1 Requirement specifications

The degradation factor with respect to speed introduced with layout design was far worse than expected, reaching a factor of up to 10 for some 0 V body-bias circuits. While the 16by12-bit Ripple-Carry Adder was designed with a safety margin and reached a speed of 294.1 MHz with regards to the schematics, it could not go faster than 44.26 MHz when layout design and parasitic capacitances were included. However the design passes the requirement specification of 40 MHz, but mismatch and process variation may also, and most likely will, affect this performance. Attention to low power consumption was also emphasized by the requirements, and by looking at table 26, a static power consumption of 26.60 μW and switching energy of 207.95 fJ were simulated for the 16by12-bit adder through the critical path.

Unless more speed is required, there should be no reason to use the ±2 V forward body-bias circuits because of its significantly increased static power consumption.

6.2 Comparing 22 nm FDSOI with 28 nm FDSOI

For subthreshold operation, of the 5 Full-Adders investigated it was the Minority-3 based Full-Adder which yields the best performance with respect to speed. Comparing this Full-Adder with a same Full-Adder in 28 nm FDSOI, both delay and switching energy for the 22 nm FDSOI were reduced by 46 % and 51 %, respectively [29].

The layout design with the highest density of transistors on a given area belongs to the Minority-3 Full-Adder seen in figure 59. 34 transistors on an area of 2.872 μm^2 gives a density of 84.5 nm^2 per transistor on average. As a general thumb of rule, the transistor density should be roughly doubled for every new generation of CMOS technology, according to Moore's law. However, comparing this project's layout with the technology provider's layout designs is not fully reasonable, as they may apply more aggressive design rules and achieve an even higher transistor density. But by comparison, a Minority-3 based Full-Adder in [37] by 28 nm FDSOI technology was designed with a layout area of 12.06 μm^2, resulting in a density of 354.7 nm^2 per transistor. Another Minority-3 based Full-Adder in [29] by the same 28 nm FDSOI technology, have a significantly higher transistor density of 155.9 nm^2 per transistor. The Minority-3 based Full-Adder in this report does not manage to double the transistor density, but does increase the density by 45.8 %.

Deciding whether to use a 28 nm or 22 nm FDSOI technology should not only be considered against layout area, performance and static power consumption, but also with respect to production costs were the 22 nm FDSOI surely will be more expensive.

6.3 Layout considerations

From table 21, 22 and 23 it is seen that by extracting parasitic capacitances from layout design, the performance is severely affected. Especially the ±2 V body-bias circuits were highly impacted, as seen in figure 75.

While delay and energy consumption increases with a factor of roughly 5 to 10 times for delay and 2 to 3 times for switching energy, static power consumption is decreased by approximately 3 to 10 times for the ***mosfet_low*** with 0 V body-bias.

For the ±2 V body-bias circuits, ***mosfet_low_bb***, static power was averagely reduced by as much as 2 orders of magnitude. Switching energy consumption was not changed much, and was sometimes even reduced. However, the switching energy ranges from decreasing by an order of magnitude, to increase by up to 2.8 times. However these results seems somewhat inconsistent and should be considered so. Delay however, was increased dramatically by up to 20 times, and thus removing much of the advantage by applying forward body-bias.

The increased switching energy and delay, in combination with a decreased static power consumption may indicate an increased threshold voltage of the transistors. A new simulation of the threshold voltage, where layout design and parasitic capacitances were extracted, was performed and can be seen in figure 77, which confirms an increased threshold voltage of 14.6 % for the NMOS transistor, and 32.5 % for the PMOS.

It can also be seen from tables 21, 22 and 23, that the layout degradation factor seems to increase more for the larger and more complex logic gates, though with some exceptions. This is most likely the result of increased routing between transistors, and the number of via-connections between different metal layers. Therefore, a shortest possible path with few via-connections should be emphasized when routing between transistors in layout design.

6.4 Body-bias considerations

Before layout design and extracting parasitic capacitances, ±2 V body-bias were found to enhance speed by up to 5 times, while increasing switching energy by roughly a factor of 2 and up to 6. However, static power consumption were found to dramatically increase by a factor of up to 38. Though after layout design and parasitic capacitances were taken into account, ±2 V body-bias would in best case double the speed, or have the same as 0 V body-bias. Switching energy would also be mostly the same, or slightly increased, while static power consumption was generally reduced by 2 orders of magnitude. The results for the body-bias circuits seen in figure 75 also seem a little inconsistent compared to 0 V body-bias circuits in figure 74. The cell libraries with 0 V body-bias and ±2 V body-bias should therefore be applied with regards to the circuits' activity factor. Layout area requirements should also be considered, as the ±2 V body-bias circuits occupies a 23.15 % larger layout area, compared to the 0 V body-bias circuits.

6.5 Further development

For completely realistic results regarding performance and reliability, the design should be realised to a physical chip where process variation, mismatch and temperature will certainly have significant impact on performance. Optimizing the circuits for an even lower supply voltage should be possible, though the safety margin for the 0 V body-bias circuits are limited.

Layout design may also be optimized with respect to area and performance by someone with more experience. Some of the more complex layout designs were assembled by the basic logic gates, and then wired together. This will result in a significant amount of wasted area for larger circuits, and in addition more routing and increased parasitic capacitances will be introduced to the circuits and thus reduce performance. Increased chip manufacturing costs as a result of this should therefore be taken into account.

It should also be noted that the direction of the drain-to-source current, $\mathrm{I_{DS}}$, does not follow one common direction in the layout designs for this report. This may introduce varying current mobility, μ_n, in the different transistor channels, and thus affect performance and mismatch [44]. However, this is beyond the scope of this thesis, and should rather be considered for further development. Taking care of this issue would also result in a larger layout area and increase parasitic capacitances.

7 Conclusion

Presented in this report is a cell library of the most common basic logic gates, given by a commercial available 22 nm FDSOI CMOS technology. The basic logic gates are assembled into more complex gates, like D Flip-Flops, Half-, Full- and Ripple-Carry Adders, and should be sufficient to create most digital logic circuits. Several versions of the Full-Adder were designed and simulated, where the Minority-3 based Full-Adder proved to be most effective for subthreshold operation.

The cell library is optimized for a subthreshold operation at a supply voltage of V_{DD}=350 mV, right below the transistors' threshold voltage. The cell library may either use a 0 V body-bias, or a ±2 V forward body-bias, and has layout design for every logic gate. All the circuits are appended with simulation results of static power consumption, delay and switching energy, where layout design and parasitic capacitances are included. However, the 0 V body-bias and ±2 V body-bias circuits can not be combined in the same design due to layout design rules.

For the project requirements where a minimum speed of 40 MHz was required for a 16by12-bit adder, the 0 V body-bias circuits were found to be the most suitable approach to meet the project requirements, with emphasis on lowest possible power consumption. The 16by12-bit adder designed in this report reaches a speed of 44.26 MHz at V_{DD}=350 mV. In addition, the static power and switching energy consumption were simulated to 26.60 μW and 207.95 fJ, respectively.

Compared to a similar 28 nm FDSOI technology [29], the transistor density in this report was increased by 45.8 % for the Minority-3 based Full-Adder. The energy consumption of a 22 nm FDSOI 16-bit adder was reduced by 65.6 % compared to a 16-bit adder in 28 nm FDSOI, at VDD=350 mV [37]. The same 16-bit adder requires an area of 209.28 μm^2, while the 16-bit adder in this report only requires an area of 42.71 μm^2, thus reducing the layout area by 79.6 %.

References

[1] S. Sun, V. K. Narayana, T. El-Ghazawi, and V. J. Sorger. Chasing moore's law with clear. pages 1–2, May 2017.

[2] Lanny L Lewyn, Trond Ytterdal, Carsten Wulff, and Kenneth Martin. Analog circuit design in nanoscale cmos technologies. *Proceedings of the IEEE*, 97(10):1687–1714, 2009.

[3] Xiaoning Qi, Sam C Lo, Alex Gyure, Yansheng Luo, Mahmoud Shahram, Kishore Singhal, and Don B MacMillen. Efficient subthreshold leakage current optimization-leakage current optimization and layout migration for 90-and 65-nm asic libraries. *IEEE Circuits and Devices Magazine*, 22(5):39–47, 2006.

[4] R. Robertazzi, M. Scheurman, M. Wordeman, S. Tian, and C. Tyberg. Analytical test of 3d integrated circuits. pages 1–10, Oct 2017.

[5] V. S. Melikyan and A. G. Harutyunyan. 3d integrated circuits multifactor placement. pages 1–4, Sept 2017.

[6] Sherif M. Sharroush. Analysis of the subthreshold cmos logic inverter. *Ain Shams Engineering Journal*, 2016.

[7] Massimo Alioto. Ultra-low power vlsi circuit design demystified and explained: A tutorial. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 59(1):3–29, 2012.

[8] Gerhard Schrom, D Liu, Ch Pichler, Ch Svensson, and Siegfried Selberherr. Analysis of ultra-low-power cmos with process and device simulation. In *Solid State Device Research Conference, 1994. ESSDERC'94. 24th European*, pages 679–682. IEEE, 1994.

[9] G Schrom and Siegfried Selberherr. Ultra-low-power cmos technologies. 1:237–246, 1996.

[10] Lars-Frode Schjolden. Low energy implementation of robust digital arithmetic in sub/near-threshold nanoscale cmos. 2013.

[11] M. J. Riezenman. Wanlass's cmos circuit. *IEEE Spectrum*, 28(5):44–, May 1991.

[12] Eric A Vittoz. Micropower techniques, 1994.

[13] Neil H. E. Weste and David Money Harris. *Integrated Circuit Design.* Pearson, Reading, Massachusetts, 1993.

[14] CGBt Garrett and W He Brattain. Physical theory of semiconductor surfaces. *Physical Review*, 99(2):376, 1955.

[15] F. Wanlass and C. Sah. Nanowatt logic using field-effect metal-oxide semiconductor triodes. In *1963 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, volume VI, pages 32–33, Feb 1963.

[16] Yannis Tsividis. Eric vittoz and the strong impact of weak inversion circuits. *IEEE Solid-State Circuits Society Newsletter*, 13(3):56–58, 2008.

[17] Alice Wang, Benton H Calhoun, and Anantha P Chandrakasan. *Sub-threshold design for ultra low-power systems*, volume 95. Springer, 2006.

[18] David J Comer and Donald T Comer. Operation of analog mos circuits in the weak or moderate inversion region. *IEEE Transactions on Education*, 47(4):430–435, 2004.

[19] Ameet Chavan, Gaurav Dukle, Ben Graniello, and Eric MacDonald. Robust ultra-low power subthreshold logic flip-flop design for reconfigurable architectures. In *Reconfigurable Computing and fpga'S, 2006. ReConFig 2006. IEEE International Conference on*, pages 1–7. IEEE, 2006.

[20] Ayhan A Mutlu, Norman G Gunther, and Mahmud Rahman. Analysis of two-dimensional effects on subthreshold current in submicron mos transistors. *Solid-State Electronics*, 46(8):1133–1137, 2002.

[21] Eric Vittoz and Jean Fellrath. Cmos analog integrated circuits based on weak inversion operations. *IEEE journal of solid-state circuits*, 12(3):224–231, 1977.

[22] Eric A. Vittoz. Origins of weak inversion (or sub-threshold) circuit design. *Sub-threshold Design for Ultra Low-Power Systems*, pages 7–9, 2006.

[23] Tony Chan Carusone, David Johns, and Kenneth Martin. *Analog Integrated Circuit Design*. Wiley, Reading, Massachusetts, 1993.

[24] Neil HE West and Kamran Eshraghian. Principles of cmos vlsi design. *Reading, MA: Addion-Wesley, ch8*, 1993.

[25] Preeti Ranjan Panda, BVN Silpa, Aviral Shrivastava, and Krishnaiah Gummidipudi. *Power-efficient system design*. Springer Science & Business Media, 2010.

[26] Ams design with globalfoundries 22fdx™ technology. `https://www.globalfoundries.com/sites/default/files/articles/ams-design-with-globalfoundries-22fdx-technology.pdf`.

[27] Hourieh Attarzadeh, Snorre Aunet, and Trond Ytterdal. An ultra-low-power/high-speed 9-bit adder design: Analysis and comparison vs. technology from 130nm-lp to utbb fd-soi-28nm. In *Nordic Circuits and Systems Conference (NORCAS): NORCHIP & International Symposium on System-on-Chip (SoC), 2015*, pages 1–4. IEEE, 2015.

[28] Jonathan Edvard Bjerkedok. Subthreshold real-time counter. 2013.

[29] Aslak Lykre Holen. Implementation and comparison of digital arithmetics for low voltage/low energy operation. 2015.

[30] M Muker and M Shams. Designing digital subthreshold cmos circuits using parallel transistor stacks. *Electronics letters*, 47(6):372–374, 2011.

[31] Ali Asghar Vatanjou, Trond Ytterdal, and Snorre Aunet. 4 sub-/near-threshold flip-flops with application to frequency dividers. pages 1–4, 2015.

[32] Snorre Aunet and Yngvar Berg. Three sub-fj power-delay-product subthreshold cmos gates. *IFIP VLSI SoC, Perth, Australia*, 1719, 2005.

[33] Hans Kristian Otnes Berge and Snorre Aunet. Multi-objective optimization of minority-3 functions for ultra-low voltage supplies. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pages 2313–2316. IEEE, 2011.

[34] Hans Kristian Otnes Berge, Amir Hasanbegović, and Snorre Aunet. Muller c-elements based on minority-3 functions for ultra low voltage supplies. In *Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2011 IEEE 14th International Symposium on*, pages 195–200. IEEE, 2011.

[35] K. Kato, Y. Takahashi, and T. Sekine. Two phase clocking subthreshold adiabatic logic. pages 598–601, June 2014.

[36] K. Granhaug and S. Aunet. Six subthreshold full adder cells characterized in 90 nm cmos technology. pages 25–30, April 2006.

[37] Ali Asghar Vatanjou, Even Låte, Trond Ytterdal, and Snorre Aunet. Ultra-low voltage adders in 28 nm fdsoi exploring poly-biasing for device sizing. *Nordic Circuits and Systems Conference (NORCAS), 2016 IEEE*, 2016.

[38] Hiroki Iwamura, Masamichi Akazawa, and Yoshihito Amemiya. Single-electron majority logic circuits. *IEICE Transactions on Electronics*, 81(1):42–48, 1998.

[39] Snorre Aunet. Real-time reconfigurable devices implemented in uv-light programmable floating-gate cmos. 2002.

[40] Snorre Aunet. On the reliability of ultra low voltage circuits built from minority-3 gates. In *Circuit Theory and Design (ECCTD), 2011 20th European Conference on*, pages 540–543. IEEE, 2011.

[41] Stian Østerhus. Subthreshold cmos cell library by 22 nm fdsoi technology. 2017.

[42] Adelmo Ortiz-Conde, FJ Garcıa Sánchez, Juin J Liou, Antonio Cerdeira, Magali Estrada, and Y Yue. A review of recent mosfet threshold voltage extraction methods. *Microelectronics Reliability*, 42(4-5):583–596, 2002.

[43] Tarek K. Darwish Ahmed M. Shams and Magdy A. Bayoumi. Performance analysis of low-power 1-bit cmos full adder cells. *IEEE transactions on very large scale integration (VLSI) systems*, 10(1):20–29, 2002.

[44] J Oh, S-H Lee, K-S Min, J Huang, BG Min, B Sassman, K Jeon, W-Y Loh, J Barnett, I Ok, et al. Sige cmos on (110) channel orientation with mobility boosters: Surface orientation, channel directions, and uniaxial strain. In *VLSI Technology (VLSIT), 2010 Symposium on*, pages 39–40. IEEE, 2010.

Appendix

Schematics

Inverter

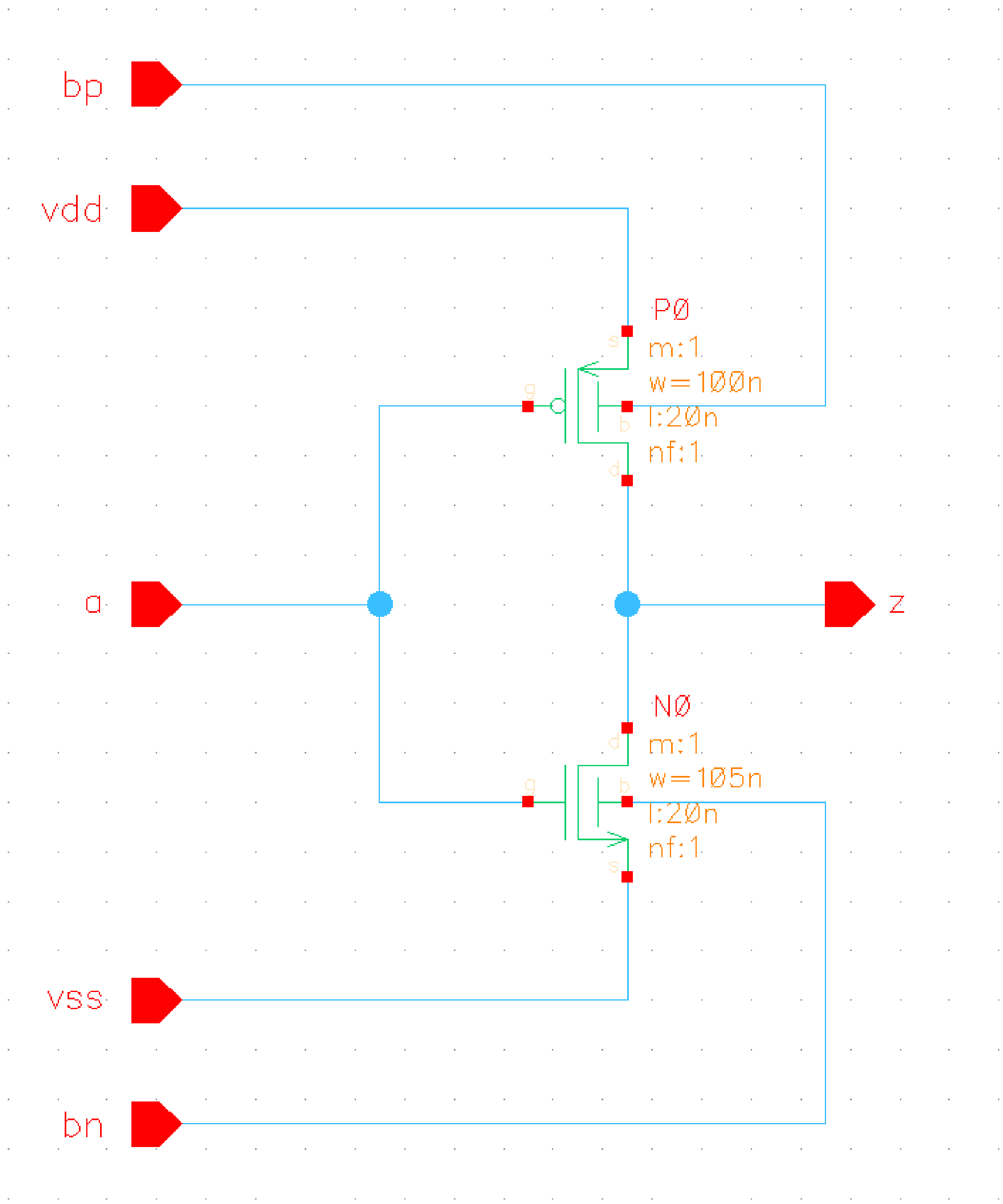


Figure 78: Schematic design of an inverter by ***mosfet_high*** transistors, optimized for V_{DD}=350 mV according to table 18.

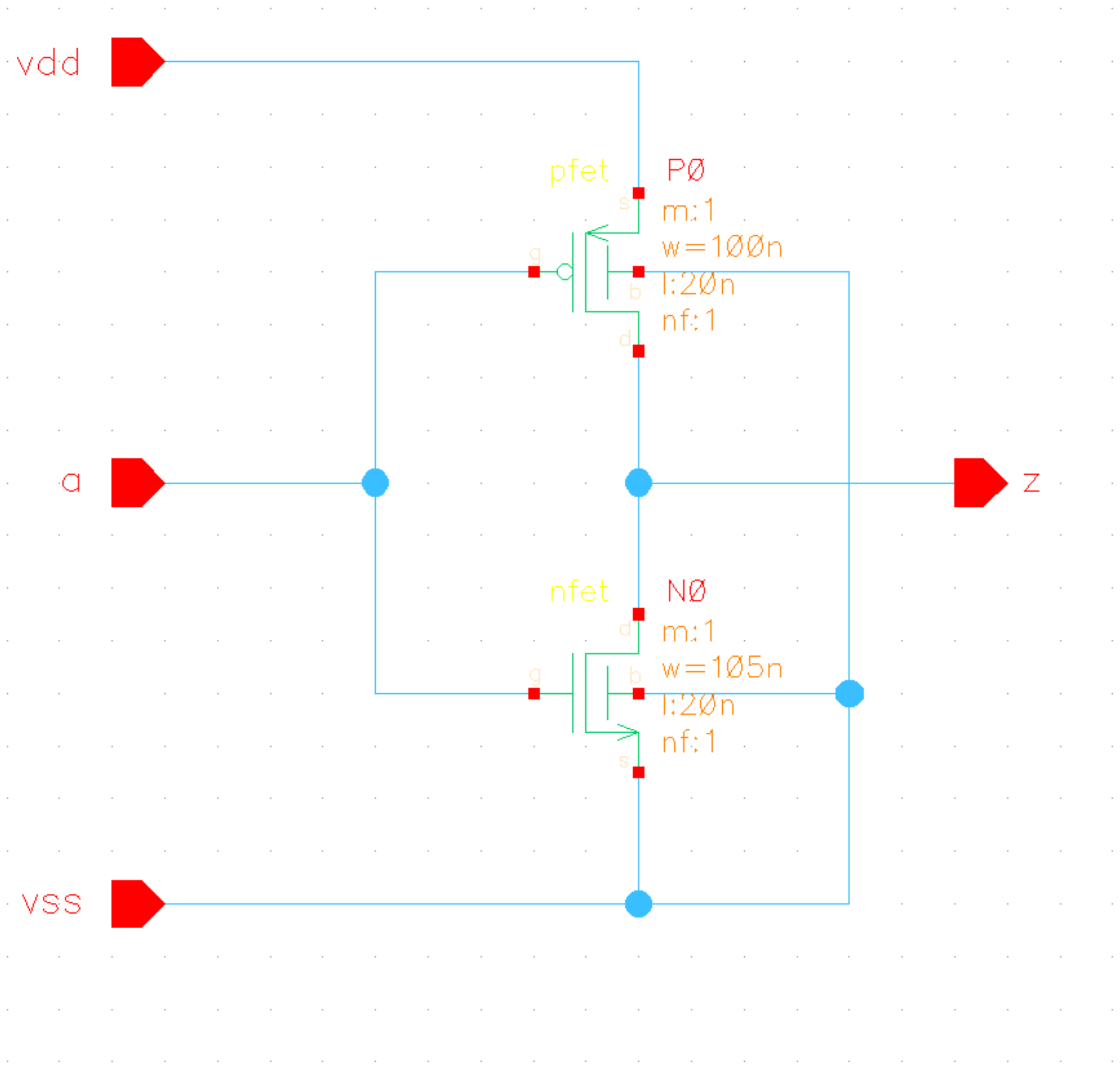


Figure 79: Schematic design of an inverter by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with no body-bias according to table 18.

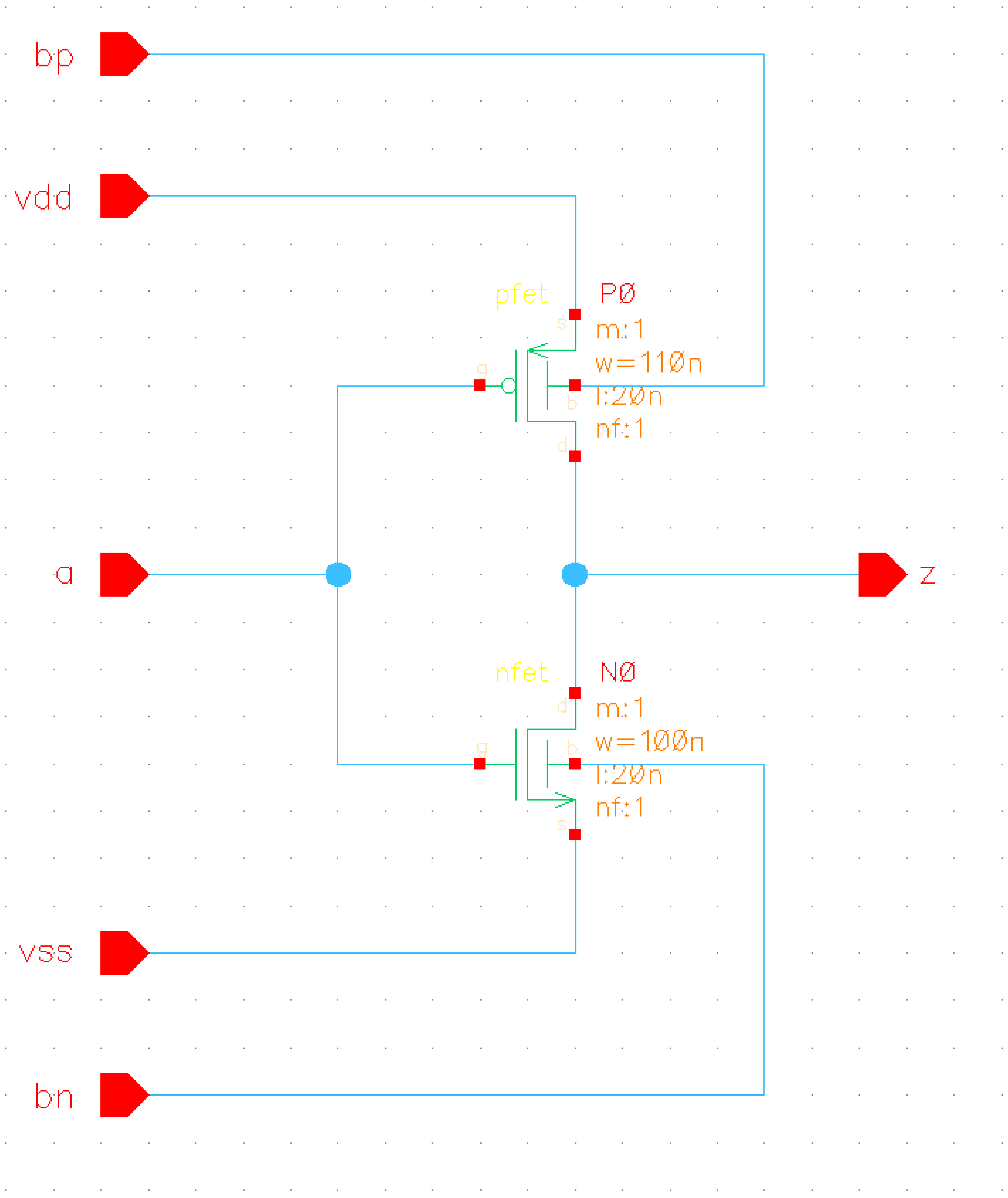


Figure 80: Schematic design of an inverter by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with body-bias according to table 18.

