Chapter 5

CMOS Logic Building Blocks

In this chapter we discuss structures, layout and transient properties of basic CMOS logic building blocks. These blocks come into two groups referred to as **gate logic** and **switch logic**.

We introduce first layout design rules, next we consider a serial and parallel connection of transistors, then we study NOR and NAND gates and their generalization known as a CMOS **composite gate**.

Next we discuss the concept of the switch logic and basic representative of this approach, namely, a transmission gate and a week multiplexer introduced already in sec. 3.4.

Finally we consider two examples which combine the concepts of the gate and switch logic, namely, the tri-state inverter and the Exclusive-OR (XOR) gate.

5.1 Layout Design Rules

Layout design rules specify dimensions of all objects created on the silicon surface and separation between them. Traditionally two sets of rules are used:

- **Micron rules**, in which all circuit layout dimensions such as minimum feature sizes and their separations are stated in terms of their absolute dimensions in micrometers. These rules specify the manufacturing limits.
- **Lambda rules**, in which all layout dimensions are specified in multiples of a single parameter λ. This parameter is related to the minimum feature available in a given technology. For the AMI0.5 technology that we use as a benchmark it can be assumed that the parameter λ is:

  $$\lambda = 0.25\mu\mathrm{m}$$

  The λ rules are also called **scalable rules** and can be used across different technologies by a simple re-definition of the λ value.

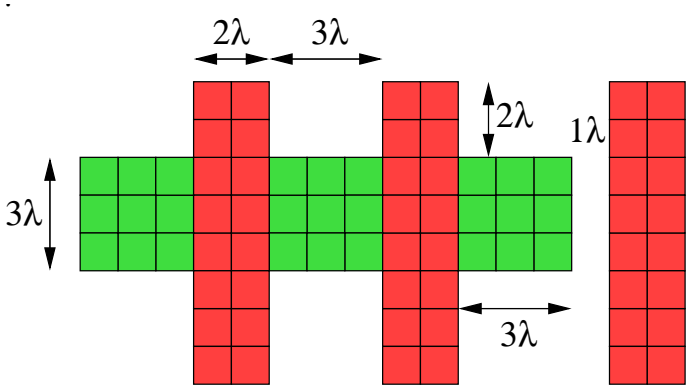
  The price of simplicity and uniformity is, however, that the designs generated using the λ rules are usually larger than those based on the micron rules which are closely tied to a given technology.

One of the best sources of the current design rules is available from an American MOSIS company offering small-volume production services for VLSI circuit development. The design rules are available from

`http://www.mosis.com/Technical/Designrules/scmos/`

Using a CAD package like Mentor Graphics, a set of the design rules is incorporated into the specification of the technology.

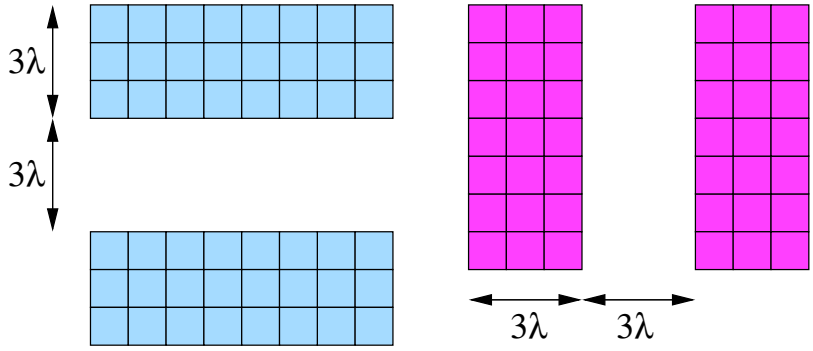
Basic design rules associated with transistors, that is, with polysilicon and active (diffusion) areas are illustrated in Figure 5.1 both in graphical and tabular forms.



Rule	λ
Minimum poly width	2
Minimum active width	3
Minimum poly spacing	3
Minimum poly extension	2
Minimum spacing between poly and active	1
Minimum active extension	3

Figure 5.1: A selection of basic λ design rules for polysilicon and active (diffusion) areas.

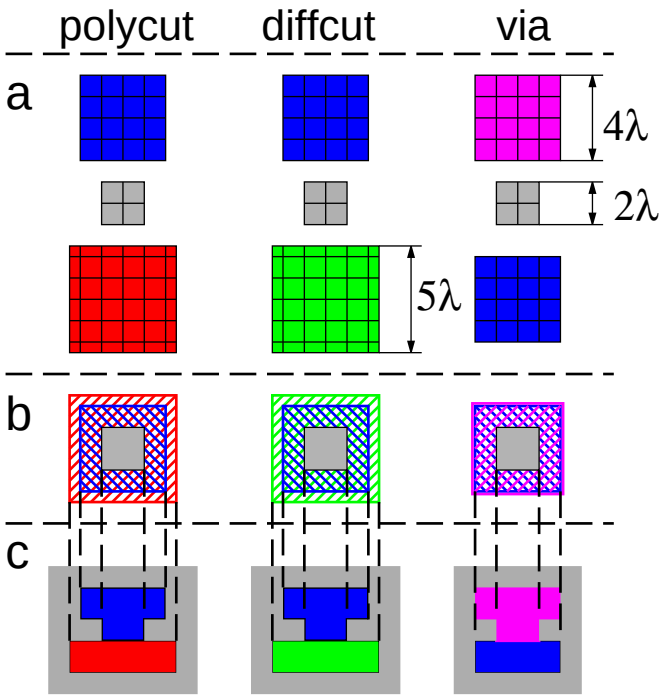
The design rules for metal 1 and metal 2 paths are illustrated in Figure 5.2.



Rule	λ
Minimum width .	3
Minimum spacing	3

Figure 5.2: Metal design rules.

The design rules for polysilicon and diffusion contacts and the metal 1 to metal 2 via are illustrated in Figure 5.3. In addition, simplified cross-sectional views of the contacts showing inter layer connection are also presented in Figure 5.3.



Rule	λ
Exact contact size	2 × 2
Minimum poly overlap	1.5
Minimum metal overlap	1
Minimum contact spacing	2
Minimum spacing to gate of a transistor	2

Figure 5.3: The λ design rules for three basic contacts: polycut (poly-to-metal1), diffcut (diffusion-to-metal 1), and via (metal1-to-metal2). **a**. Constituent masks and related design rules. **b**. Layout view of the contacts. **c**. Cross-sections of the contact silicon structures

It can be observed that the **cut**, which is filled with metal to make an electrical connection between layers, is $2\lambda \times 2\lambda$. In addition there should be at least 1 or 1.5 λ of the overlapping materials. In the cross-sectional views, it can be noticed that the contacts are surrounded with insulating silicon dioxide.

Finally there are common-sense **crossing rules** illustrated in Figure 5.4. This time, for simplicity, we use the stick diagram convention.

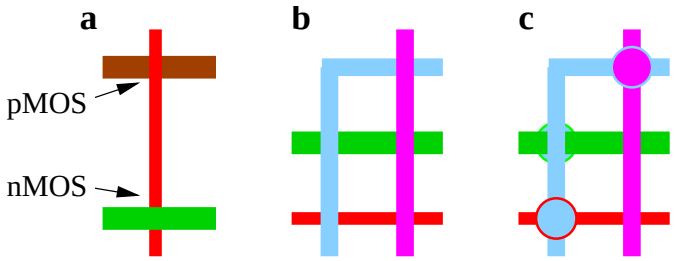


Figure 5.4: Crossing rules

The crossing rules can be summarized as follows:

a When a polysilicon crosses the active (diffusion) paths, transistors are made.

b Metal 1 and metal 2 paths can go over polysilicon, diffusion, and each other without any connections made.

c In order to connect metal 1 to polysilicon, diffusion, or metal 2, a contact must be made.

Metal 2 can be connected **only to metal 1** through the via contact.