NAND

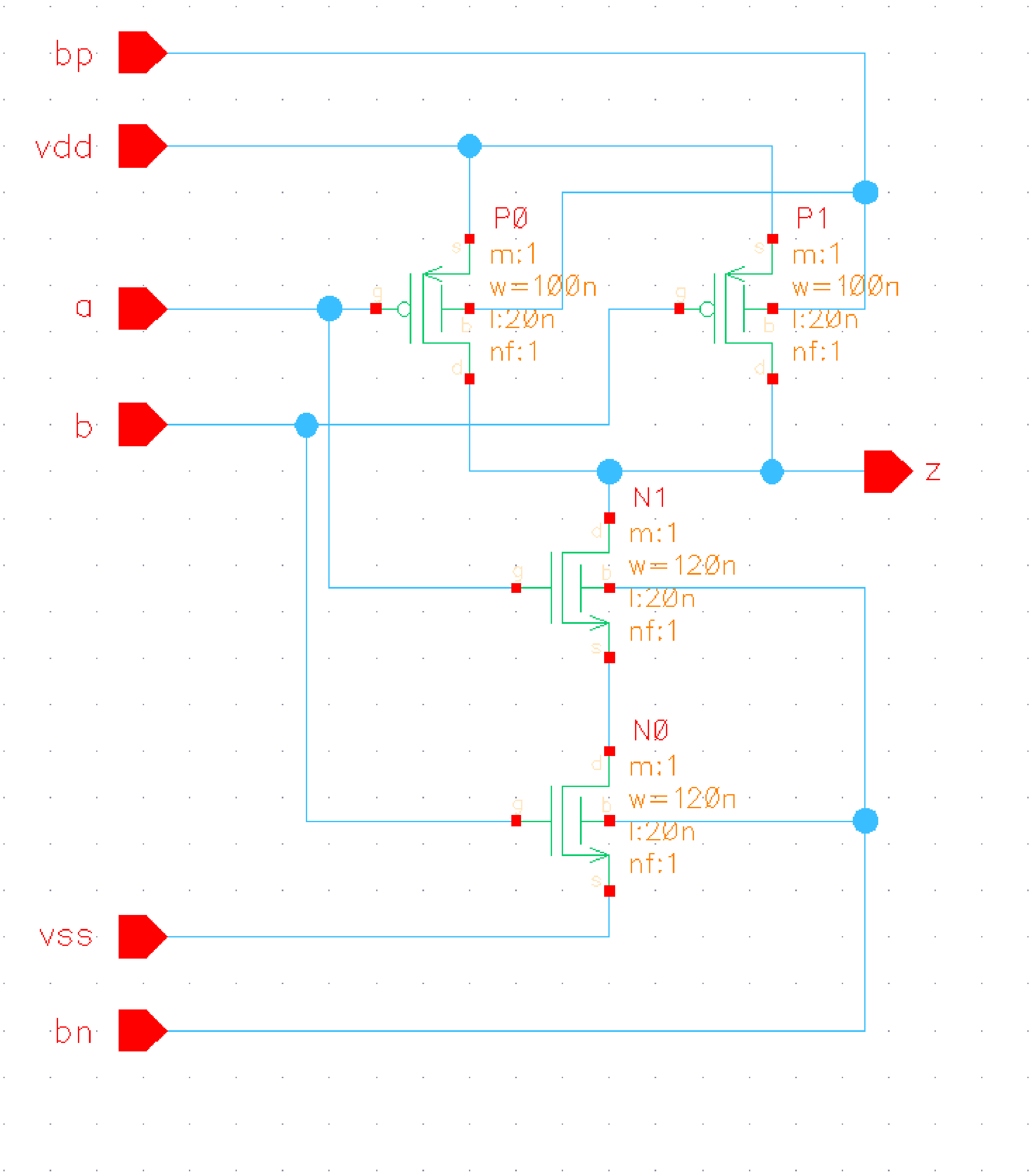


Figure 81: Schematic design of a NAND-gate by ***mosfet_high*** transistors, optimized for V_{DD}=350 mV according to table 18.

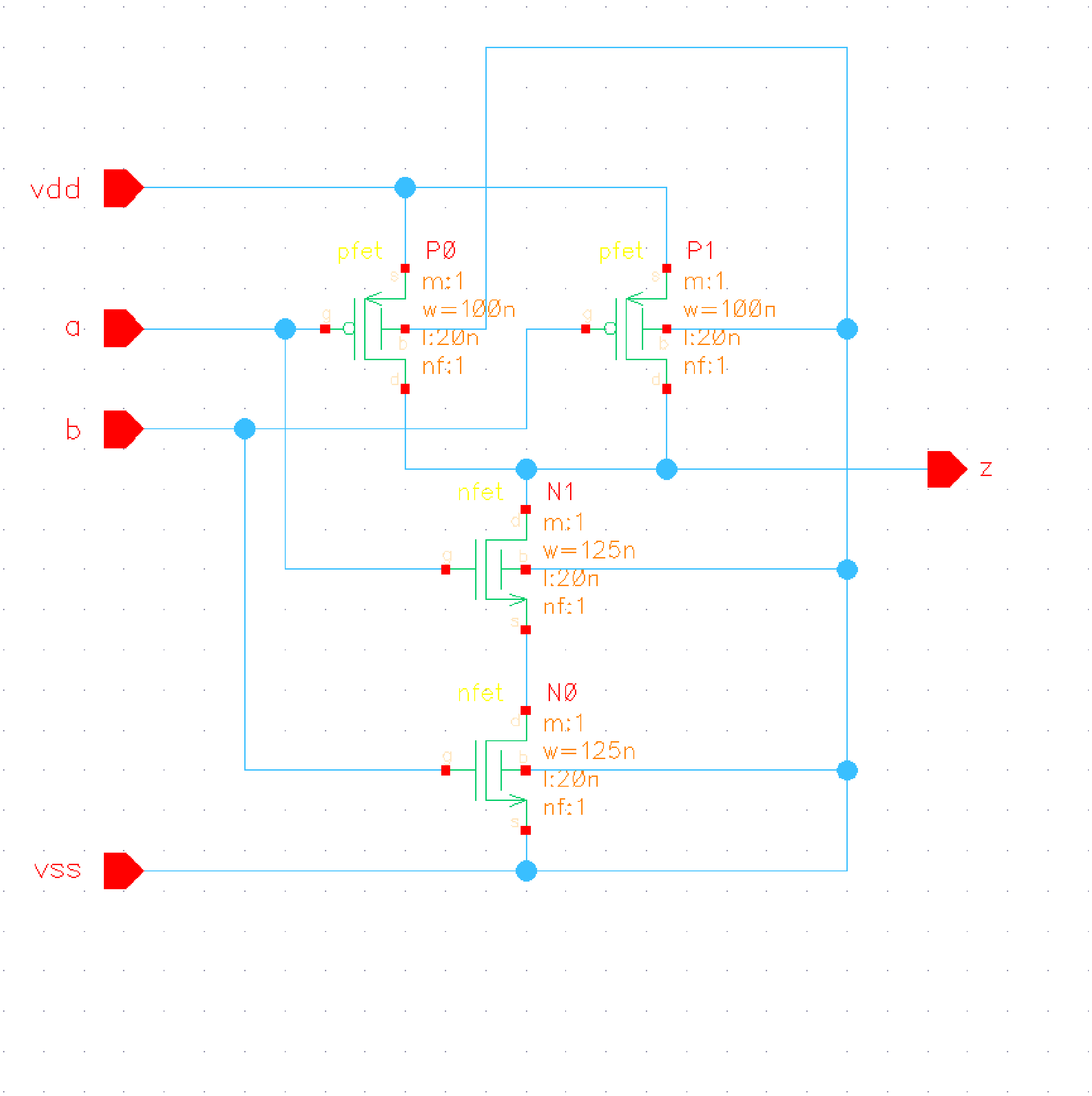


Figure 82: Schematic design of a NAND-gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 with no body-bias mV according to table 18.

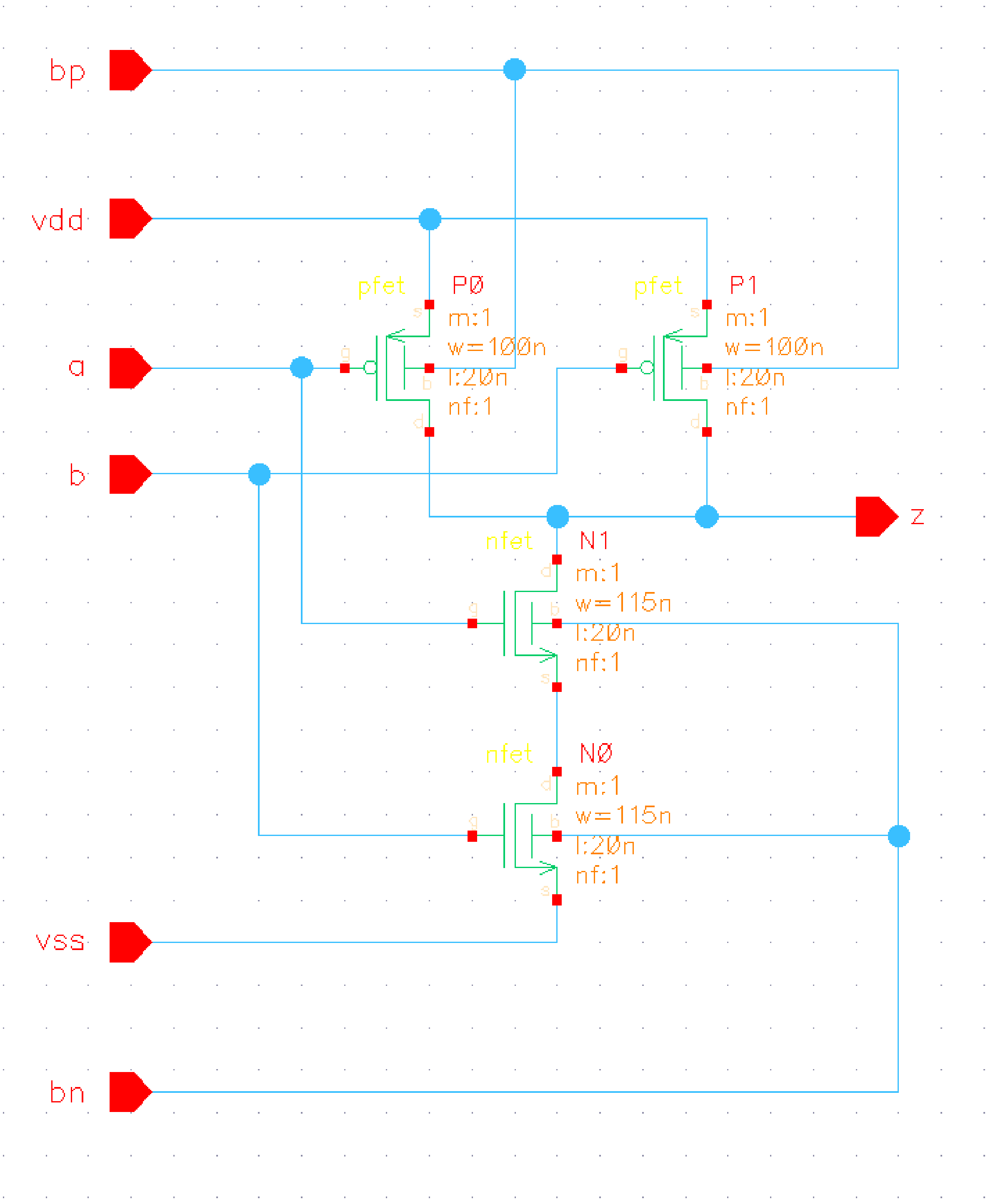


Figure 83: Schematic design of a NAND-gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with body-bias according to table 18.

NOR

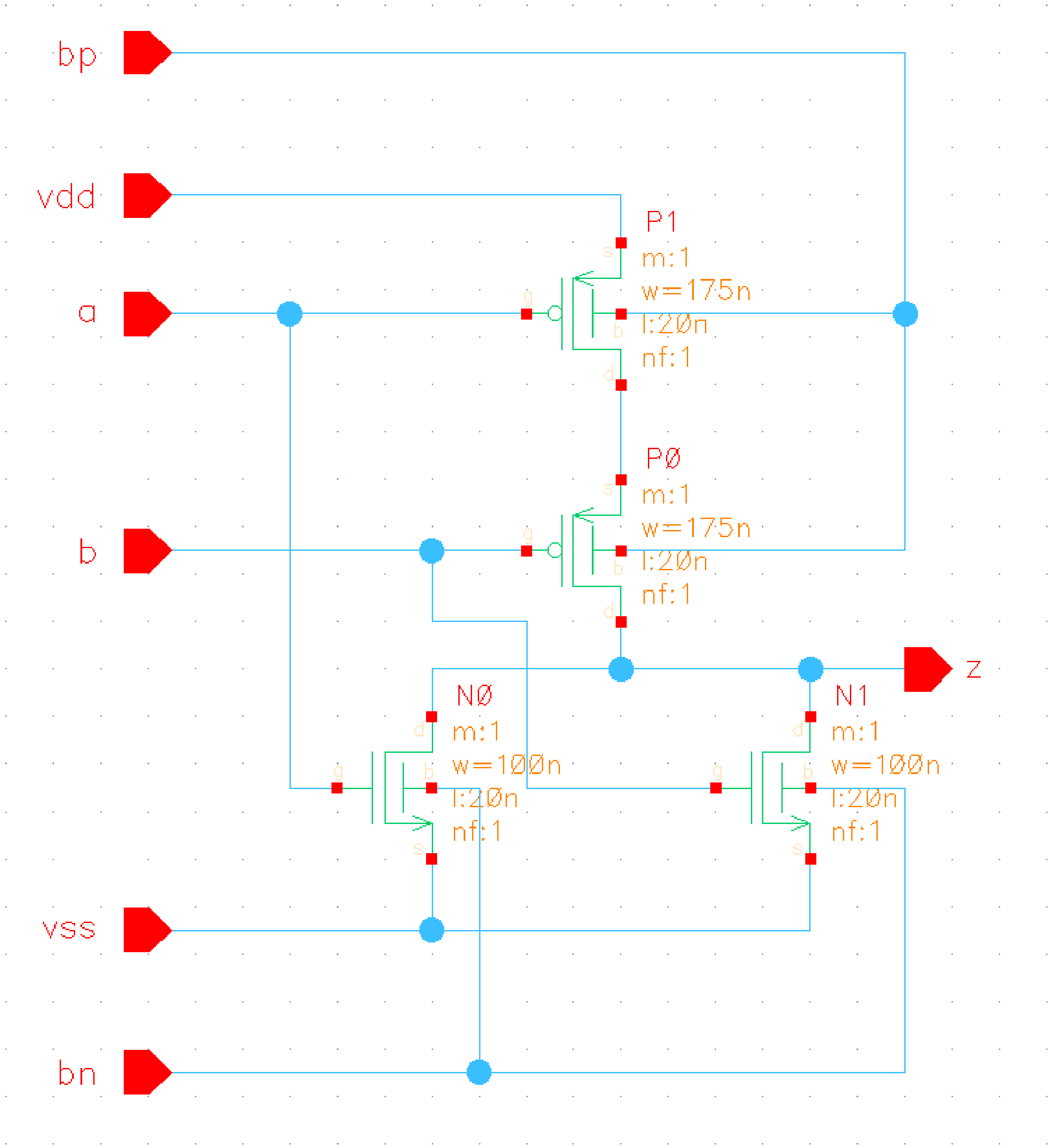


Figure 84: Schematic design of a NOR-gate by ***mosfet_high*** transistors, optimized for V_{DD}=350 mV according to table 18.

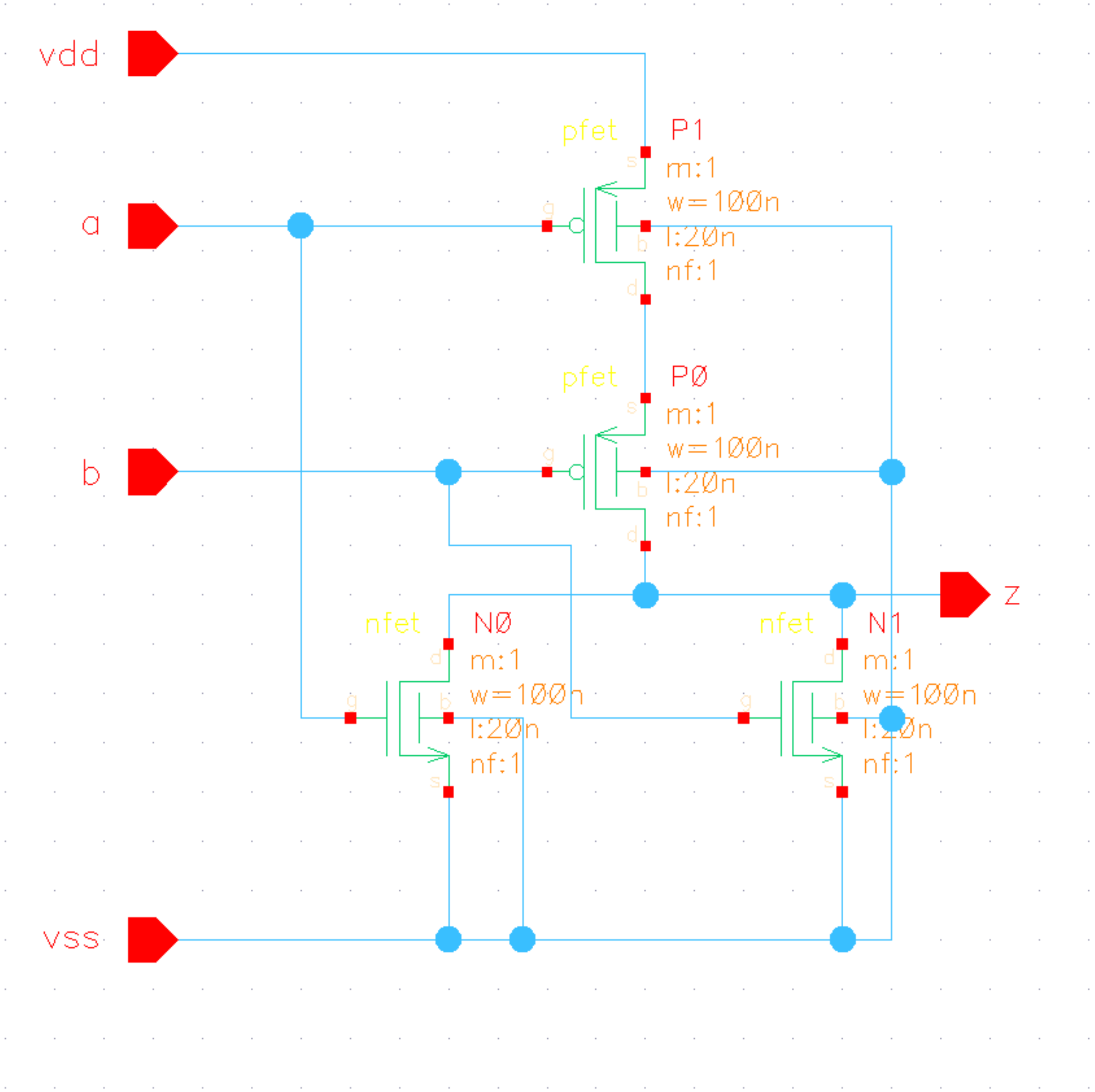


Figure 85: Schematic design of a NOR-gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with no body-bias according to table 18.

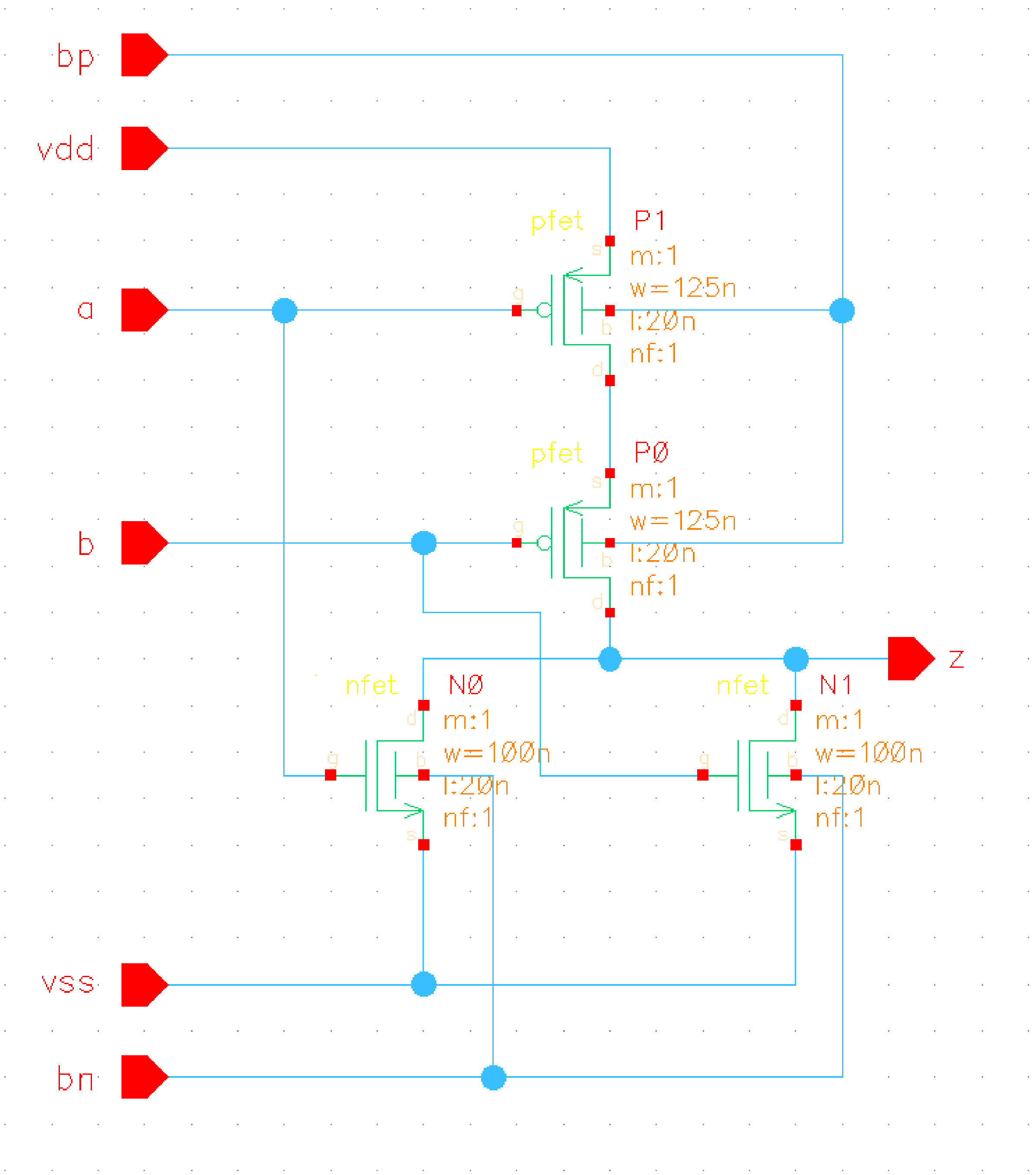


Figure 86: Schematic design of a NOR-gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with body-bias according to table 18.