5.2 Principles of a good circuit layout at the cell level

Designing a low level library cell, we have to obey the following rules of a good layout presented in Figure 5.5

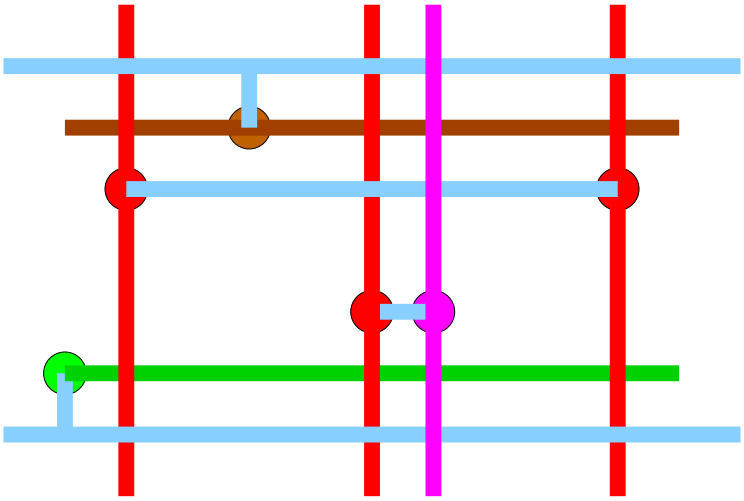


Figure 5.5: Illustration of the good layout principles at the cell level

- Two horizontal, preferably continuous (!) strips of p-type and n-type **diffusion** in their respective wells/substrates,
- Vertical **polysilicon** paths to form pMOS and nMOS transistors and to implement short, local, connections,
- Primarily horizontal **metal1** paths to implement V_{DD}, GND, and other ”longer” connections in the circuit,
- Primarily vertical **metal2** paths to implement long inter-gate connections
- Remember about the $\left\{\begin{matrix}\mathrm{n}\\ \mathrm{p}\end{matrix}\right\}$ well/substrate contacts to $\left\{\begin{matrix}V_{DD}\\ \mathrm{GND}\end{matrix}\right\}$. The diffusion area of these contacts should be adjacent to the “native” diffusion area in the given well.

5.3 Connecting MOS transistors

5.3.1 Serial connection of MOS transistors

Let us consider a serial connection of two nMOS transistors connected between GND and an output Y terminal.

According to the rules of the good layout the stick diagram and the layout of such a serial connection should look as in Figure 5.6.

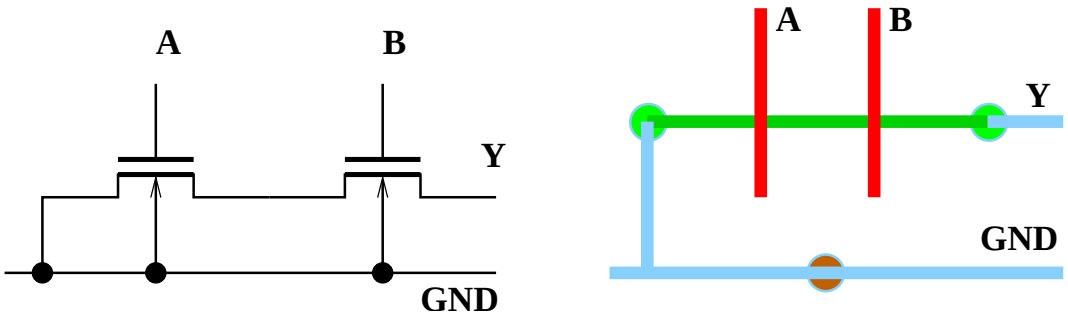


Figure 5.6: A schematic and stick diagram of a serial connection of two nMOS transistors.

In the schematic transistors are represented as four-terminal devices, with the substrate/well connection being explicitly shown. In the stick diagram transistors look as three-terminal devices (the green source and drain, and the red gate). The pWell contact is illustrated as the blue circuit filled in with brown.

An equivalent layout in the AMI0.5 technology generated using the Mentor Graphics IC Station is shown in Figure 5.7.
It seems that in the layout the most prominent feature is the 10λ of the GND metal 1 path and a relatively large pWell-to-GND contact. The length of two transistors is 2λ each and their width is 5λ to match the size of the n-diffusion-to-metal1 contact. Note also a 3λ separation between the gates of two transistors. The n diffusion (green) from a

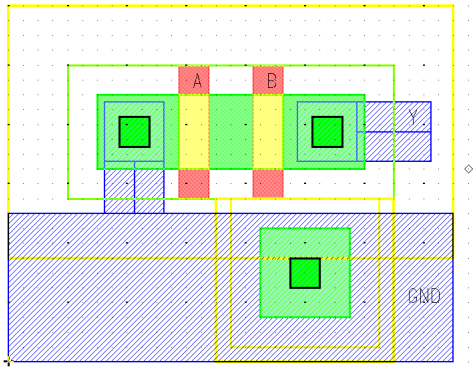


Figure 5.7: A circuit layout of a serial connection of two nMOS transistors in the AMI0.5 technology

stick diagram is mapped into a green mask for the **active** region surrounded by the green **n+ mask**.

Finally, the Ample script to generate the above layout is as follows:

```
// dof ser.do
{local x , y ;
 x = 0 ; y = 10 ;
 $add_point_device("mos",@block,[],[x+6, y+3],[["s", "cggc"],["l",""],\
    ["w", "5"], ["t", "nmos4"], ["inst", void], ["b", "[0]"],\
    ["glabel", void], ["sds", "[[@shift]]"]], @placed);
 $add_shape([[x, y-10], [x+30, y]], "METAL1", @internal, @nokeep) ;
 $add_cell("$ADK/technology/ic/ami05_via/pwell_contact", [x+20, y-4]);
 x = x+8.5 ;
 $add_path([[x,y],[x,y+3.5]],"METAL1",@internal,4,@center,...);
 x= x+15 ; y = y+5.5;
 $add_path([[x+5,y],[x,y]],"METAL1",@internal,4,@center,...);
}
```

The function `$add_point_device` generates two serially connected transistors and related contacts to metal1. Such a connecton is specified by the argument `"cggc"` (contact-gate-gate-contact).

5.3.2 Parallel connection of transistors

A serial connection of transistors implements the AND function in the **switch logic**: a connection is made when both transistors are “ON”. Similarly, a parallel connection of transistors implements the OR function in the **switch logic**: a connection is made when either or both transistors are “ON”.

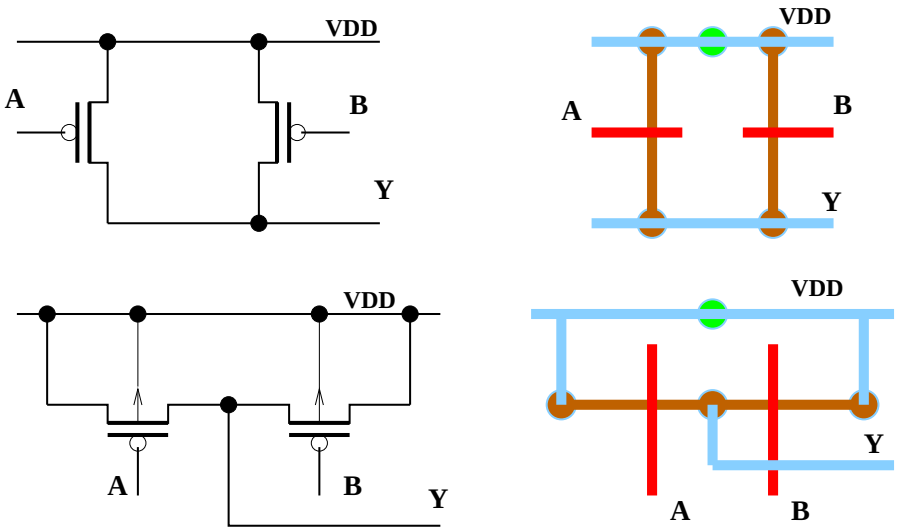


Figure 5.8: Two possible schematics and the equivalent stick diagrams of a parallel connection of two pMOS transistors.

As an example let us consider a parallel connection of pMOS transistors as in Figure 5.8. Note that the “vertical” transistors have been re-arranged horizontally, which has resulted in continuous horizontal diffusion area with vertical polysilicon gates. The transistors look as if connected in a serial way.

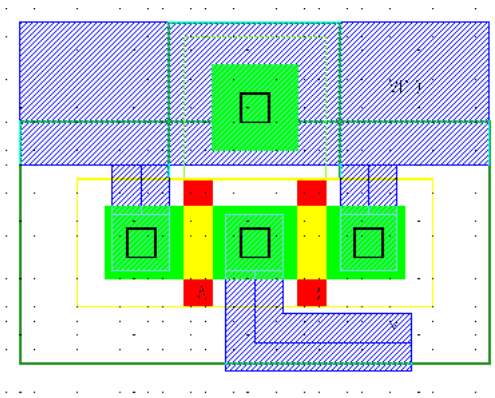


Figure 5.9: A possible layout of a parallel connection of two pMOS transistors generated with the IC Station.