XNOR

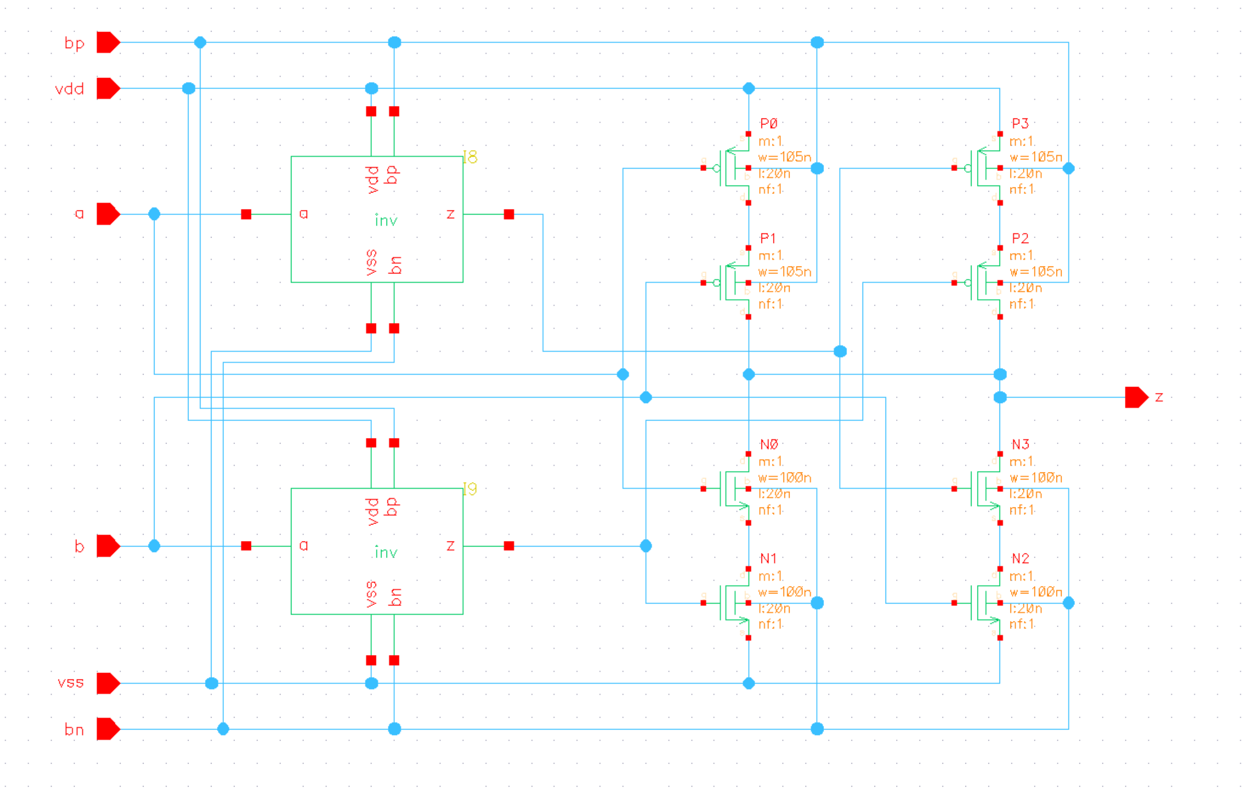


Figure 87: Schematic design of a XNOR-gate by ***mosfet_high*** transistors, optimized for V_{DD}=350 mV according to table 18.

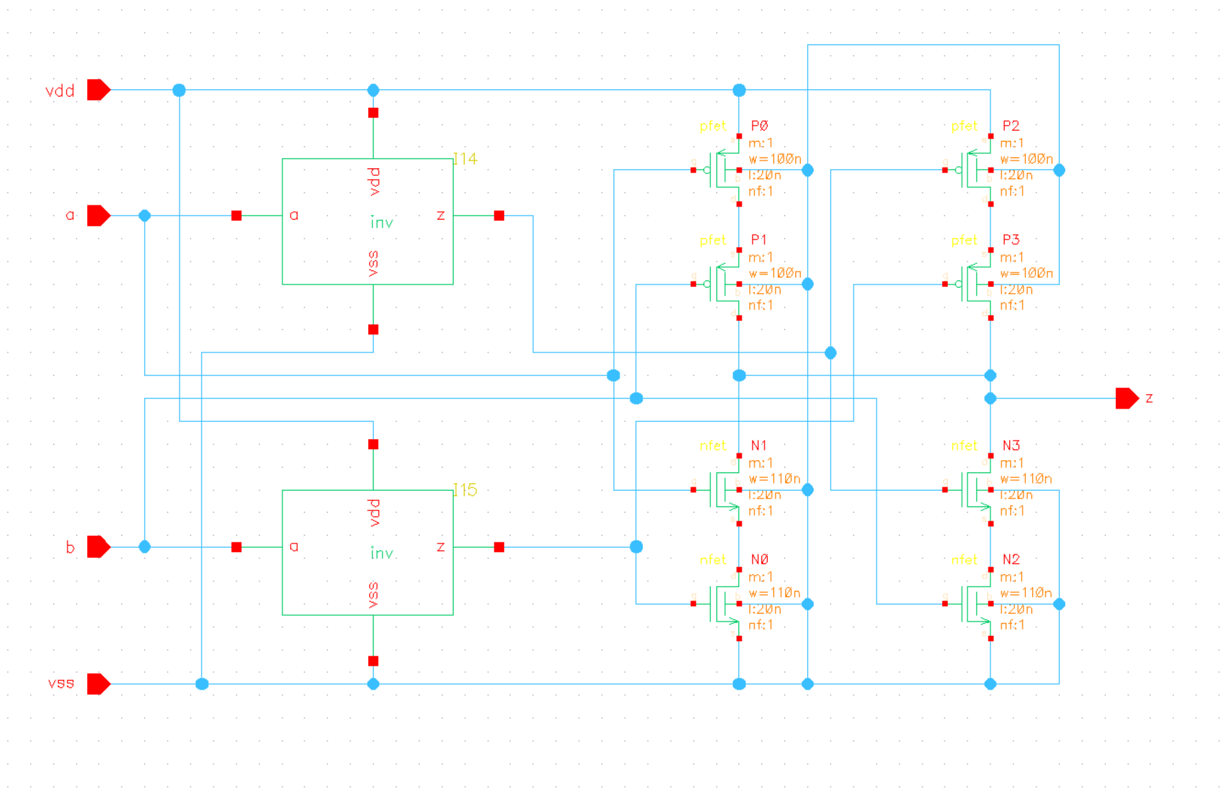


Figure 88: Schematic design of a XNOR-gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with no body-bias according to table 18.

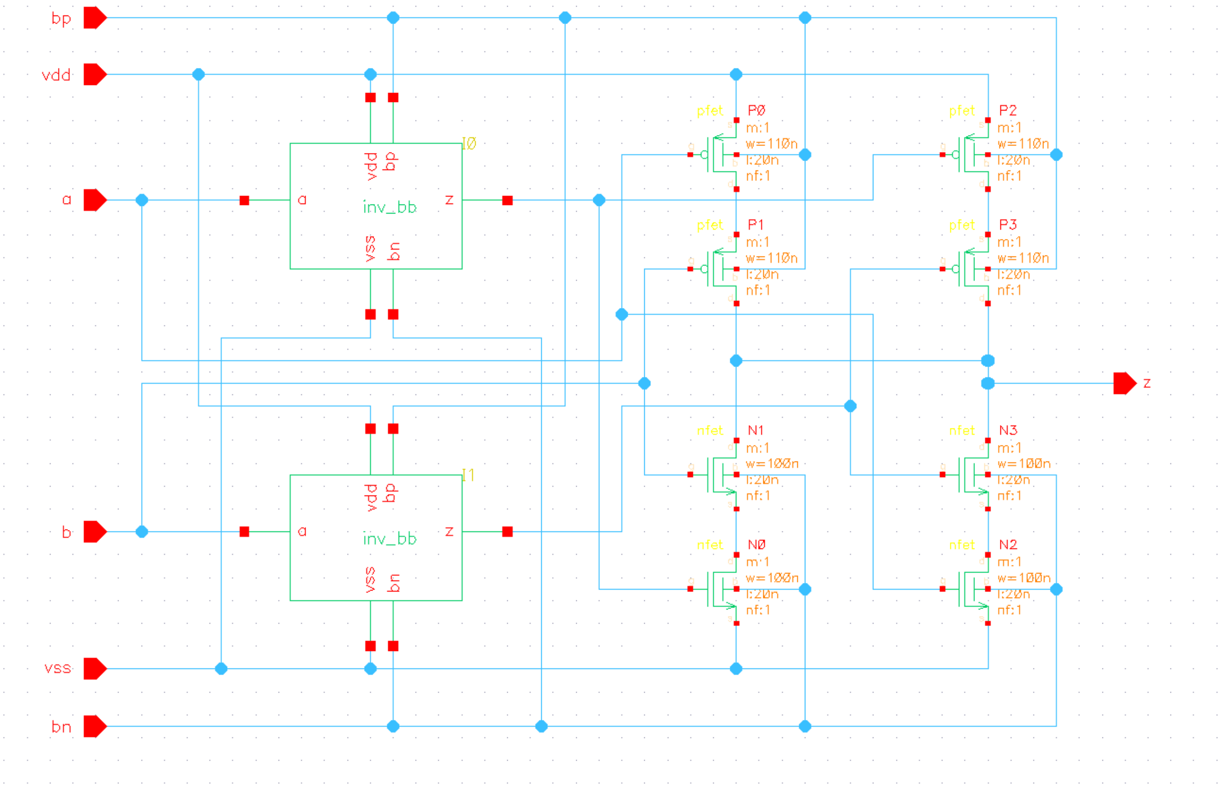


Figure 89: Schematic design of a XNOR-gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with body-bias according to table 18.

XOR

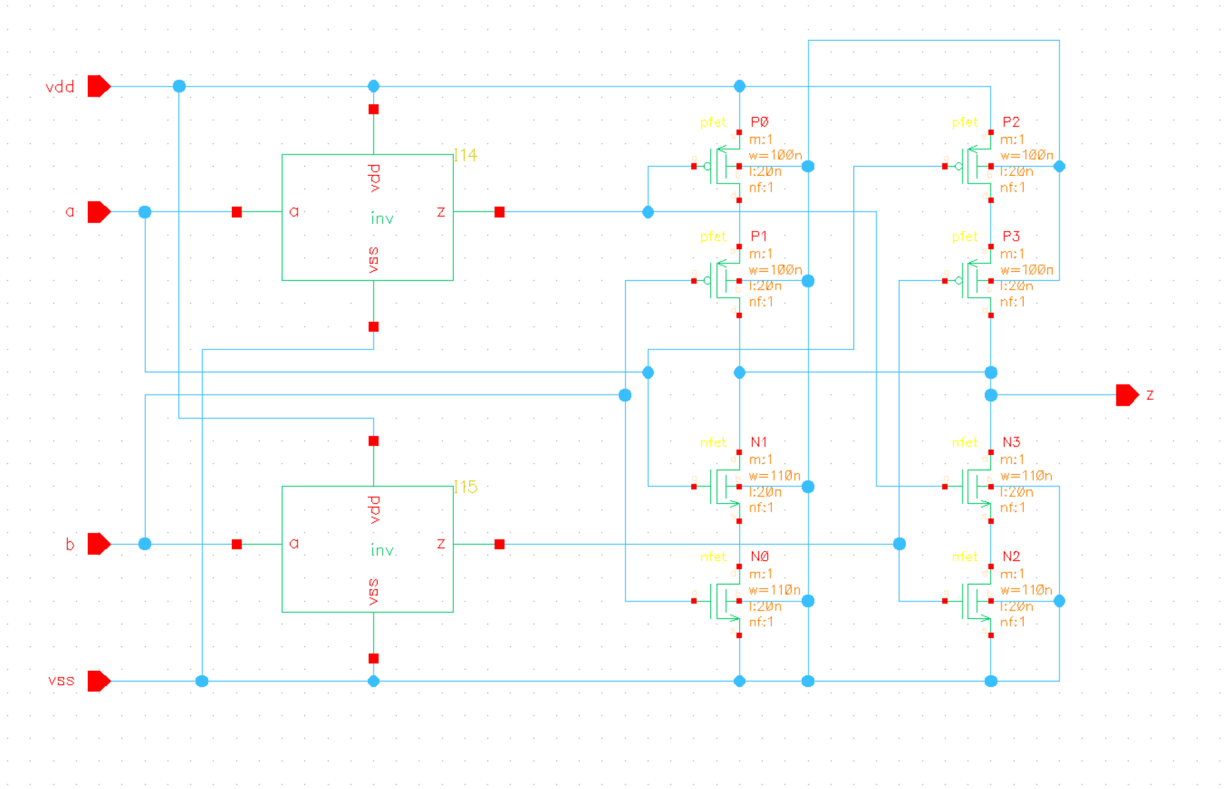


Figure 90: Schematic design of a XOR-gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with no body-bias according to table 18.

Minority-3

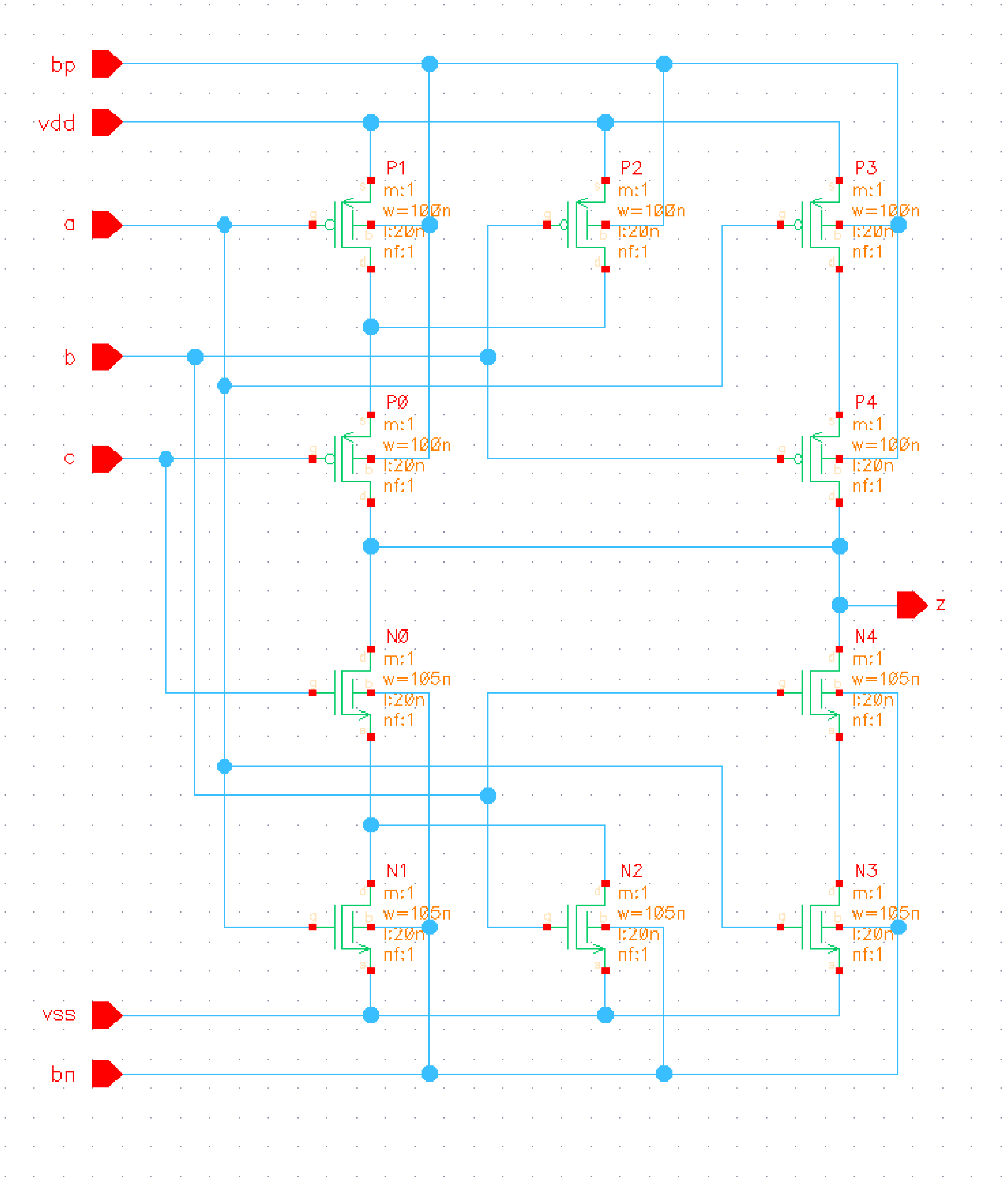


Figure 91: Schematic design of a Minority-3 gate by ***mosfet_high*** transistors, optimized for V_{DD}=350 mV according to table 18.

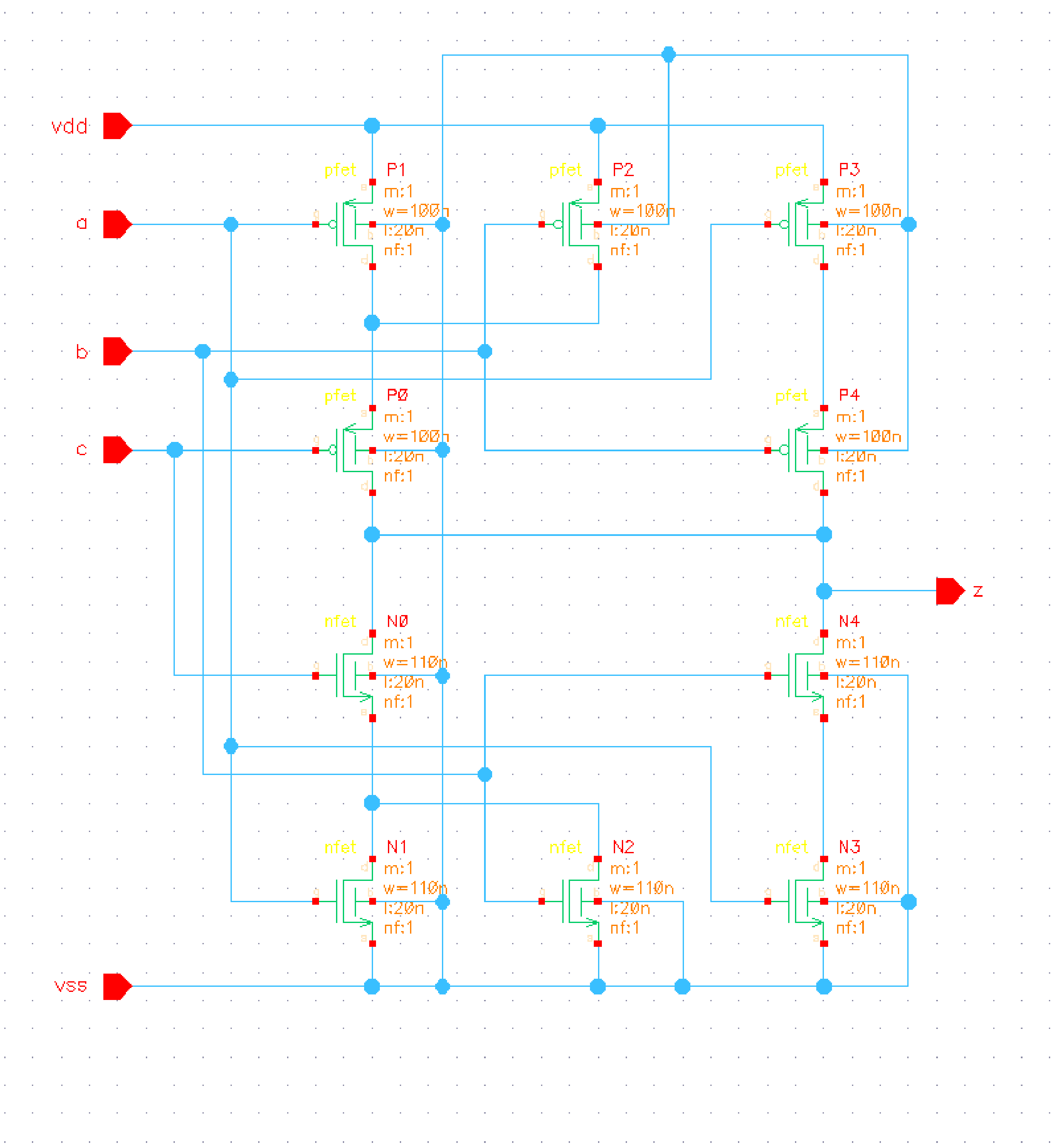


Figure 92: Schematic design of a Minority-3 gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with no body-bias according to table 18.

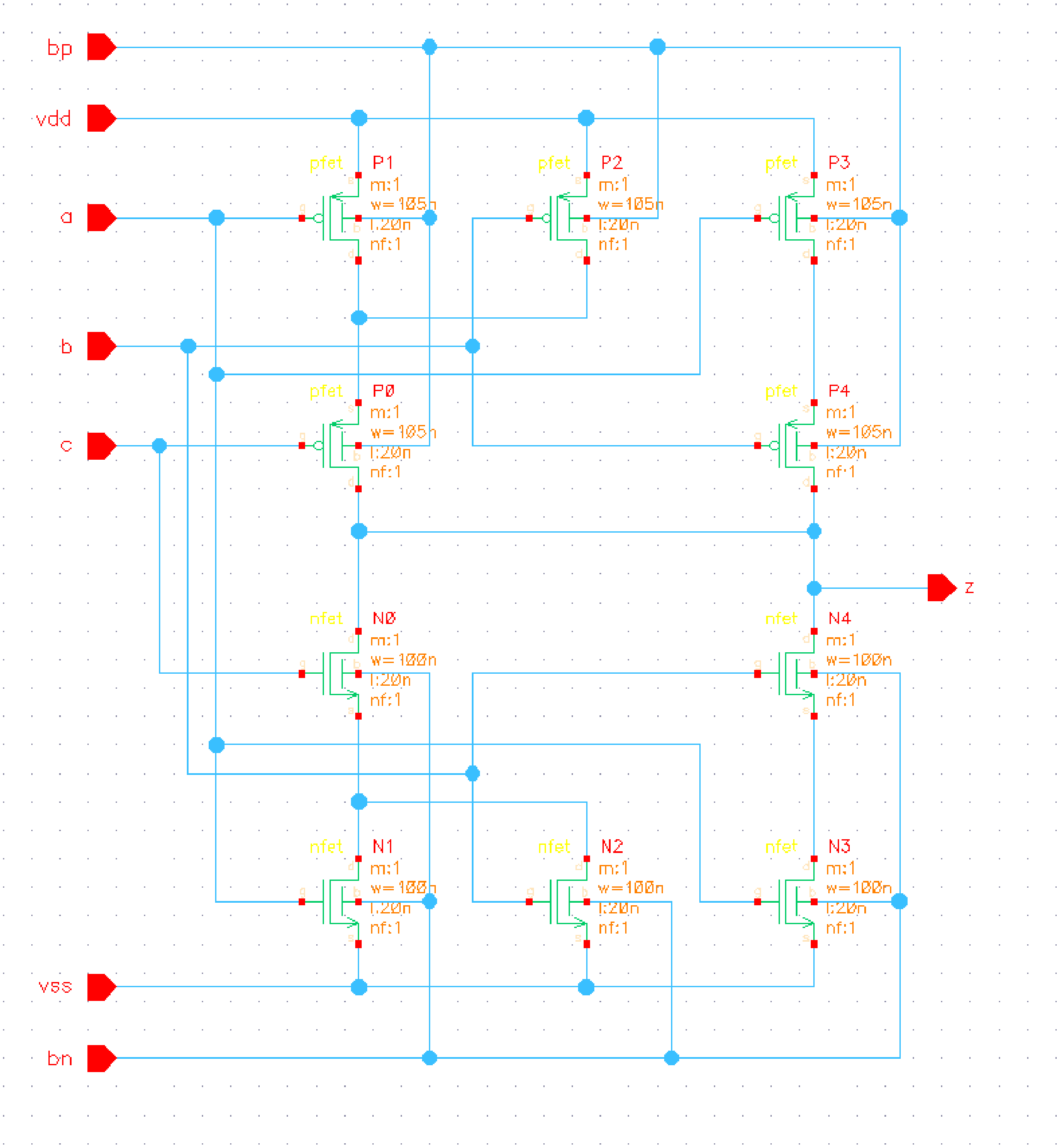


Figure 93: Schematic design of a Minority-3 gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with body-bias according to table 18.

Half-Adder

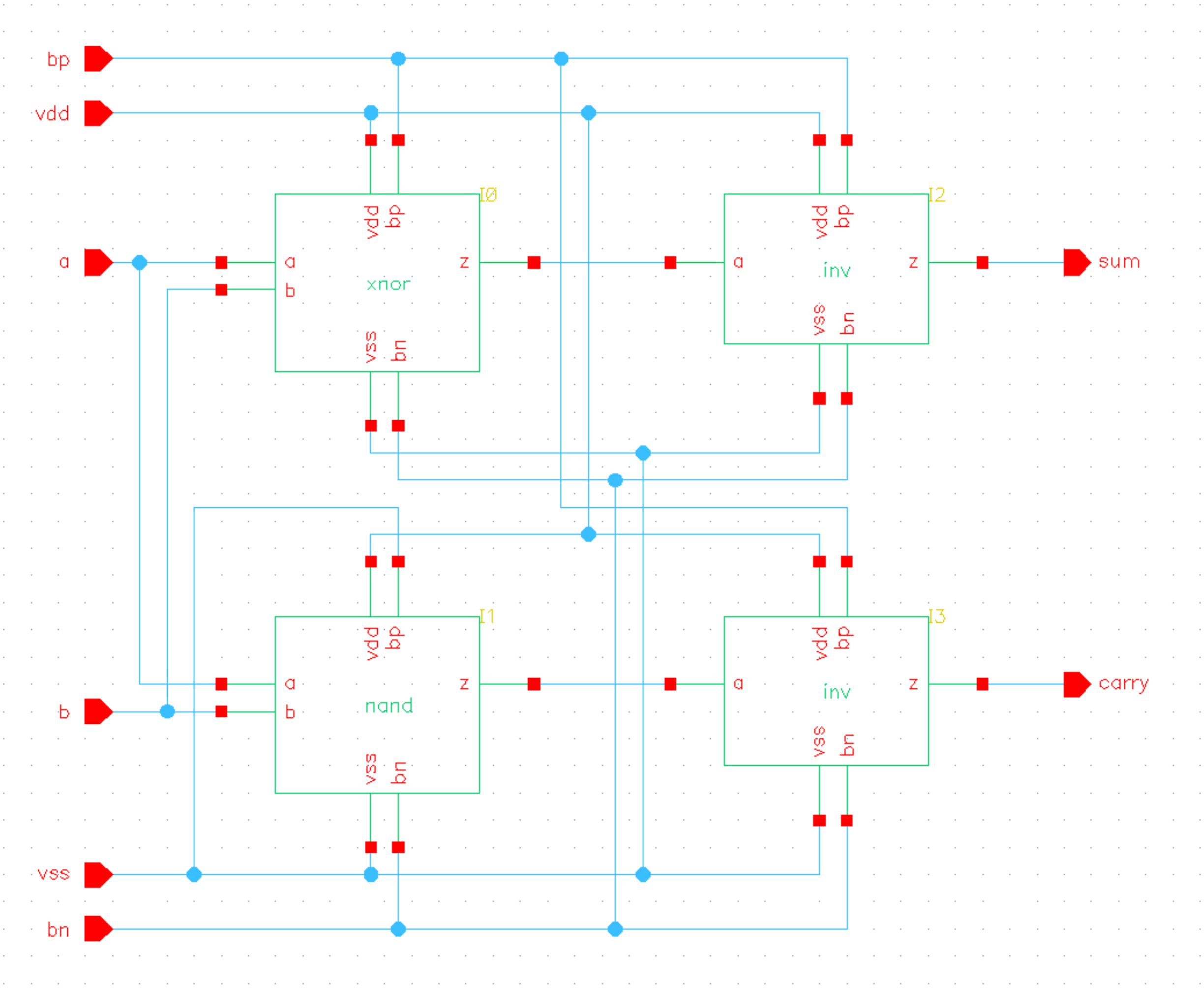


Figure 94: Schematic design of a Half-Adder gate by ***mosfet_high*** transistors, optimized for V_{DD}=350 mV according to table 18.

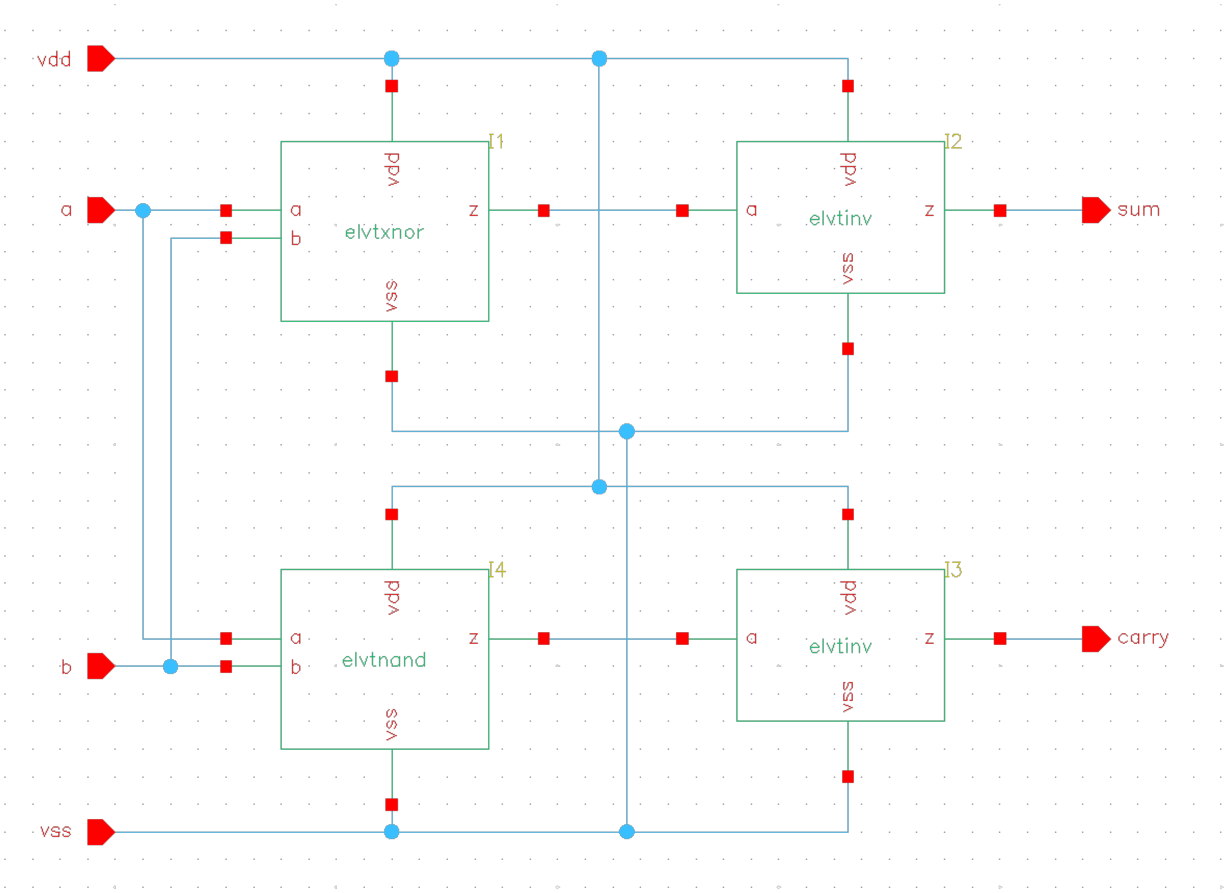


Figure 95: Schematic design of a Half-Adder gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with no body-bias according to table 18.

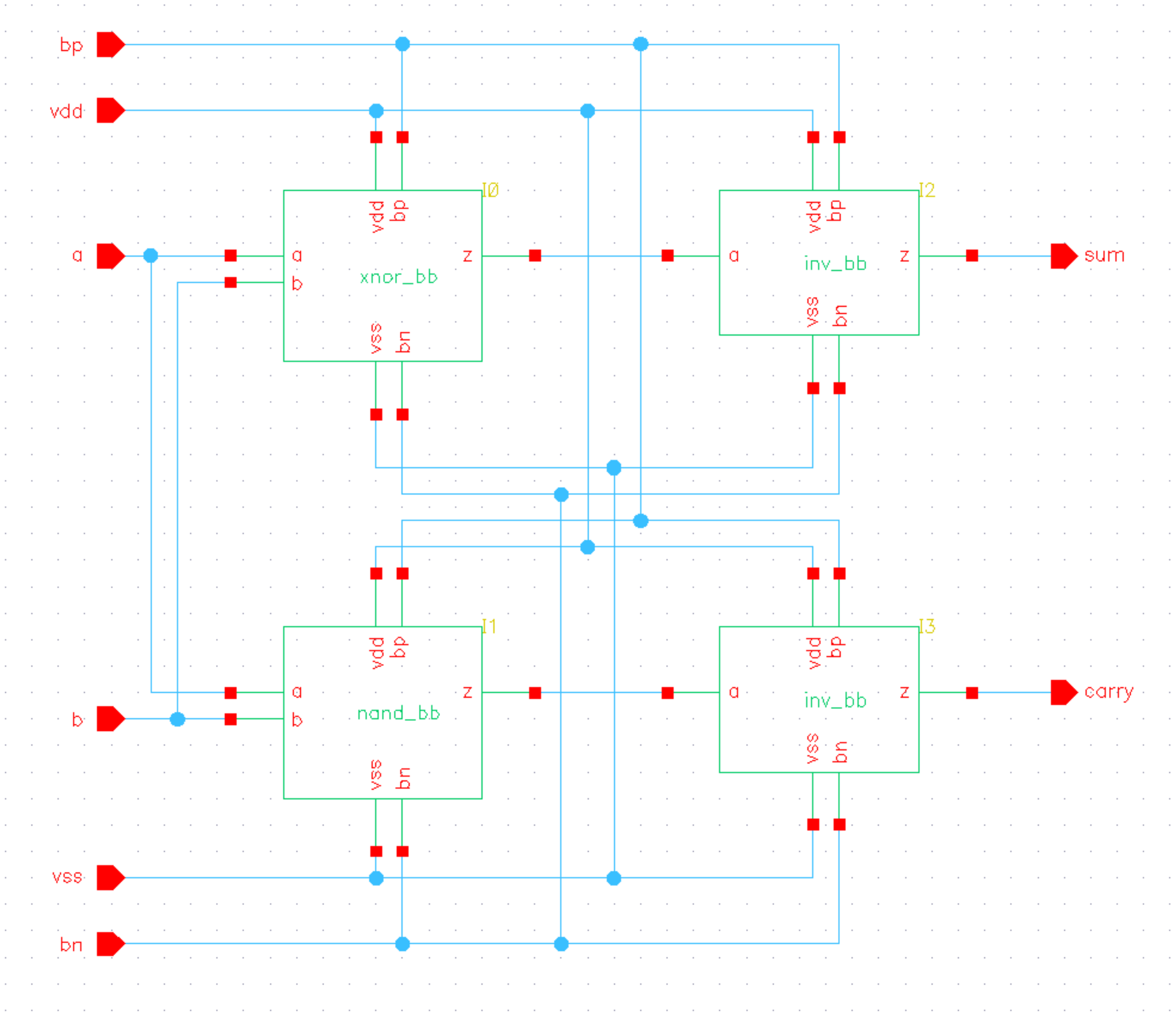


Figure 96: Schematic design of a Half-Adder gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with body-bias according to table 18.

Full-Adder

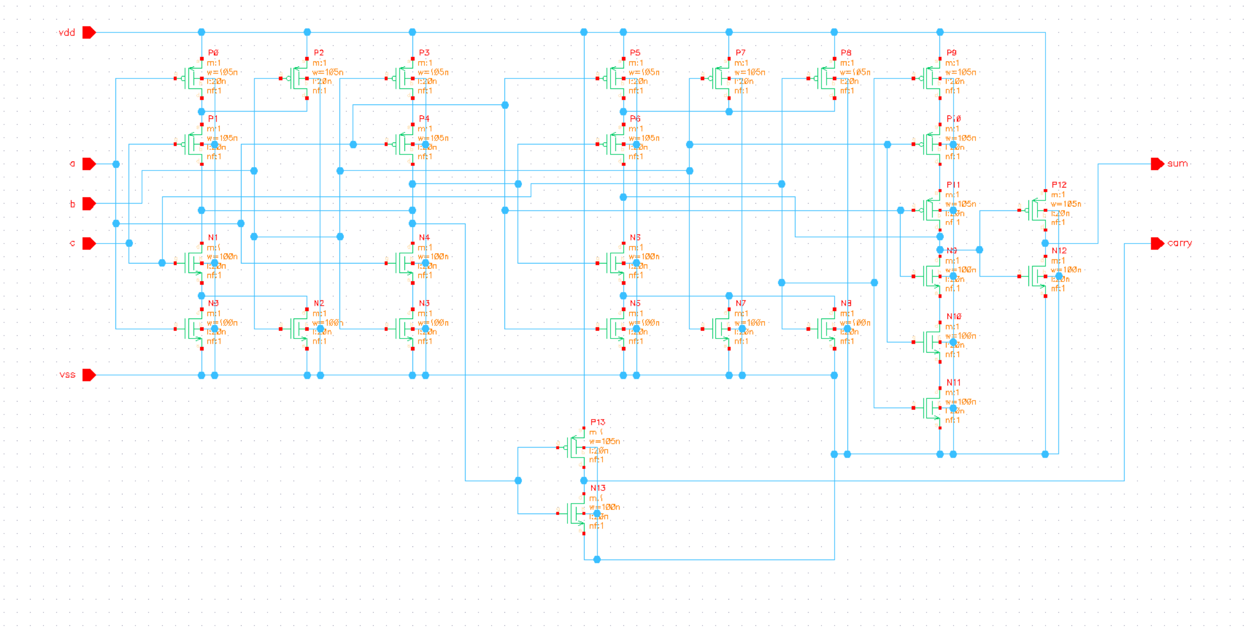


Figure 97: Schematic design of a standard Full-Adder gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with no body-bias according to table 18.

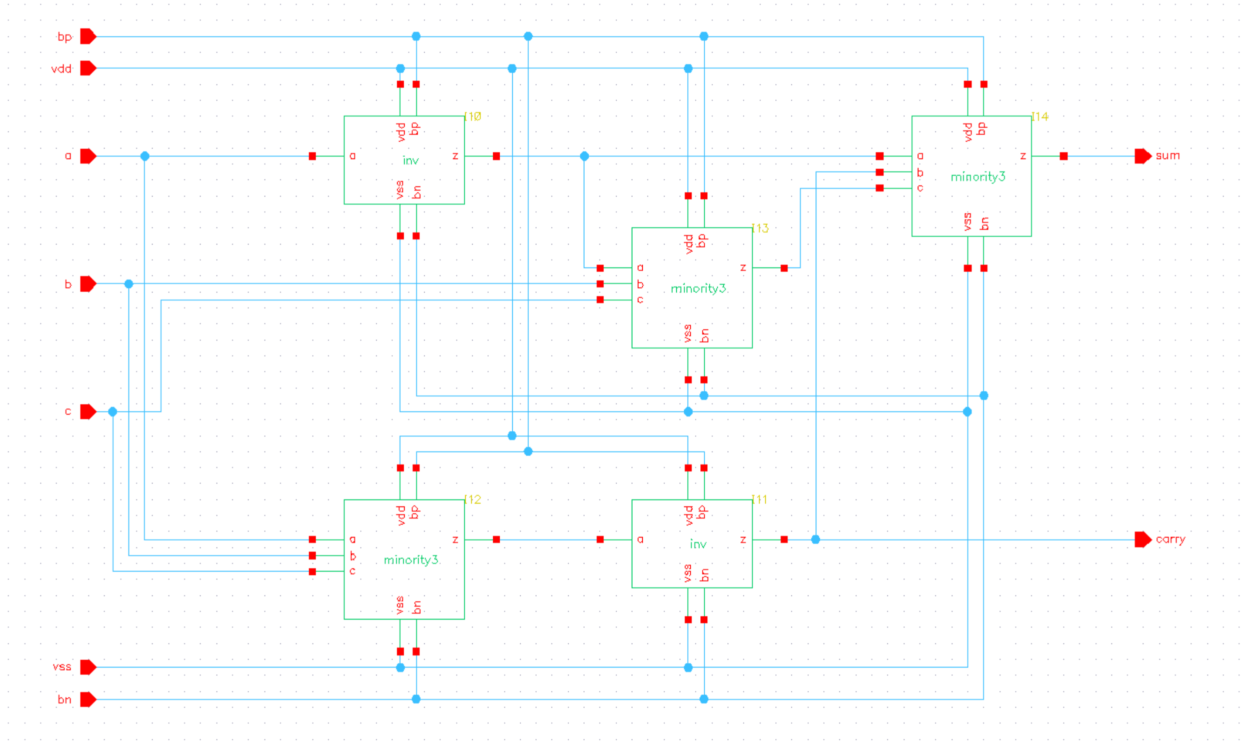


Figure 98: Schematic design of a Minority-3 based Full-Adder gate by ***mosfet_high*** transistors, optimized for V_{DD}=350 mV according to table 18.

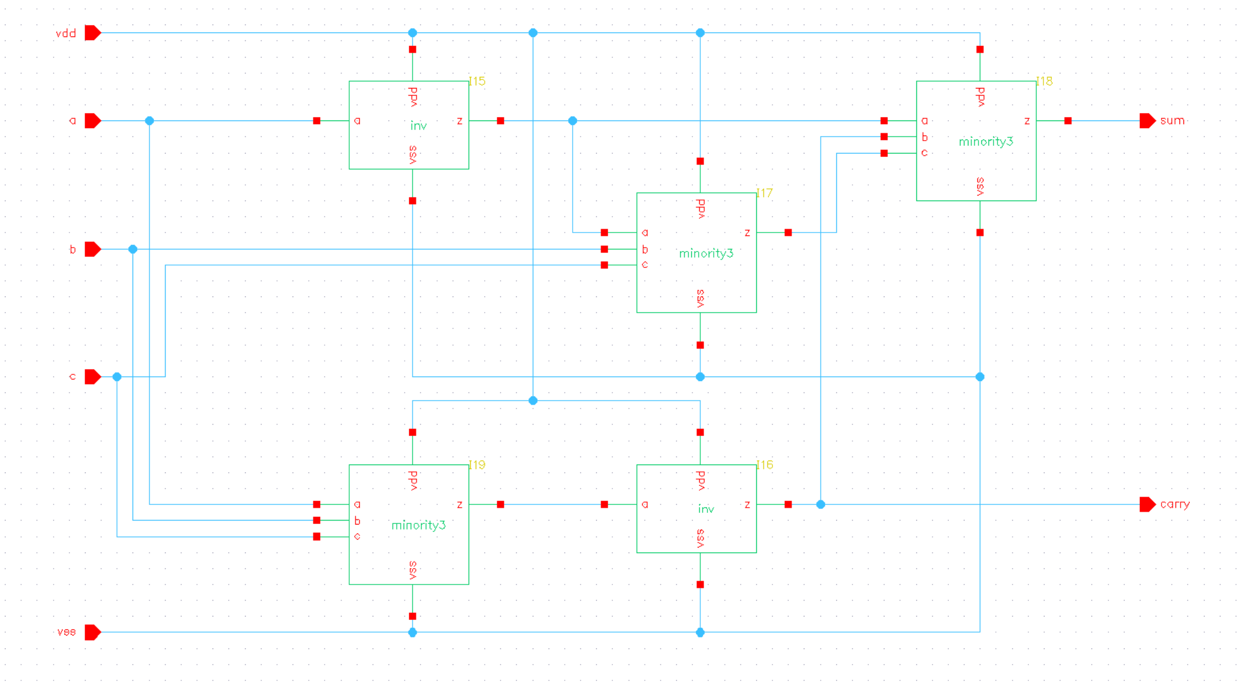


Figure 99: Schematic design of a Minority-3 based Full-Adder gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with no body-bias according to table 18.

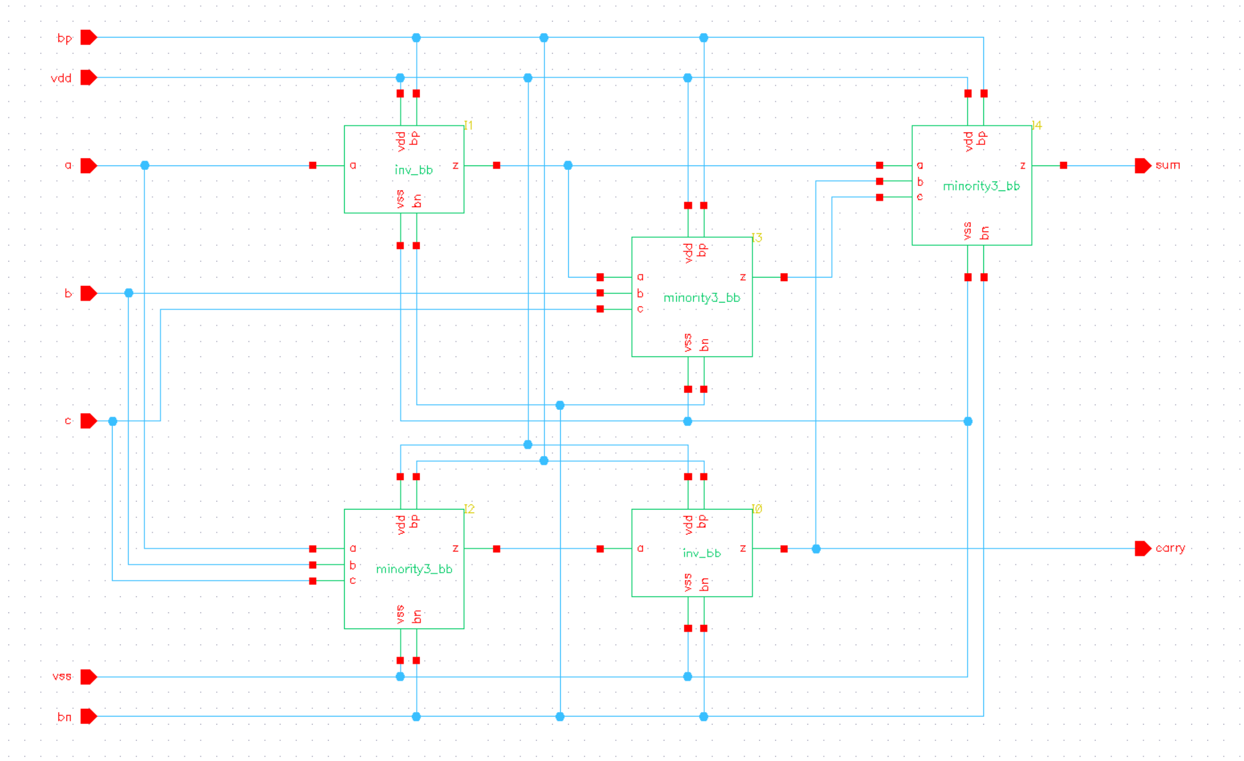


Figure 100: Schematic design of a Minority-3 based Full-Adder gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with body-bias according to table 18.

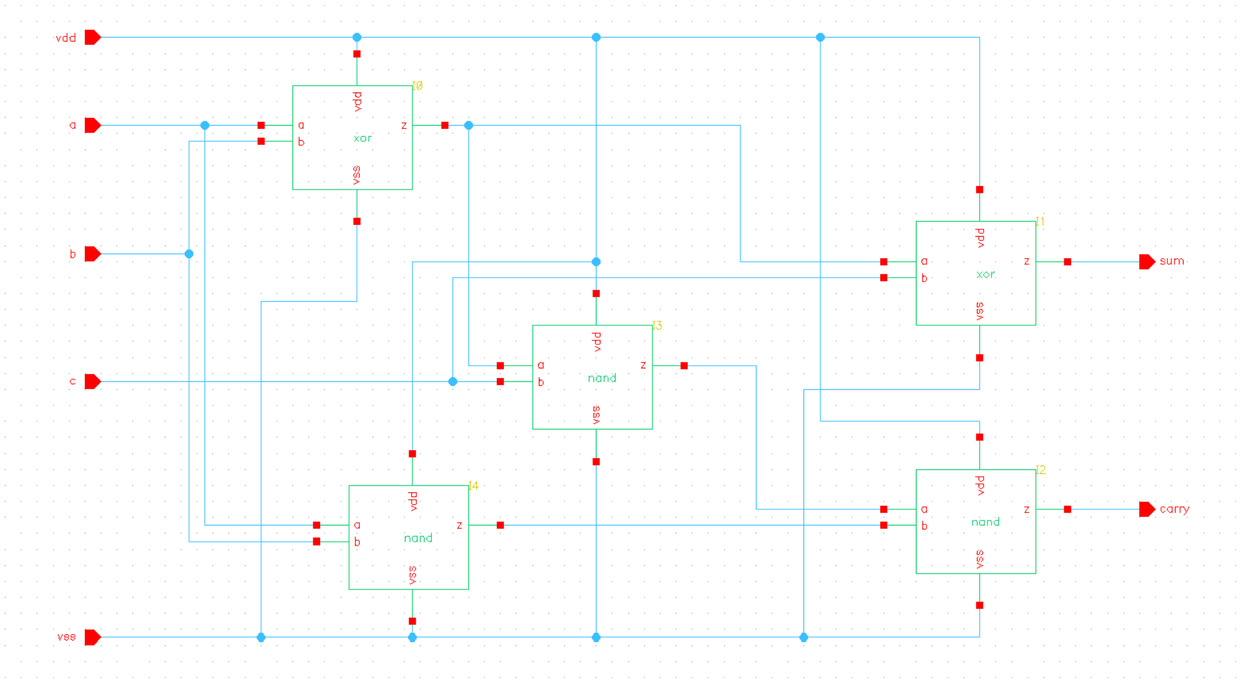


Figure 101: Schematic design of a NAND & XOR based Full-Adder gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with no body-bias according to table 18.