This time the Ample function generating the pMOS transistors have the following form:

```
$add_point_device("mos",@block,[],[x,y],[["s", "cgcgc"],...
```

which generates the “contact-gate-contact-gate-contact” sequence needed to form a parallel connection of two transistors. The p diffusion (brown) from a stick diagram is mapped into a green mask for the **active** region surrounded by the yellow **p+ mask**.

5.4 A two-input NOR gate

For the purpose of designing a layout of a two-input NOR gate we expand the standard logic equation for the NOR gate, namely,

$$y = \overline{a + b}$$

into two equivalent statements, one for generation of zeros, the other for generation of ones:

ones when	$y_p = \bar{a} \cdot \bar{b}$	is true	(serial connection of pMOSs)
zeros when	$y_n = a + b$	is true	(parallel connection of nMOSs)

These equations are now used to implement relevant sections of the switching circuit which results in the schematic and a stick diagram as in Figure 5.10.

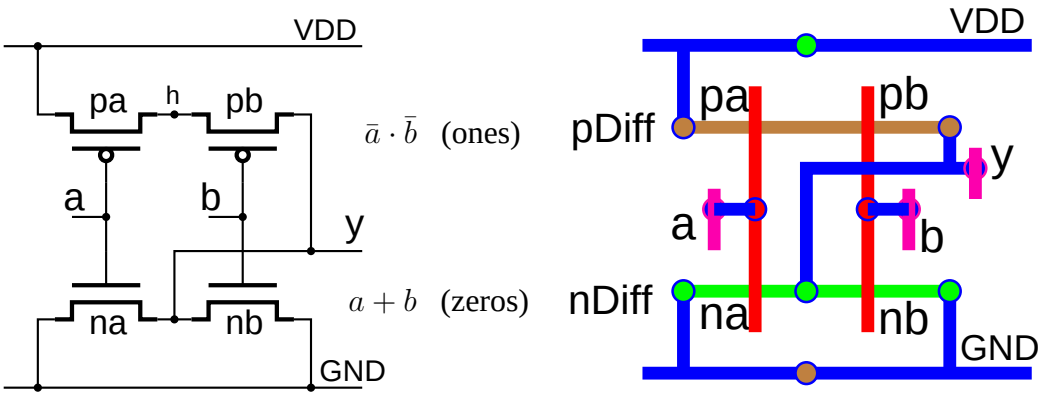


Figure 5.10: Schematic and a related stick diagram of a 2-input NOR gate

Note a parallel connection of the nMOS transistors implementing the logic OR function, and a serial connection of the pMOS transistors implementing the logic AND function. Compare the stick diagram representation of the serial and parallel connection of the transistors with those represented in Figures 5.6 and 5.8.

A new element in the stick diagram is an implementation of an input/output port. We assume that the external signals of a cell should be available in the metal 2 layer in order to facilitate the vertical interconnections of such signals. Therefore, in general, we need a via between the metal 1 and metal 2 layers, and, if required, a contact between polysilicon and metal 1 layers as it is the case with the input ports '**a**' and '**b**'.

The circuit layout of a 2-input NOR gate is shown in Figures 5.11.