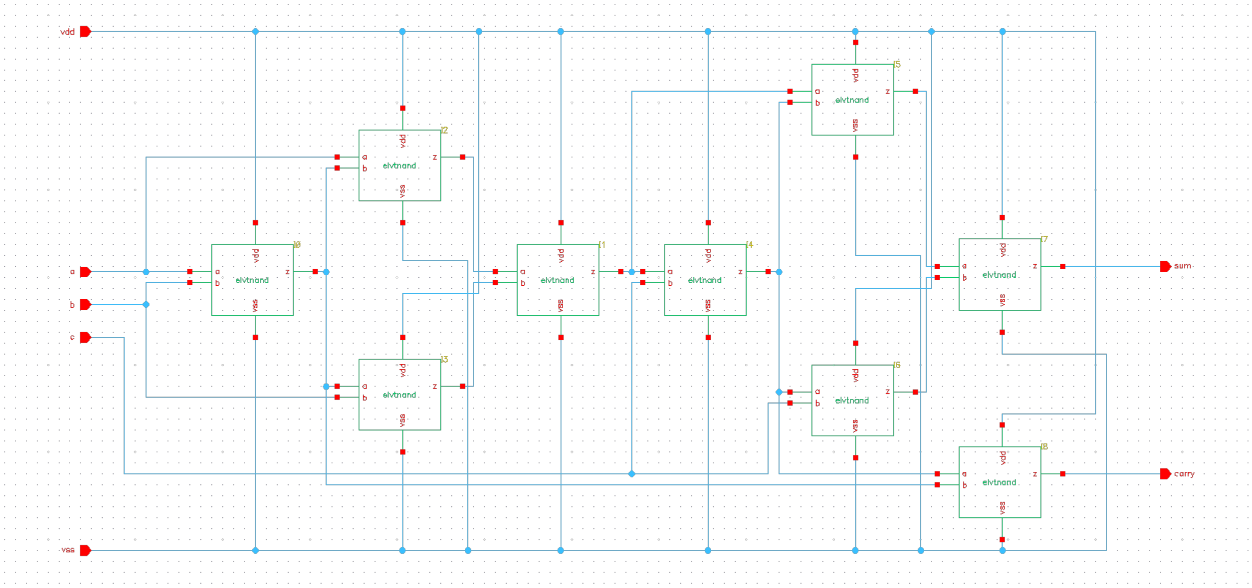


Figure 102: Schematic design of a NAND based Full-Adder gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with no body-bias according to table 18.

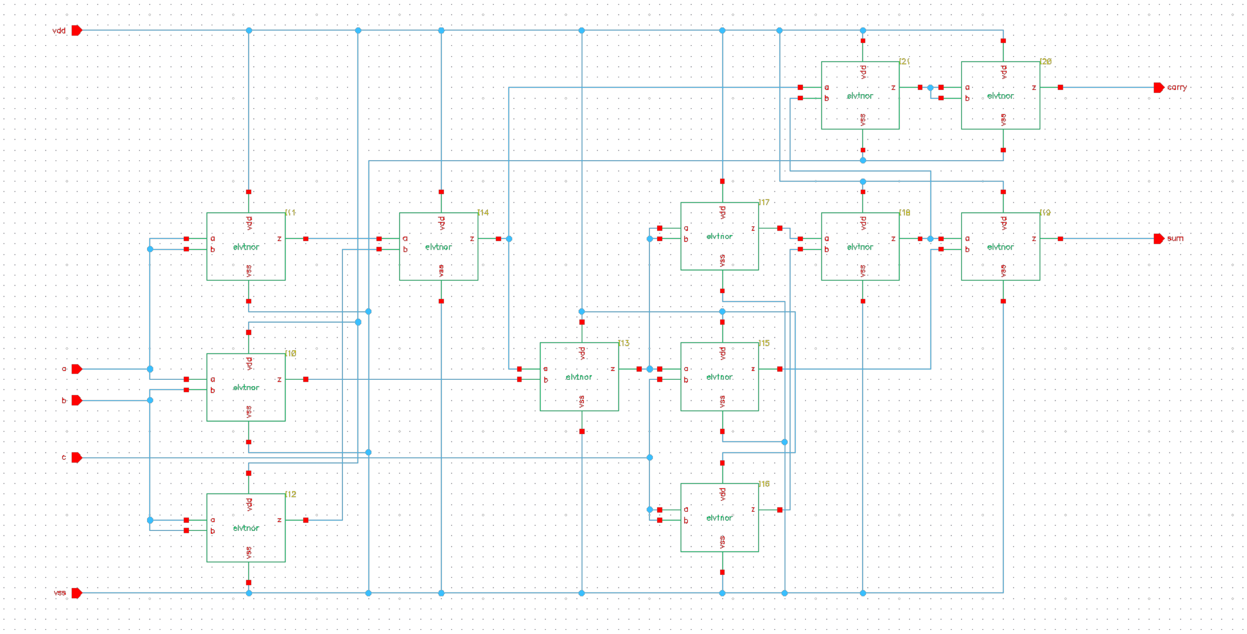


Figure 103: Schematic design of a NOR based Full-Adder gate by by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with no body-bias according to table 18.

8-bit Ripple-Carry Adder

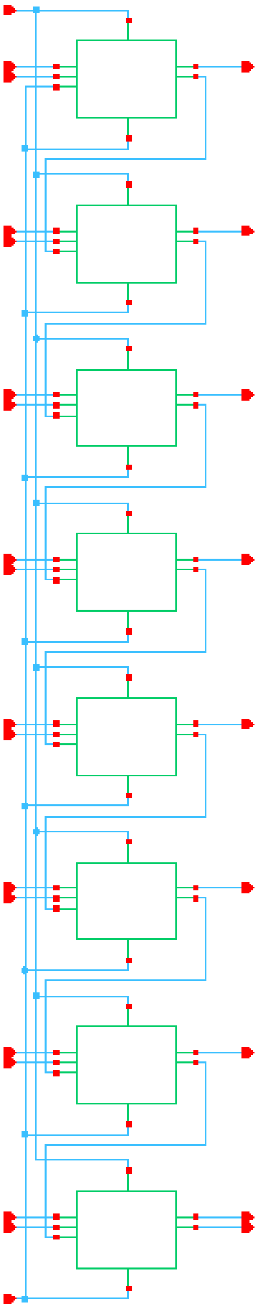


Figure 104: Schematic design of an 8-bit Ripple-Carry Adder by Minority-3 based Full-Adder gates. The Full-Adders are built by ***mosfet_low*** transistors, given by the schematic in figure 99. The design is optimized for V_{DD}=350 mV with no body-bias according to table 18.

12-bit Ripple-Carry Adder

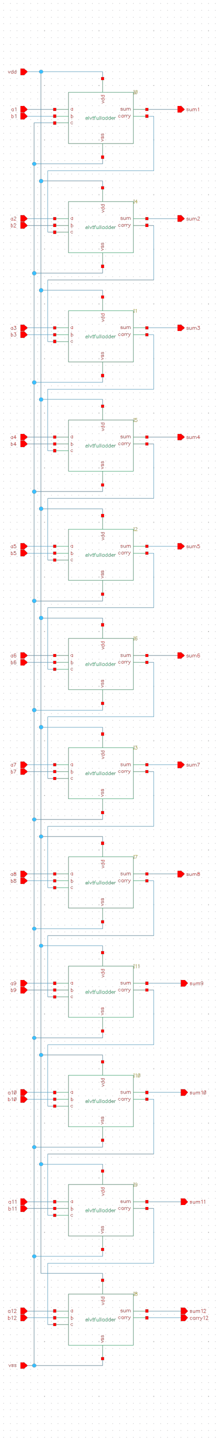


Figure 105: Schematic design of a 12-bit Ripple-Carry Adder by Minority-3 based Full-Adder gates. The Full-Adders are built by ***mosfet_low*** transistors, given by the schematic in figure 99. The design is optimized for V_{DD}=350 mV with no body-bias according to table 18.

13-bit Ripple-Carry Adder

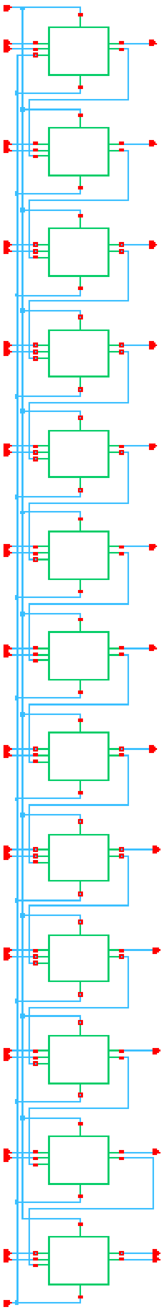


Figure 106: Schematic design of a 13-bit Ripple-Carry Adder by Minority-3 based Full-Adder gates. The Full-Adders are built by ***mosfet_low*** transistors, given by the schematic in figure 99. The design is optimized for V_{DD}=350 mV with no body-bias according to table 18.

14-bit Ripple-Carry Adder

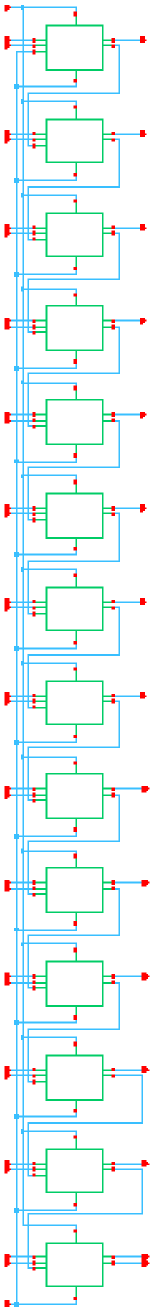


Figure 107: Schematic design of a 14-bit Ripple-Carry Adder by Minority-3 based Full-Adder gates. The Full-Adders are built by ***mosfet_low*** transistors, given by the schematic in figure 99. The design is optimized for V_{DD}=350 mV with no body-bias according to table 18.

15-bit Ripple-Carry Adder

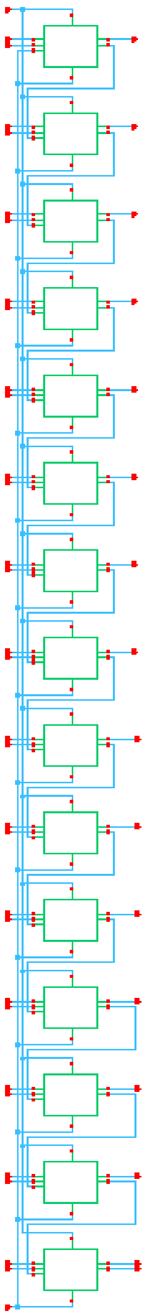


Figure 108: Schematic design of a 15-bit Ripple-Carry Adder by Minority-3 based Full-Adder gates. The Full-Adders are built by ***mosfet_low*** transistors, given by the schematic in figure 99. The design is optimized for V_{DD}=350 mV with no body-bias according to table 18.

16-bit Ripple-Carry Adder

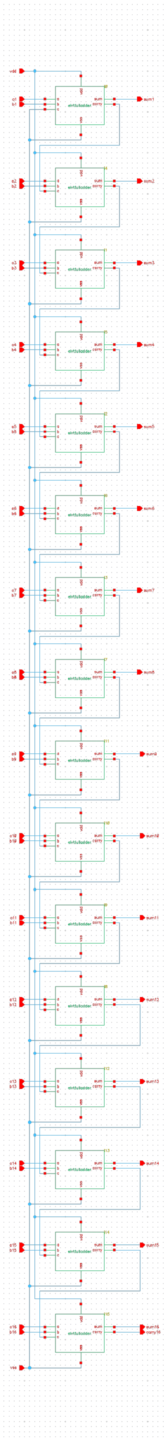


Figure 109: Schematic design of a 16-bit Ripple-Carry Adder by Minority-3 based Full-Adder gates. The Full-Adders are built by ***mosfet_low*** transistors, given by the schematic in figure 99. The design is optimized for V_{DD}=350 mV with no body-bias according to table 18.

16by12-bit Adder

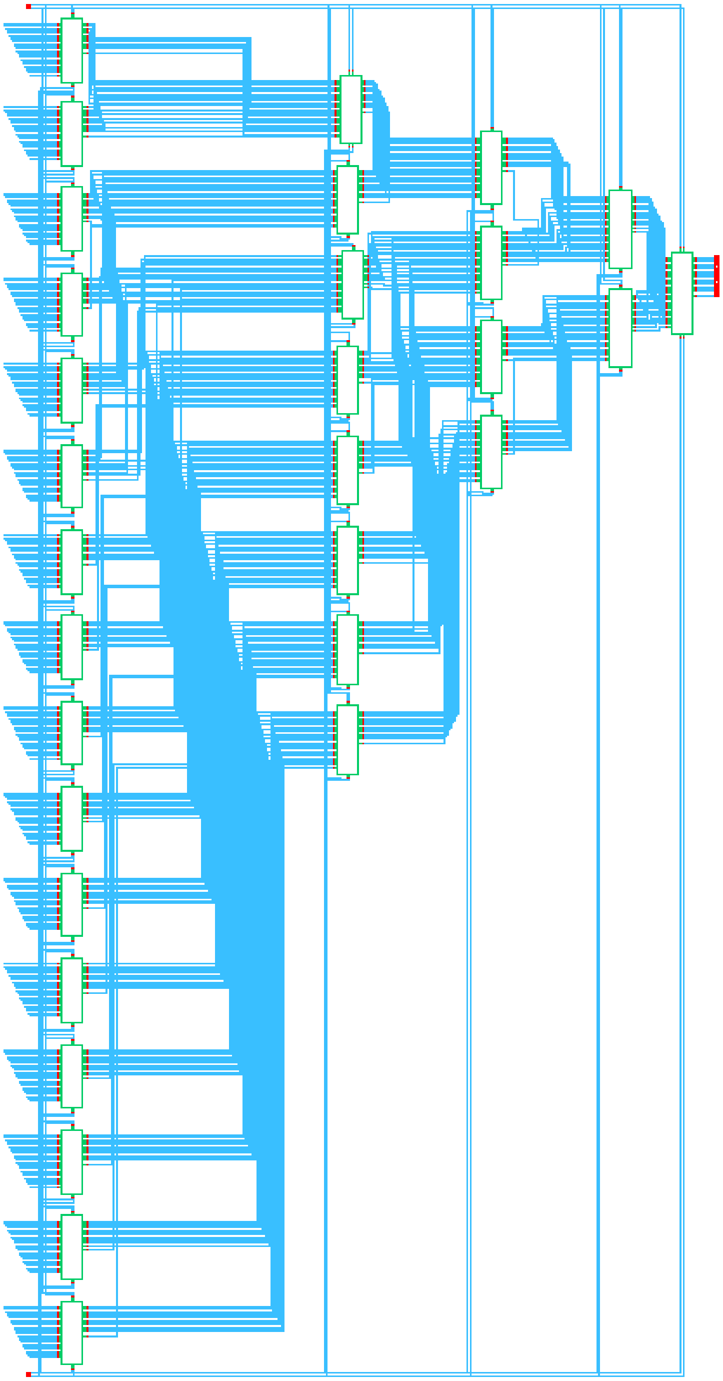


Figure 110: Schematic overview of the 16by12-bit adder.

D Flip-Flop

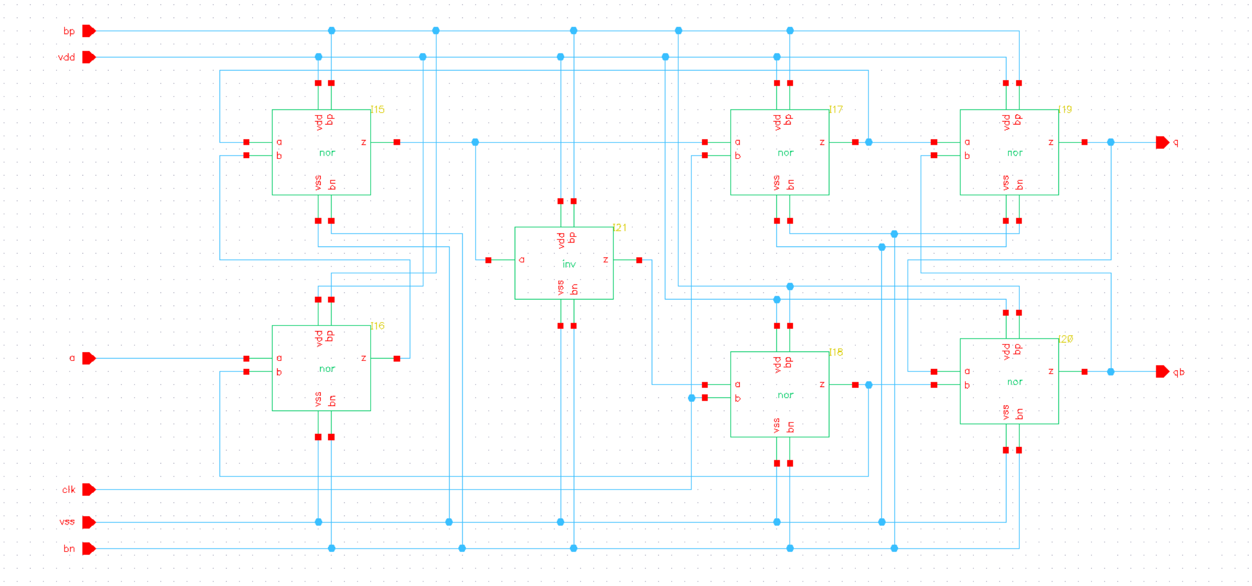


Figure 111: Schematic design of a D Flip-Flop gate by ***mosfet_high*** transistors, optimized for V_{DD}=350 mV according to table 18.

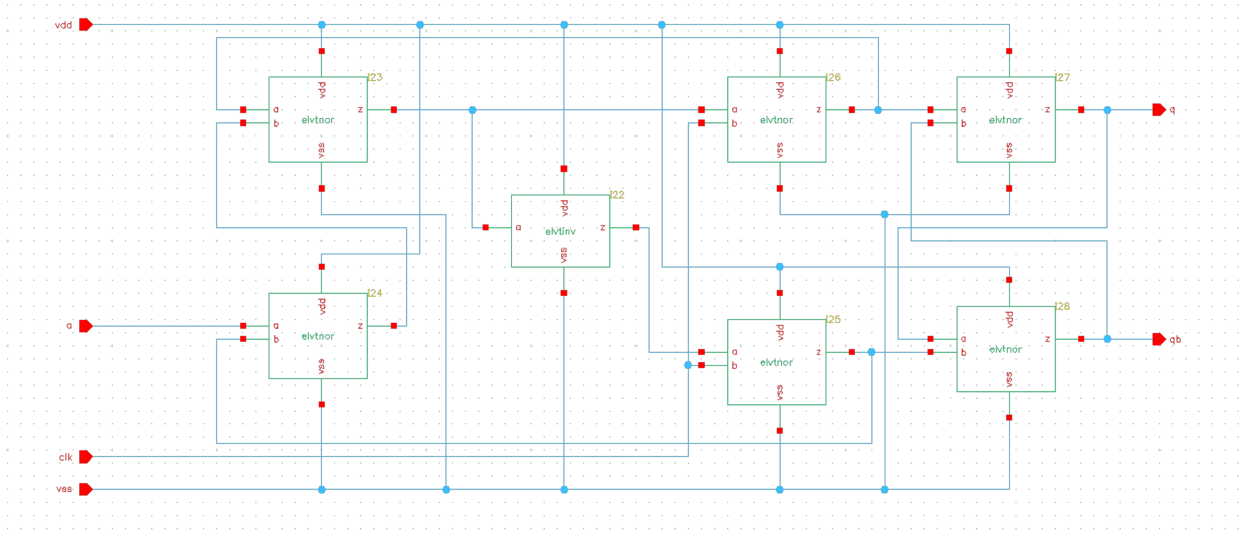


Figure 112: Schematic design of a D Flip-Flop gate by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV according to table 18.

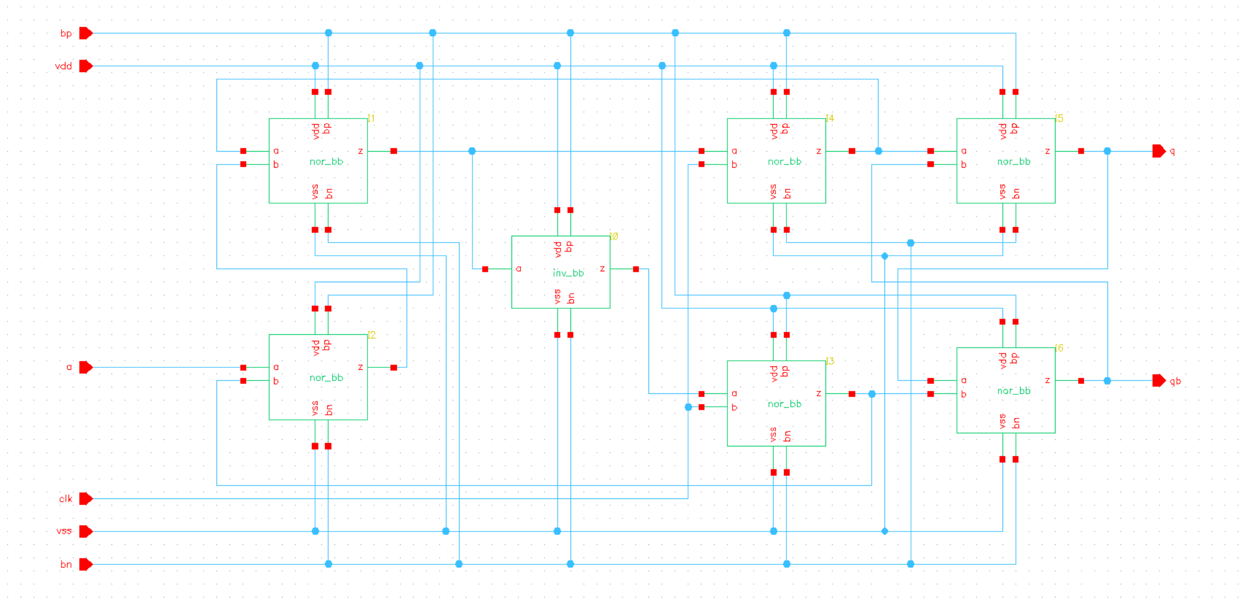


Figure 113: Schematic design of a D Flip-Flop gate by ***mosfet_low*** transistors, optimized for V_{DD}=350 mV with body-bias according to table 18.

Layout

Bulk connection

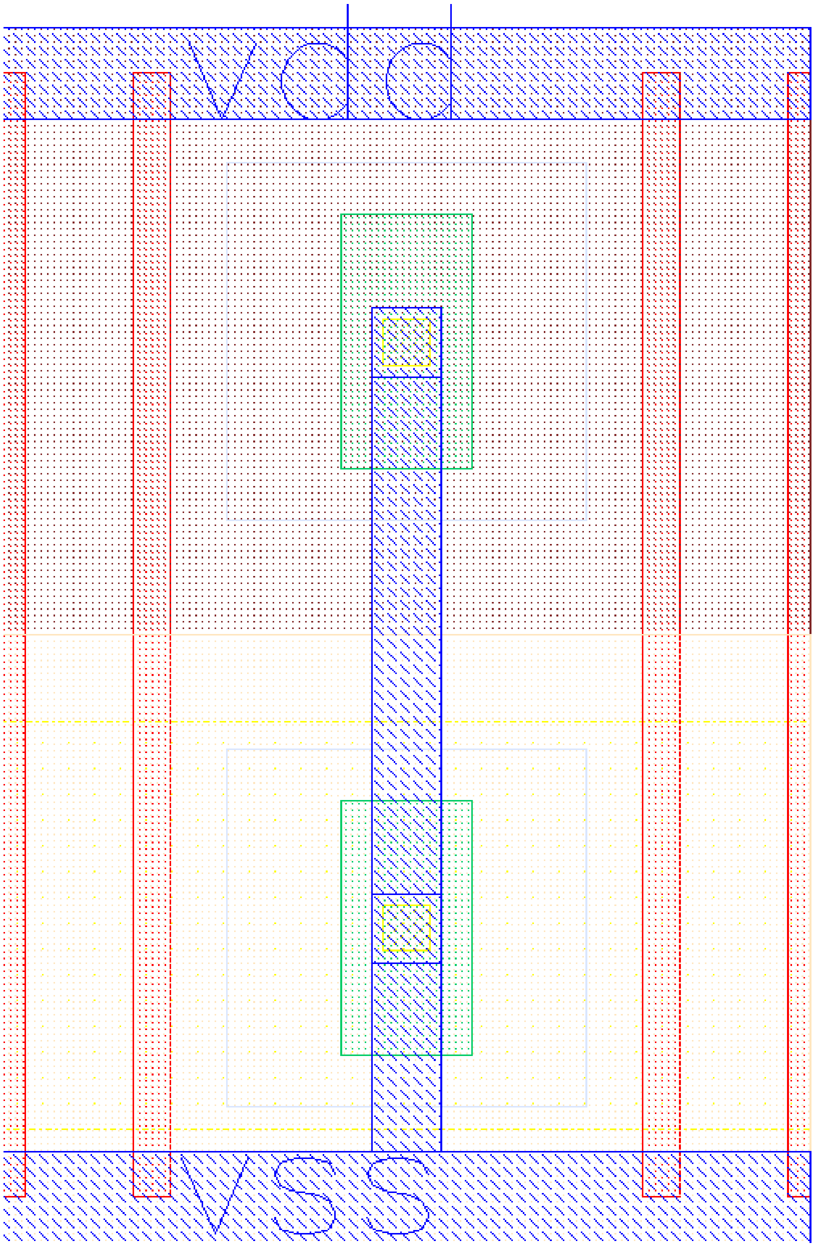


Figure 114: Layout design of a bulk connection by ***mosfet_low*** transistors, with no body-bias applied. Width and height of the cell is respectively 0.706 μm and 1.062 μm.

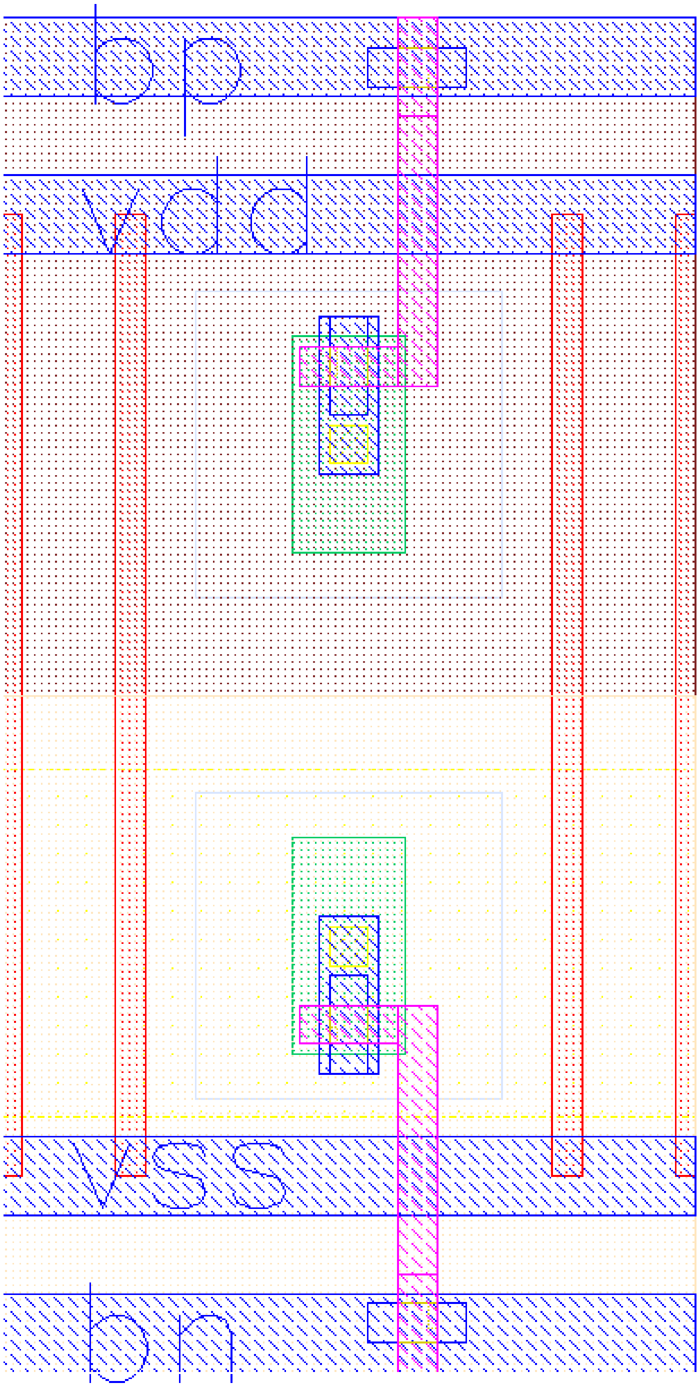


Figure 115: Layout design of a bulk connection by ***mosfet_low_bb*** transistors, with voltage applied to bulk nodes. Width and height of the cell is respectively 0.706 μm and 1.382 μm.

Inverter

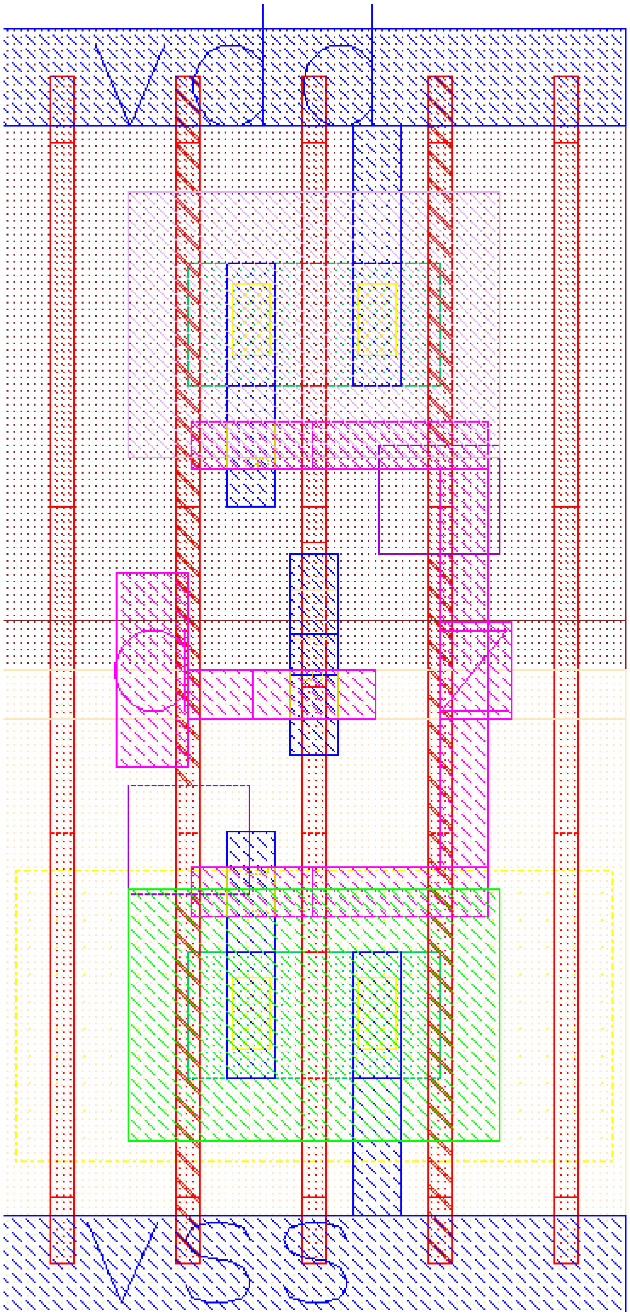


Figure 116: Layout design of an inverter by ***mosfet_low*** transistors, given by the schematic in figure 79. Width and height of the cell is respectively 0.416 μm and 1.062 μm.

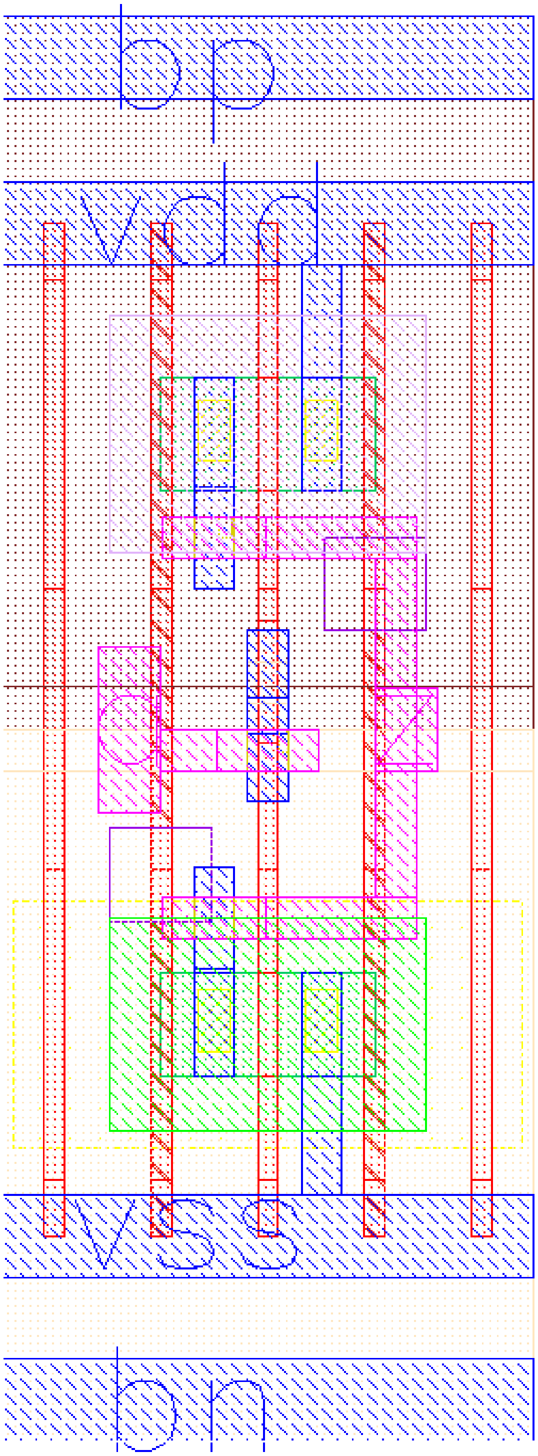


Figure 117: Layout design of an inverter by ***mosfet_low*** transistors, given by the schematic in figure 80. Width and height of the cell is respectively 0.416 μm and 1.382 μm.

NAND

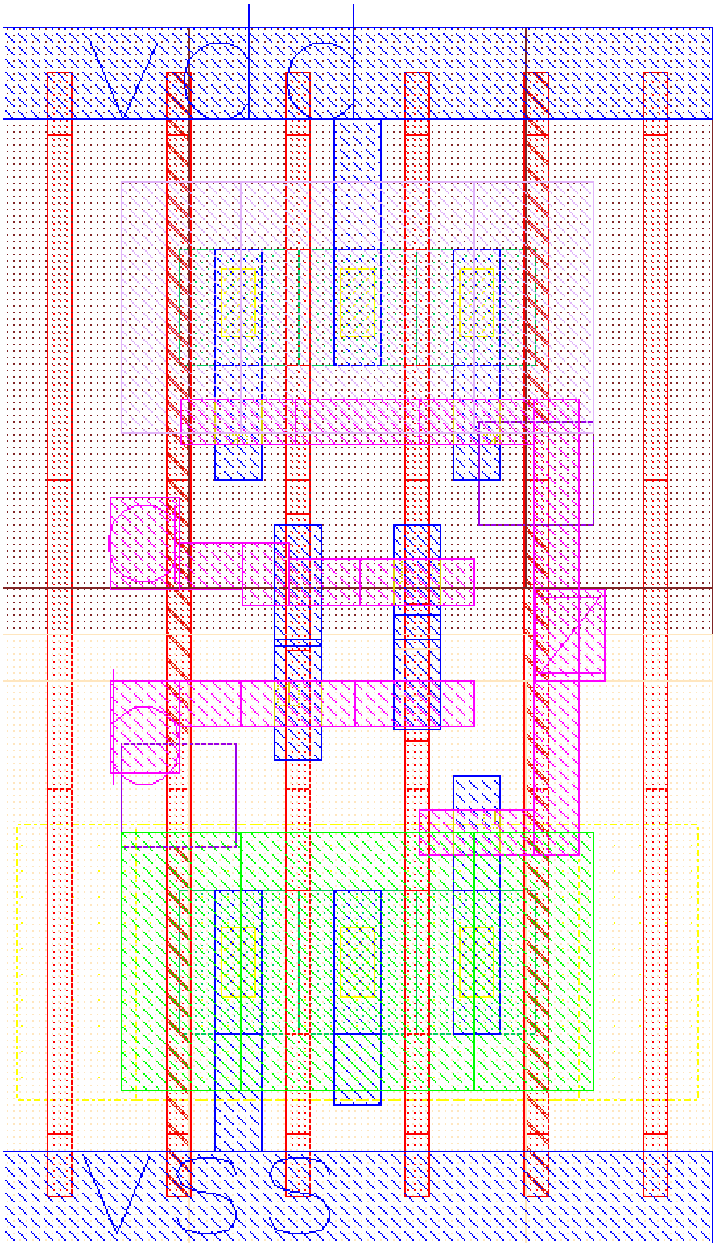


Figure 118: Layout design of a NAND-gate by ***mosfet_low*** transistors, given by the schematic in figure 82. Width and height of the cell is respectively 0.620 μm and 1.062 μm.

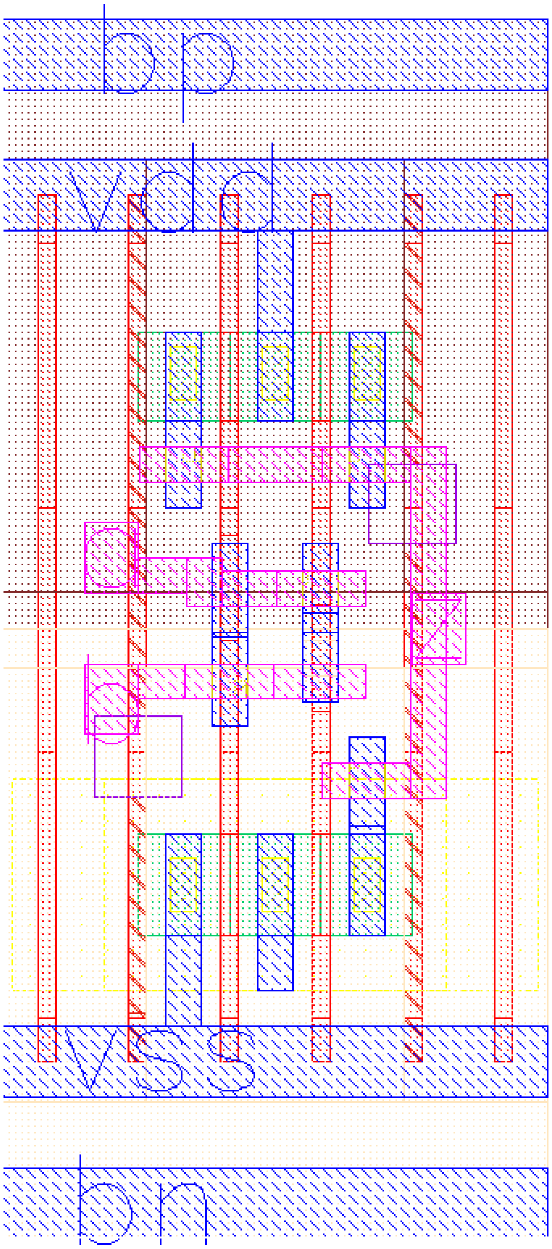


Figure 119: Layout design of a NAND-gate by ***mosfet_low*** transistors, given by the schematic in figure 83. Width and height of the cell is respectively 0.620 μm and 1.382 μm.

NOR

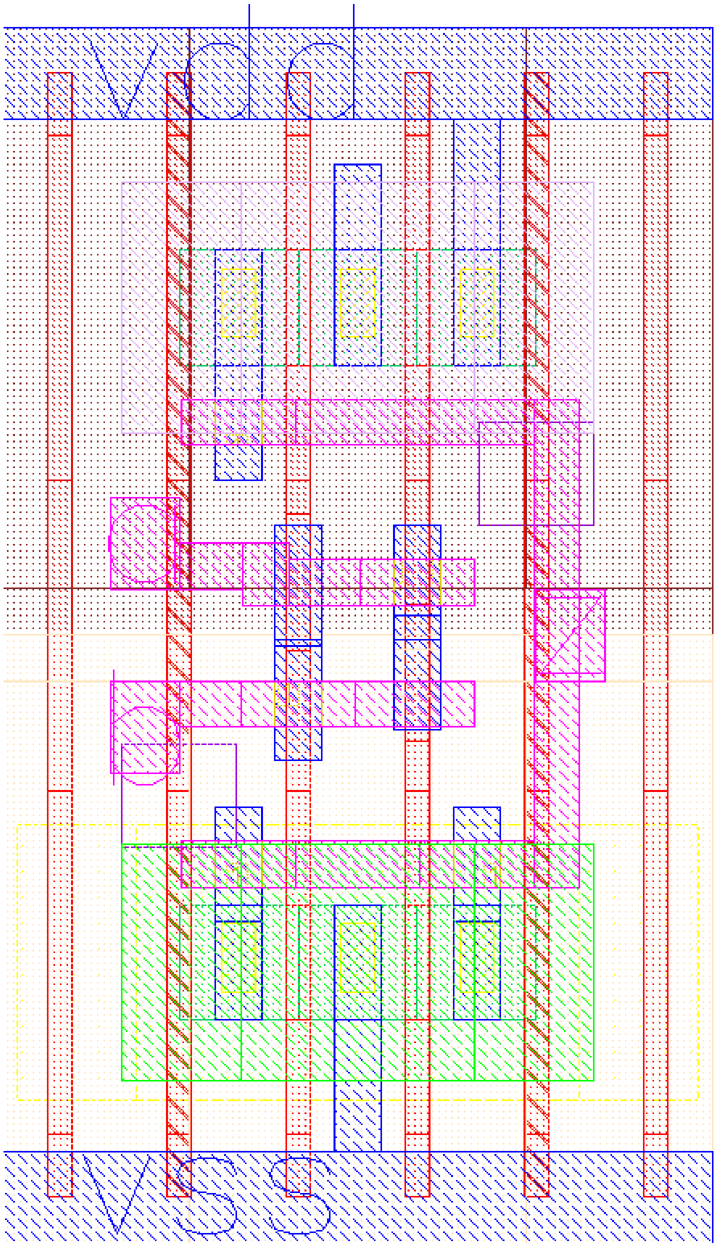


Figure 120: Layout design of a NOR-gate by ***mosfet_low*** transistors, given by the schematic in figure 85. Width and height of the cell is respectively 0.620 μm and 1.062 μm.

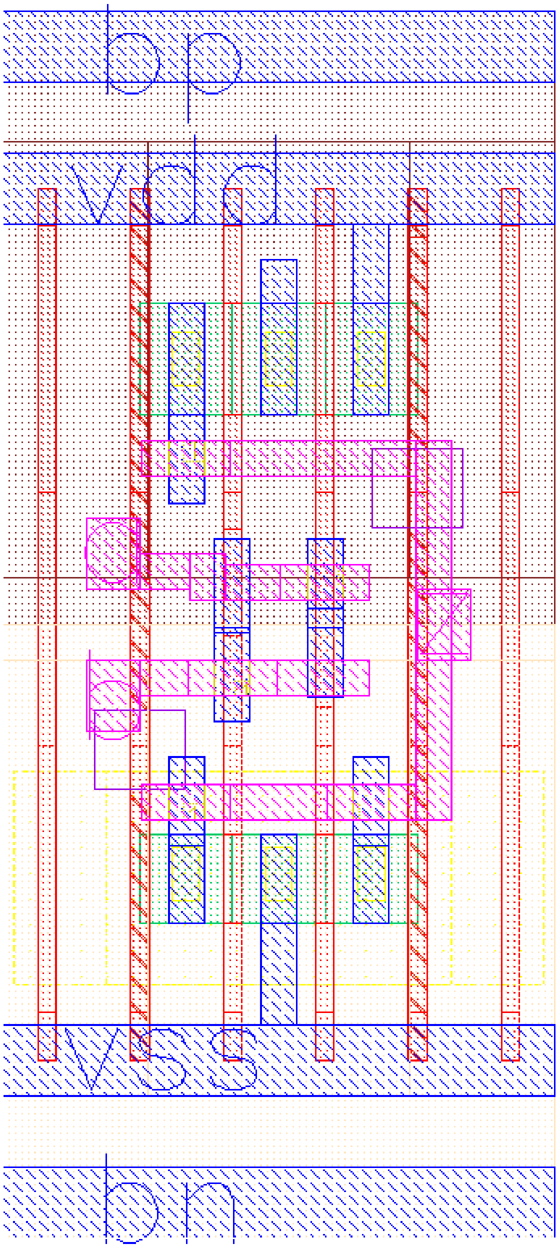


Figure 121: Layout design of a NOR-gate by ***mosfet_low*** transistors, given by the schematic in figure 86. Width and height of the cell is respectively 0.620 μm and 1.382 μm.

XNOR

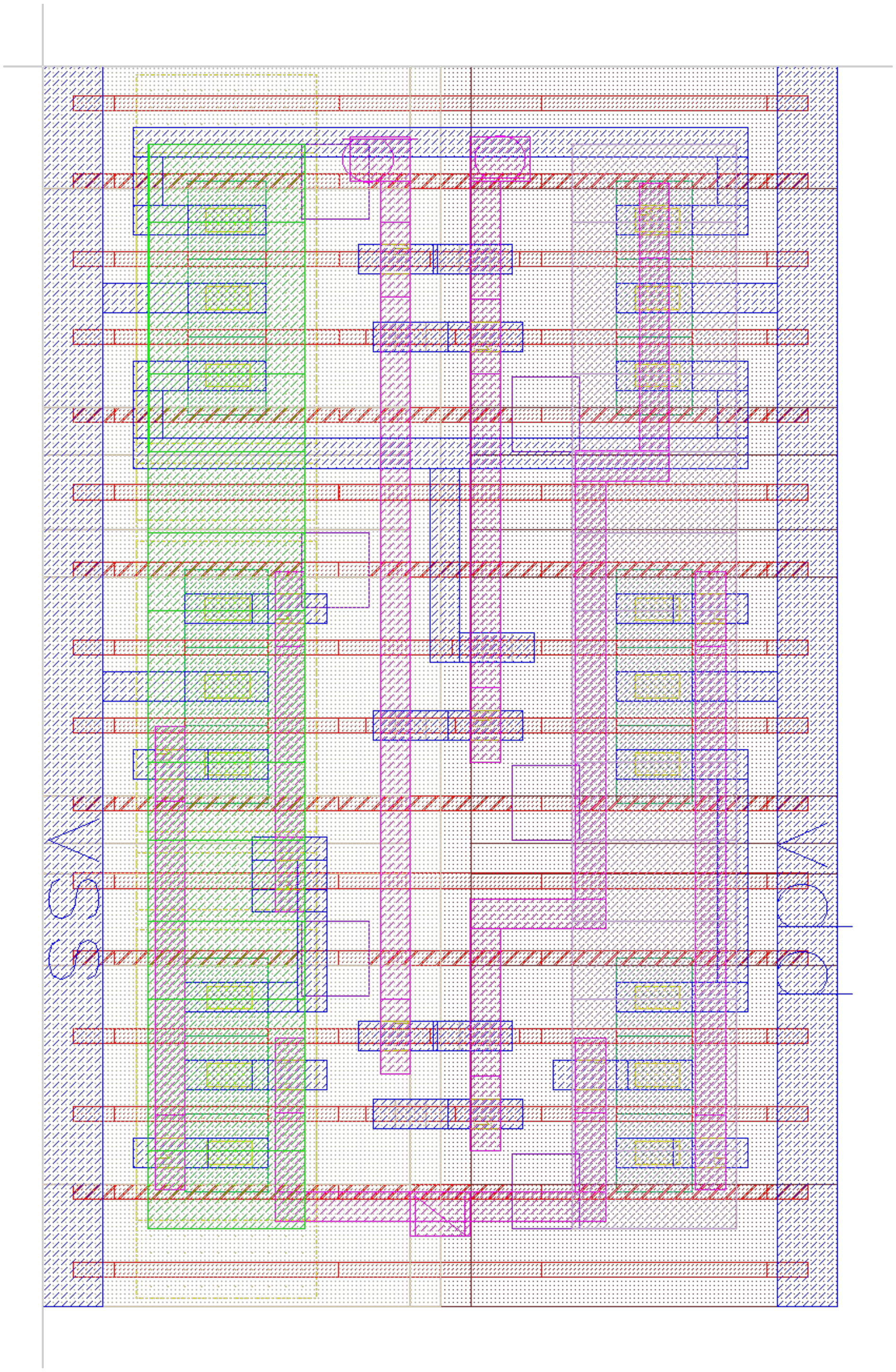


Figure 122: Layout design of a XNOR-gate by ***mosfet_low*** transistors, given by the schematic in figure 88. Width and height of the cell is respectively 1.560 μm and 1.062 μm.

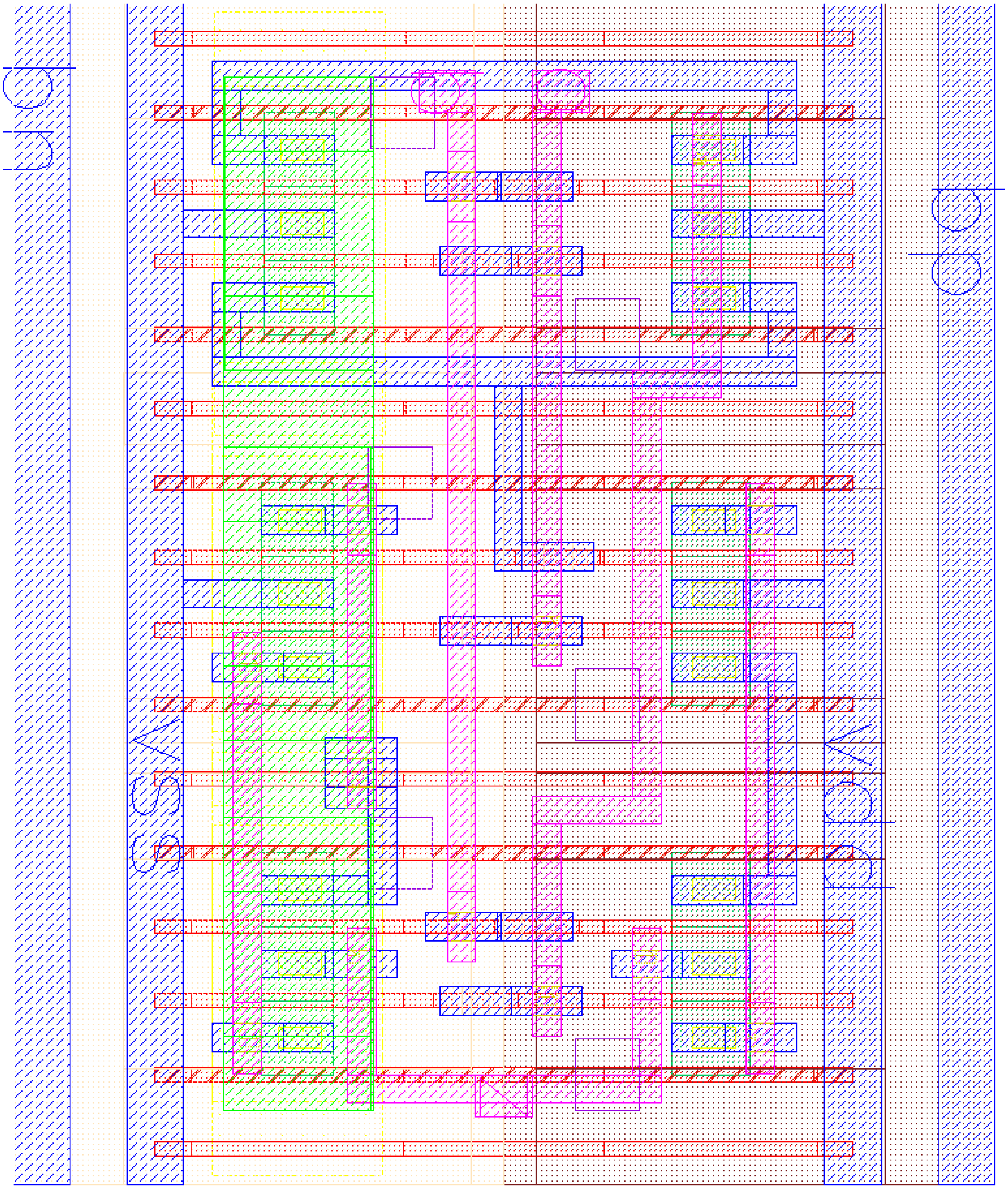


Figure 123: Layout design of a XNOR-gate by ***mosfet_low*** transistors, given by the schematic in figure 89. Width and height of the cell is respectively 1.560 μm and 1.382 μm.

XOR

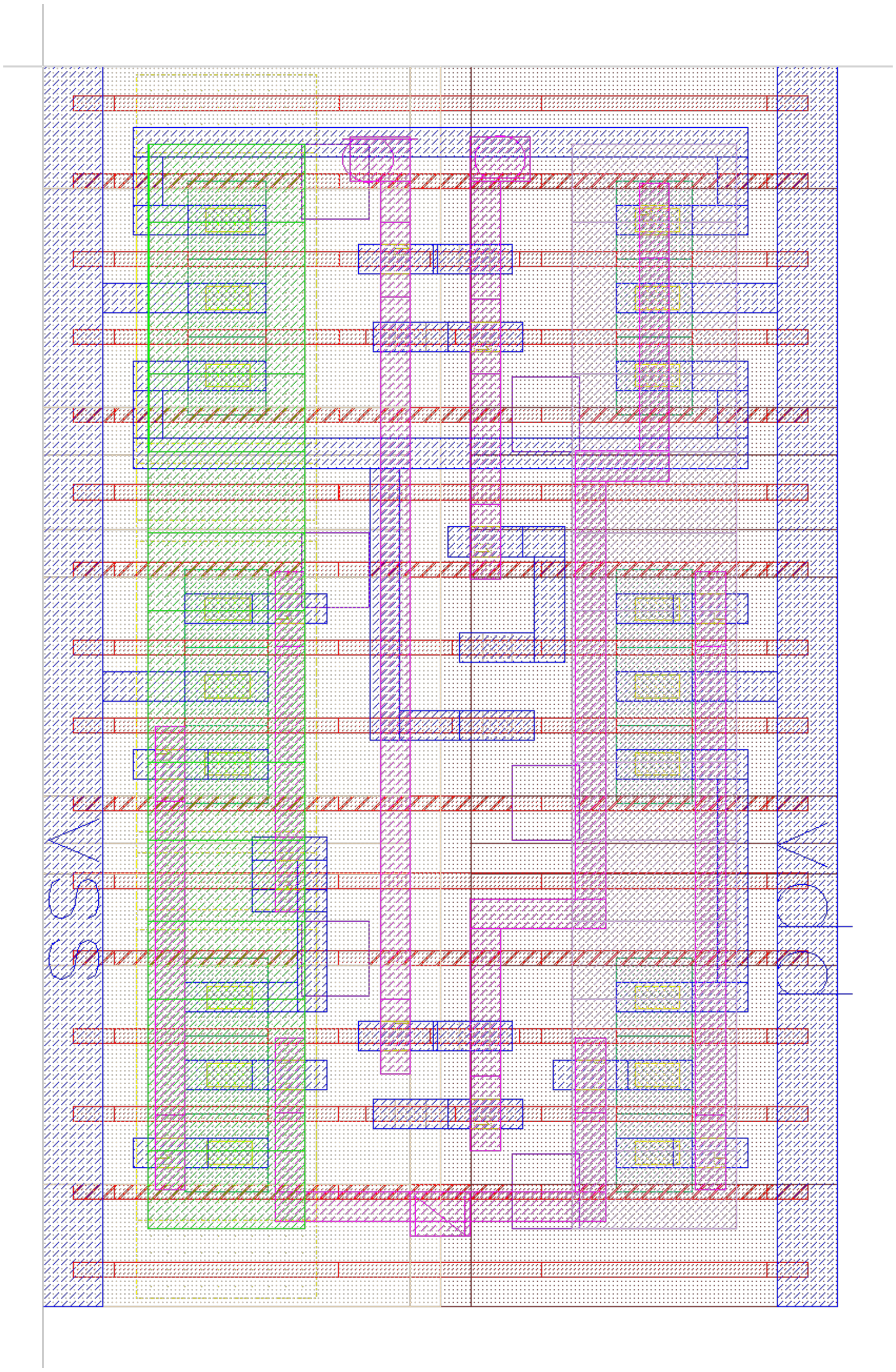


Figure 124: Layout design of a XOR-gate by ***mosfet_low*** transistors, given by the schematic in figure 90. Width and height of the cell is respectively 1.560 μm and 1.062 μm.

Minority-3

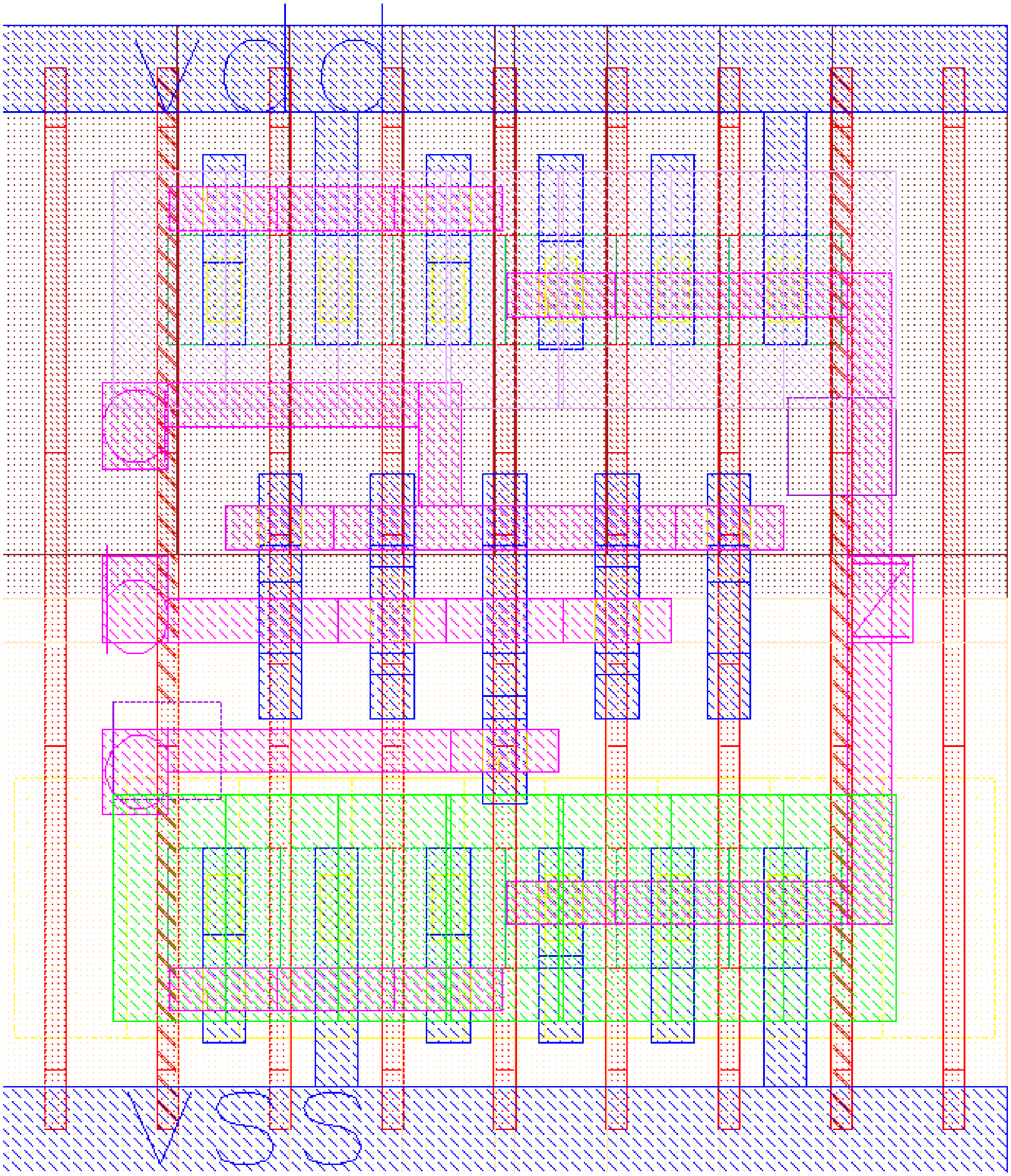


Figure 125: Layout design of a Minority-3 gate by ***mosfet_low*** transistors, given by the schematic in figure 92. Width and height of the cell is respectively 0.832 μm and 1.062 μm.

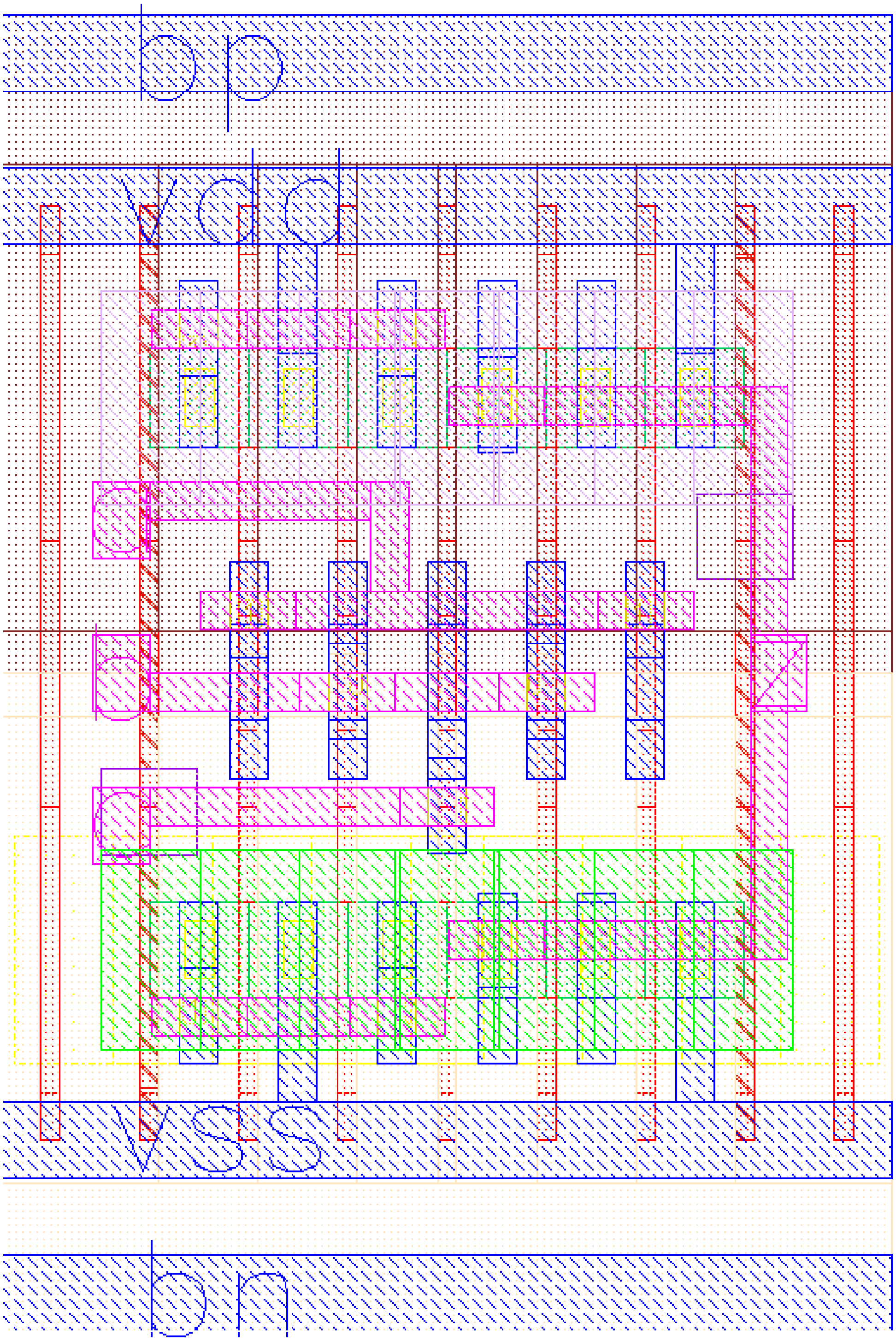


Figure 126: Layout design of a Minority-3 gate by ***mosfet_low*** transistors, given by the schematic in figure 93. Width and height of the cell is respectively 0.832 μm and 1.382 μm.

Half-Adder

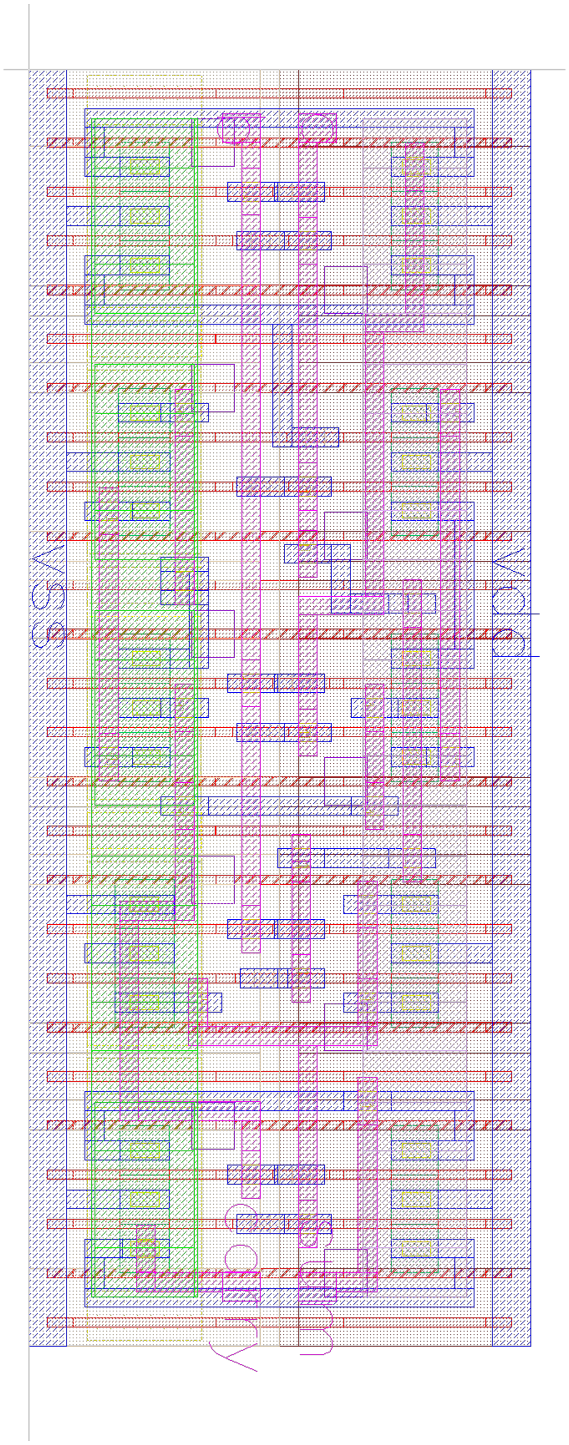


Figure 127: Layout design of a Half-Adder gate by ***mosfet_low*** transistors, given by the schematic in figure 95. Width and height of the cell is respectively 2.600 μm and 1.062 μm.

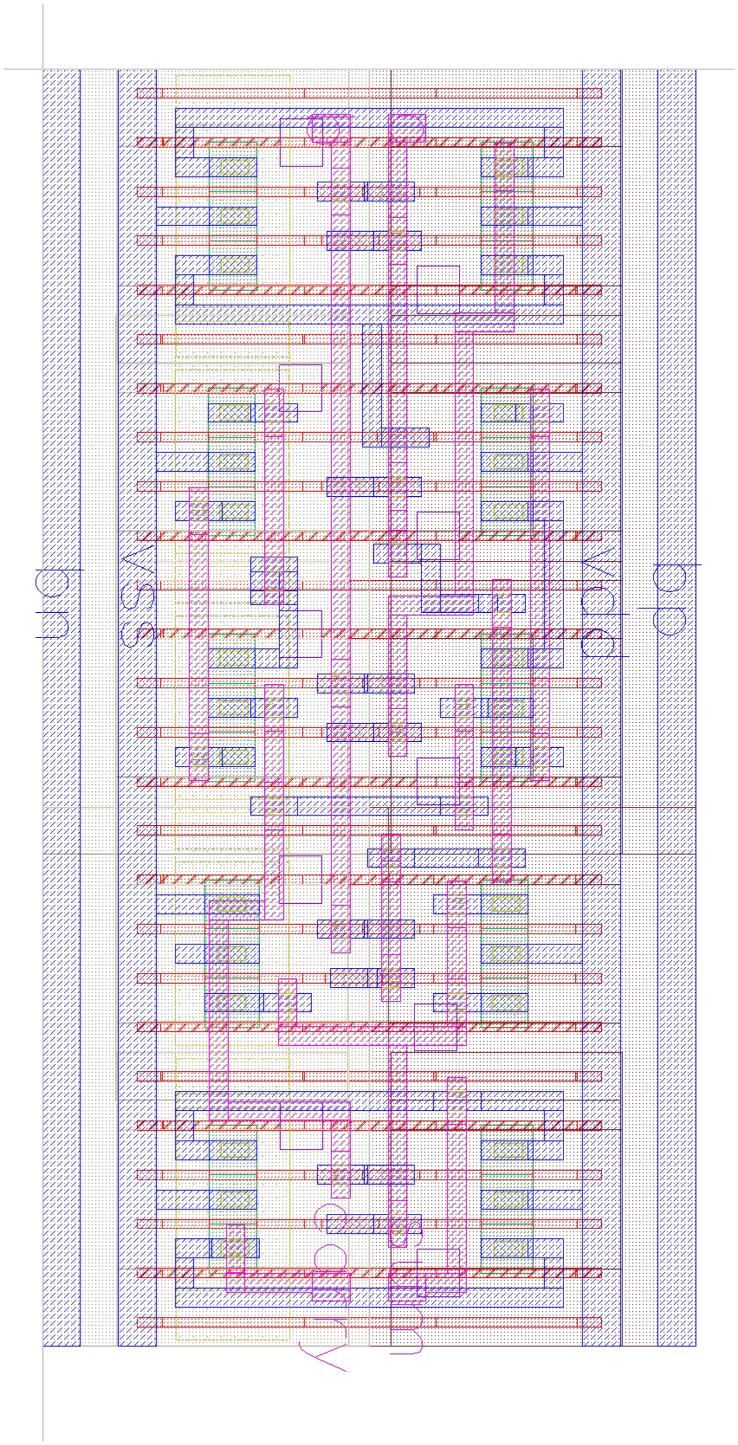


Figure 128: Layout design of a Half-Adder gate by ***mosfet_low*** transistors, given by the schematic in figure 96. Width and height of the cell is respectively 2.600 μm and 1.382 μm.

Full-Adder

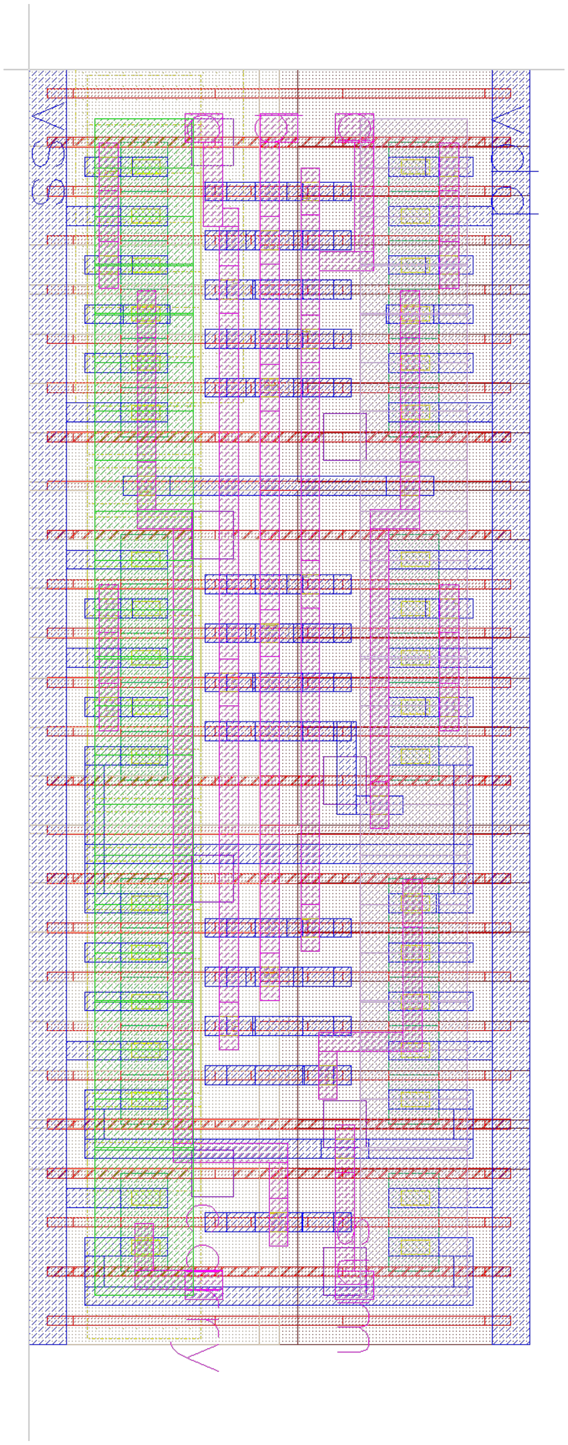


Figure 129: Layout design of a standard Full-Adder gate by ***mosfet_low*** transistors, given by the schematic in figure 97. Width and height of the cell is respectively 2.600 μm and 1.062 μm.

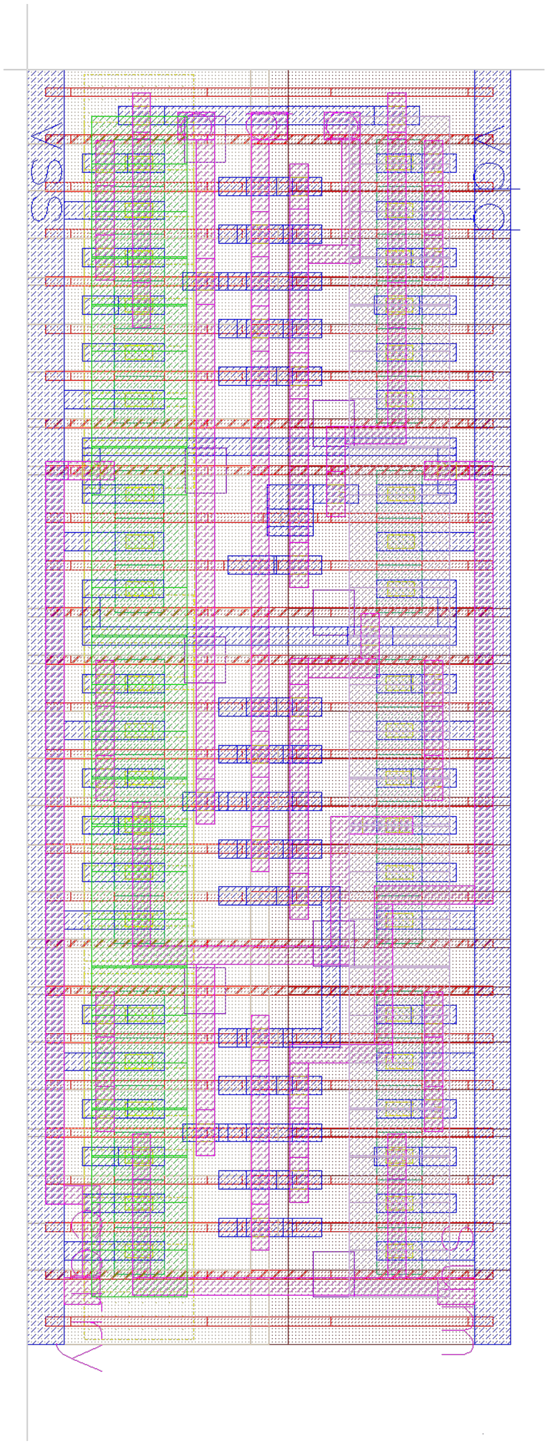


Figure 130: Layout design of a Minority-3 based Full-Adder gate by ***mosfet_low*** transistors, given by the schematic in figure 99. Width and height of the cell is respectively 2.704 μm and 1.062 μm.

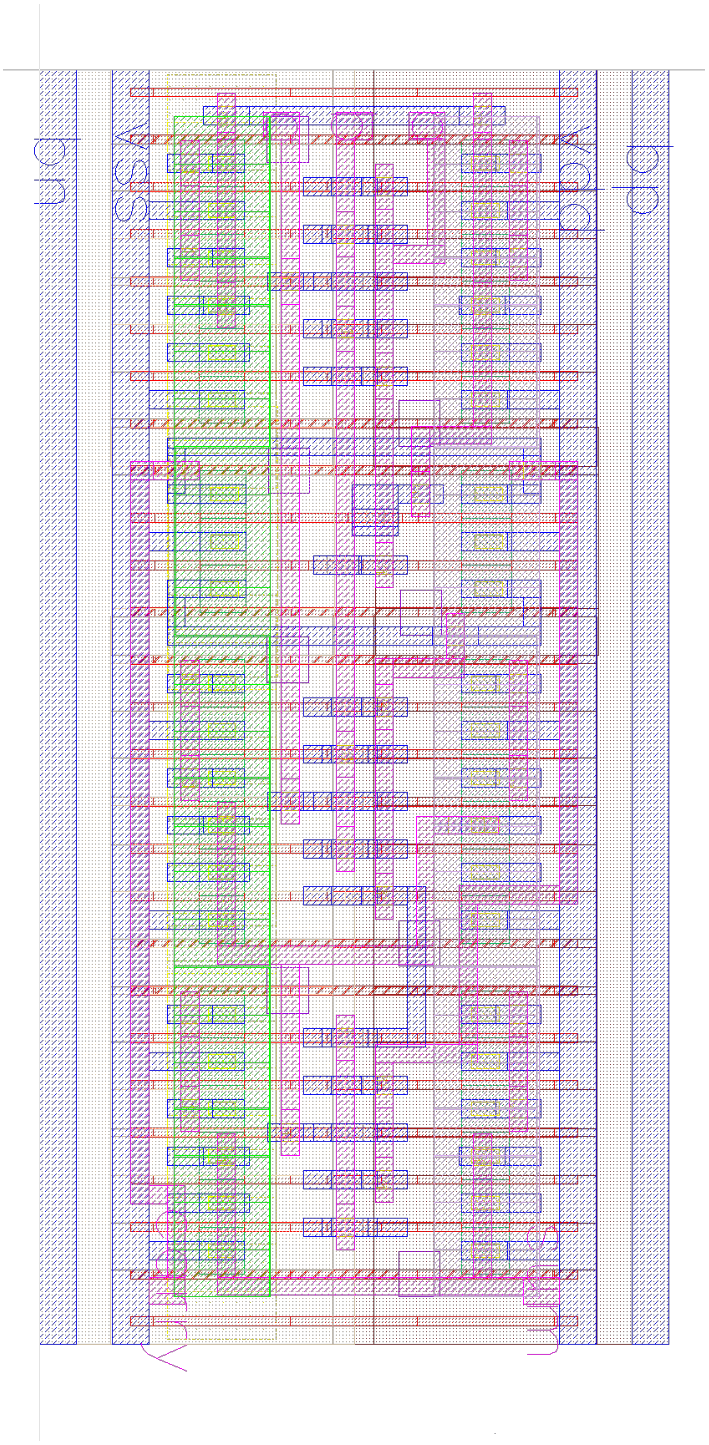


Figure 131: Layout design of a Minority-3 based Full-Adder gate by ***mosfet_low*** transistors, given by the schematic in figure 100. Width and height of the cell is respectively 2.704 μm and 1.382 μm.

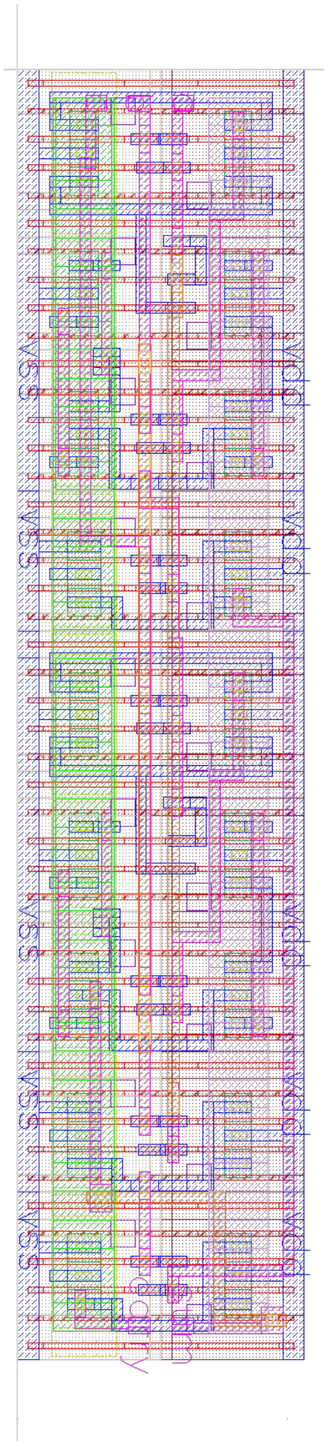


Figure 132: Layout design of a NAND based Full-Adder gate by ***mosfet_low*** transistors, given by the schematic in figure 101. Width and height of the cell is respectively 4.680 μm and 1.062 μm.

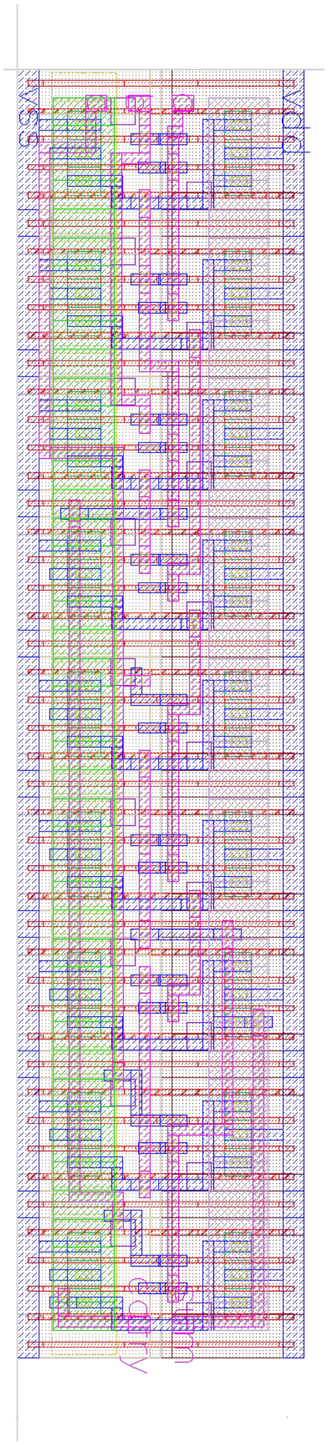


Figure 133: Layout design of a NAND based Full-Adder gate by ***mosfet_low*** transistors, given by the schematic in figure 102. Width and height of the cell is respectively 4.680 μm and 1.062 μm.

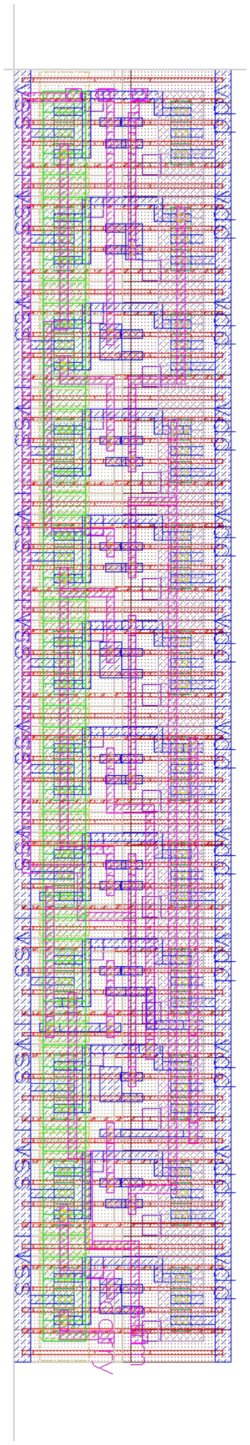


Figure 134: Layout design of a NOR based Full-Adder gate by ***mosfet_low*** transistors, given by the schematic in figure 103. Width and height of the cell is respectively 6.240 μm and 1.062 μm.

8-bit Ripple-Carry Adder

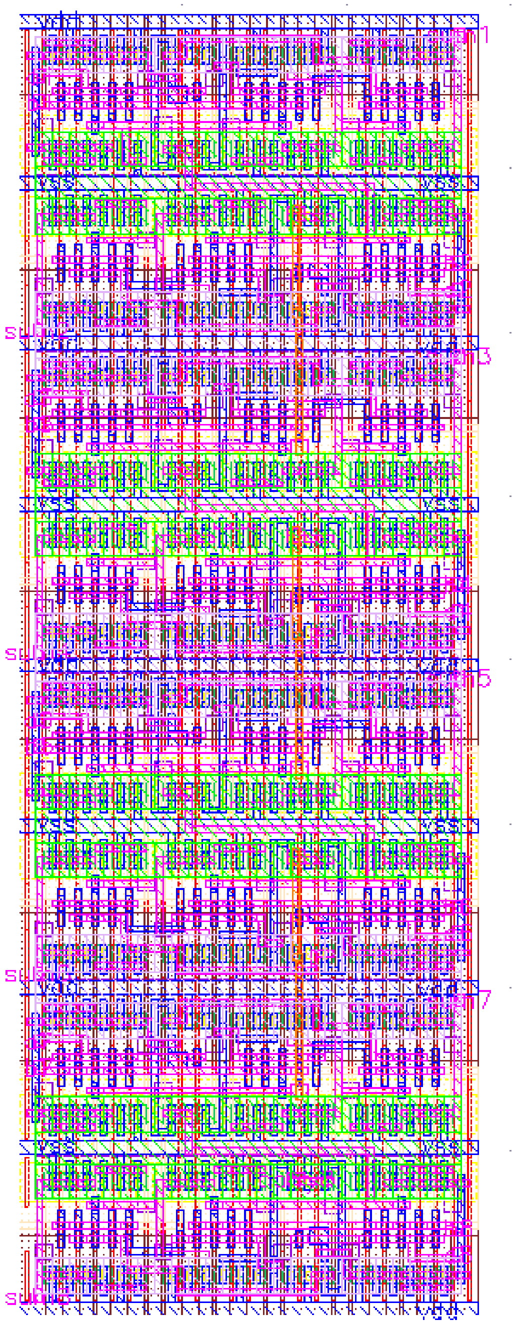


Figure 135: Layout design of an 8-bit Ripple-Carry Adder by Minority-3 based Full-Adder gates. The Full-Adders are built by ***mosfet_low*** transistors, given by the schematic in figure 99. Width and height of the cell is respectively 2.704 μm and 7.936 μm.

12-bit Ripple-Carry Adder

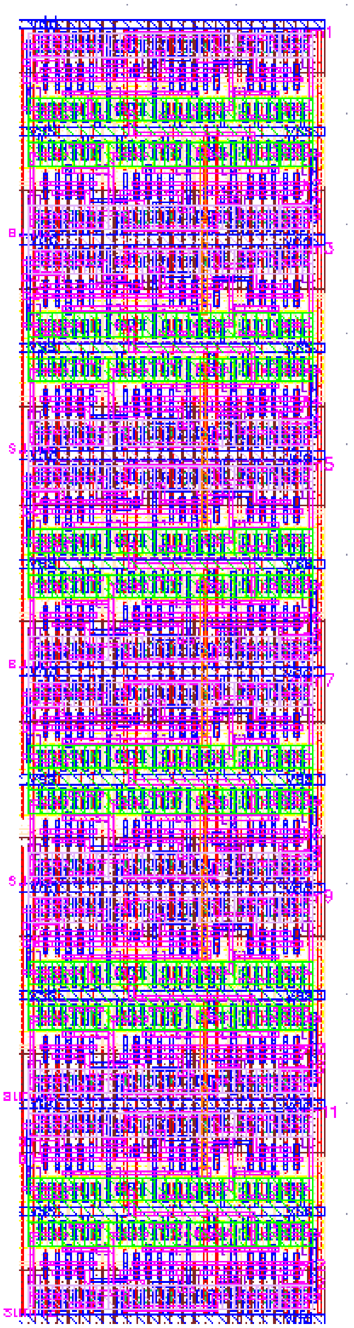


Figure 136: Layout design of a 12-bit Ripple-Carry Adder by Minority-3 based Full-Adder gates. The Full-Adders are built by ***mosfet_low*** transistors, given by the schematic in figure 99. Width and height of the cell is respectively 2.704 μm and 11.864 μm.

13-bit Ripple-Carry Adder

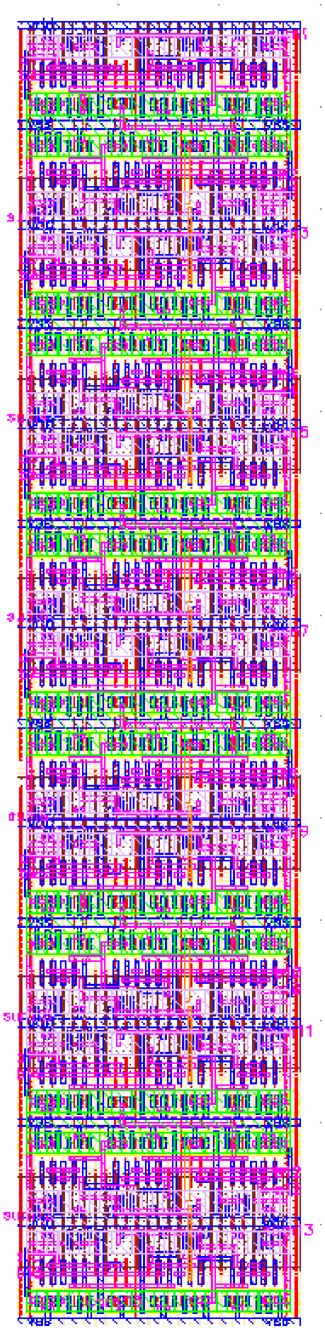


Figure 137: Layout design of a 13-bit Ripple-Carry Adder by Minority-3 based Full-Adder gates. The Full-Adders are built by ***mosfet_low*** transistors, given by the schematic in figure 99. Width and height of the cell is respectively 2.704 μm and 12.846 μm.

14-bit Ripple-Carry Adder

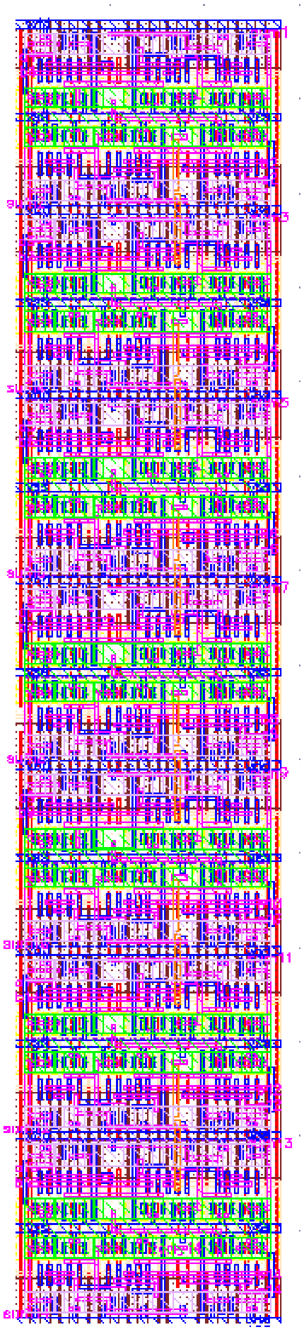


Figure 138: Layout design of a 14-bit Ripple-Carry Adder by Minority-3 based Full-Adder gates. The Full-Adders are built by ***mosfet_low*** transistors, given by the schematic in figure 99. Width and height of the cell is respectively 2.704 μm and 13.828 μm.

15-bit Ripple-Carry Adder

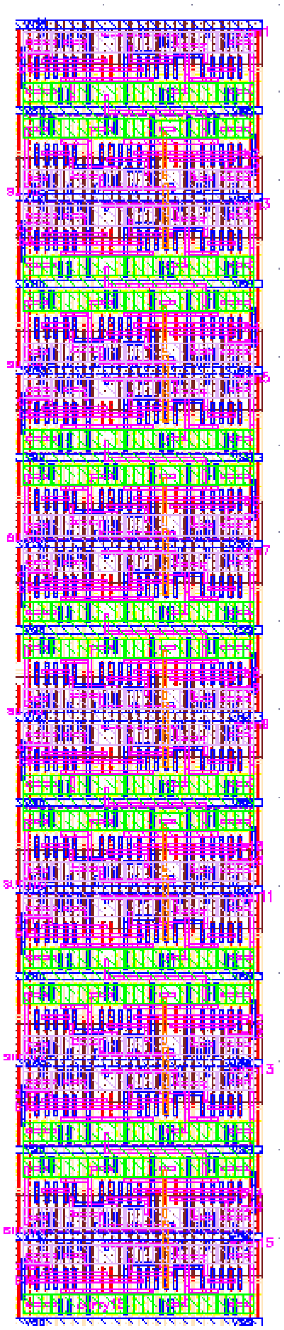


Figure 139: Layout design of a 15-bit Ripple-Carry Adder by Minority-3 based Full-Adder gates. The Full-Adders are built by ***mosfet_low*** transistors, given by the schematic in figure 99. Width and height of the cell is respectively 2.704 μm and 14.810 μm.

16-bit Ripple-Carry Adder

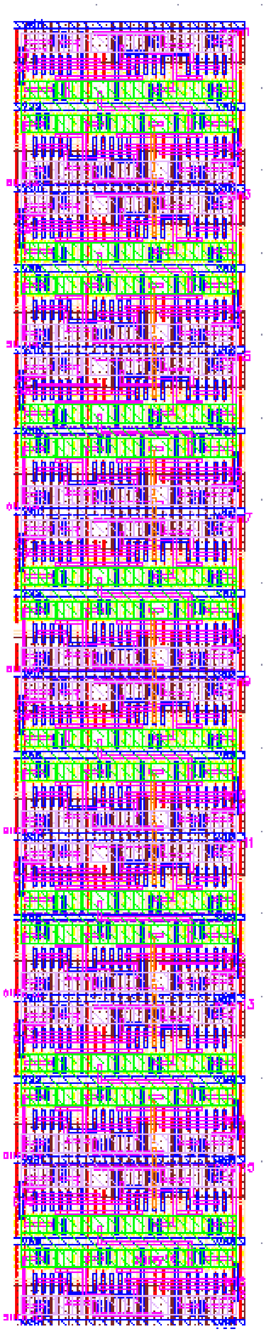


Figure 140: Layout design of a 16-bit Ripple-Carry Adder by Minority-3 based Full-Adder gates. The Full-Adders are built by ***mosfet_low*** transistors, given by the schematic in figure 99. Width and height of the cell is respectively 2.704 μm and 15.792 μm.

D Flip-Flop

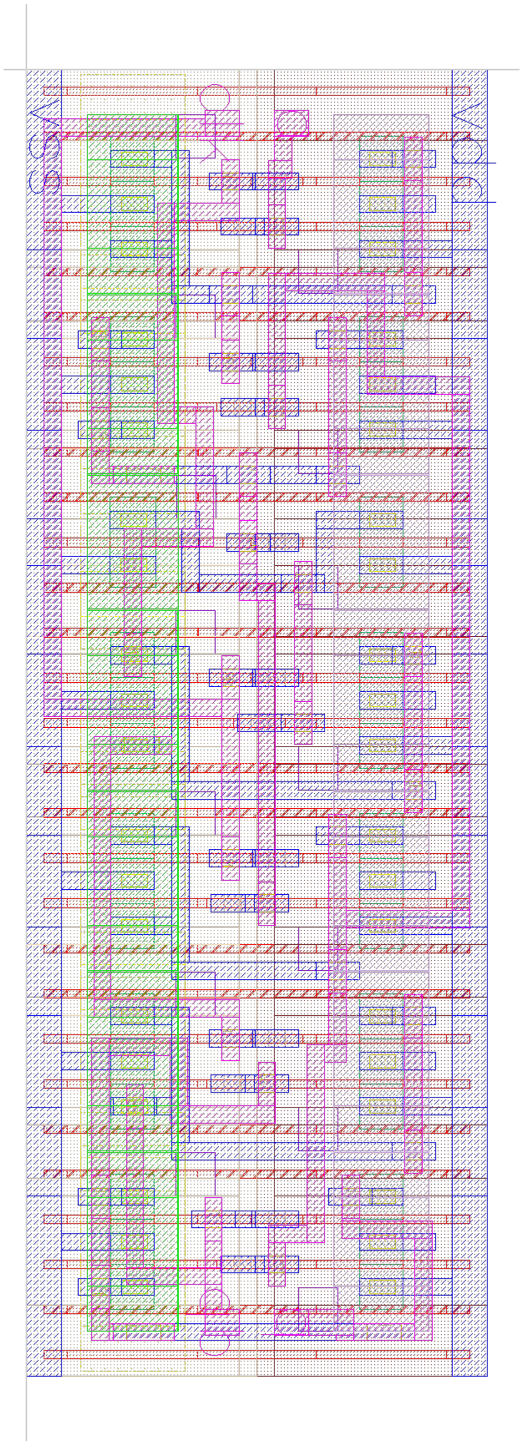


Figure 141: Layout design of a D Flip-Flop gate by ***mosfet_low*** transistors, given by the schematic in figure 112. Width and height of the cell is respectively 2.912 μm and 1.062 μm.

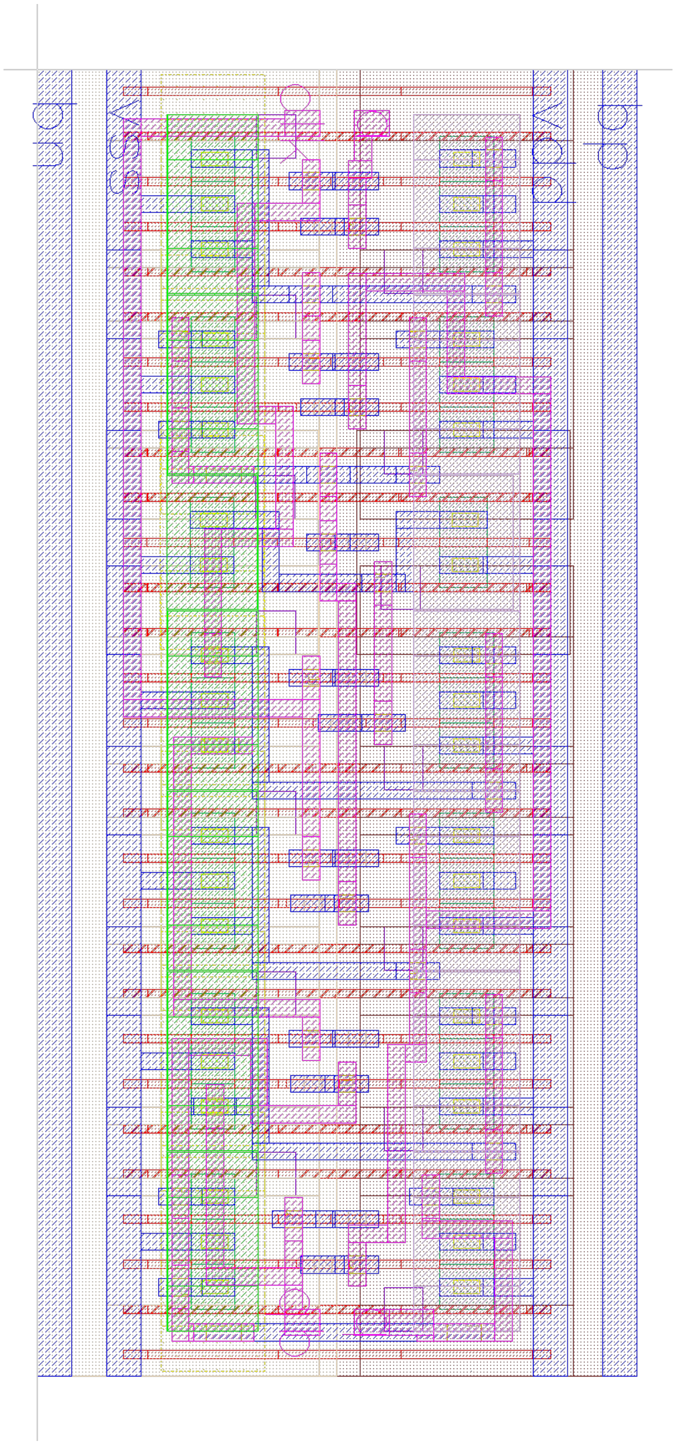


Figure 142: Layout design of a D Flip-Flop gate by ***mosfet_low*** transistors, given by the schematic in figure 113. Width and height of the cell is respectively 2.912 μm and 1.382 μm.