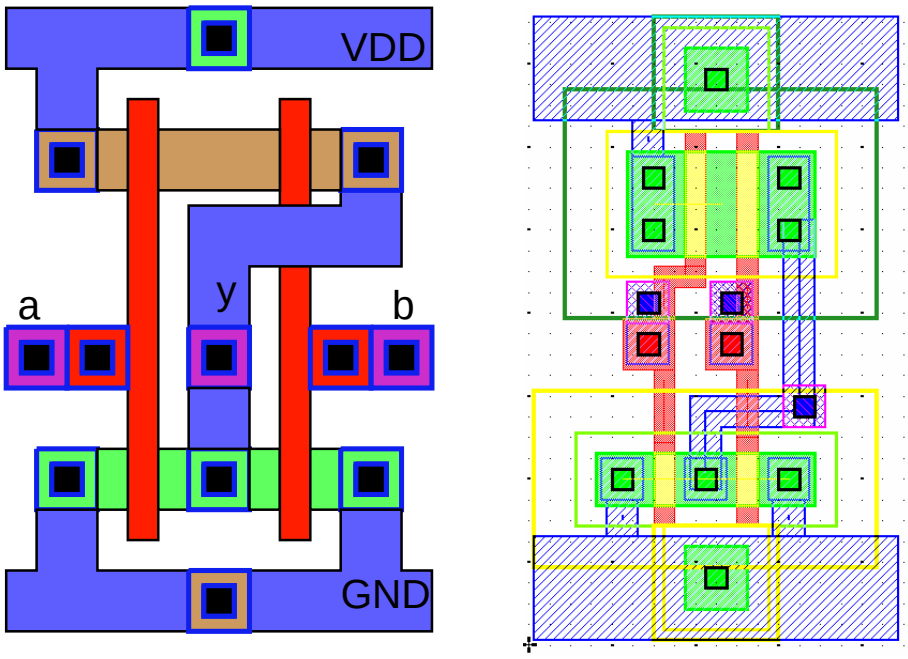


Figure 5.11: A simplified and a complete layout of a 2-input NOR gate in the AMI0.5 technology.

In the stick diagram of Figure 5.10 and the layouts of the two-input NOR gate shown in Figure 5.11 identify:

- Metal 1 connections to V_{DD}, GND and output, y.
- Two pMOS transistors connected in serial.

- Two nMOS transistors connected in parallel.
- Well contacts, (nWellCut and pWellCut).
- Input terminals, a, b which start at the Metal2-to-Metal1 vias and connected through Metal1-to-poly contacts (PolyCut) to polysilicon paths extending to the transistor gates.

Using CAD tools we can extract from the circuit layout parasitic resistances and capacitances associated with each electrical connection. Values of such parameters can be inserted (back annotated) into the circuit schematic as shown in Figure 5.12. Resistances are given in

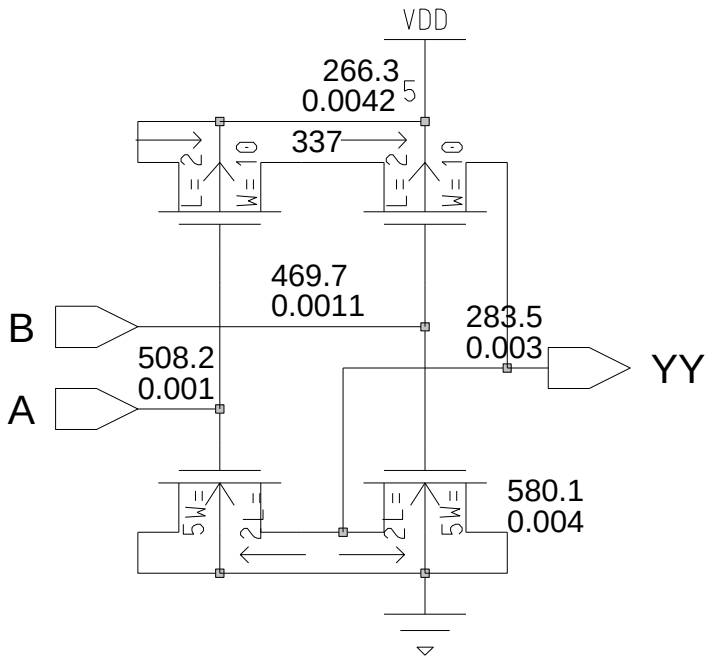


Figure 5.12: A back-annotated schematic of a 2-input NOR gate.

Omhs and capacitances in fF (10^{-15}) For example, the equivalent parasitic resistance and capacitance associated with the output node yy created by a metal 1 connection between p-diffusion, output port and n-diffusion, and related contacts are equal to 283.5Ω and 0.003fF.

Now we can perform the post-layout simulation which takes into account the extracted parasitic resistances and capacitance in addition to transistor parameters like gate capacitances. The waveforms resulting from such a simulation are shown in Figure 5.13.

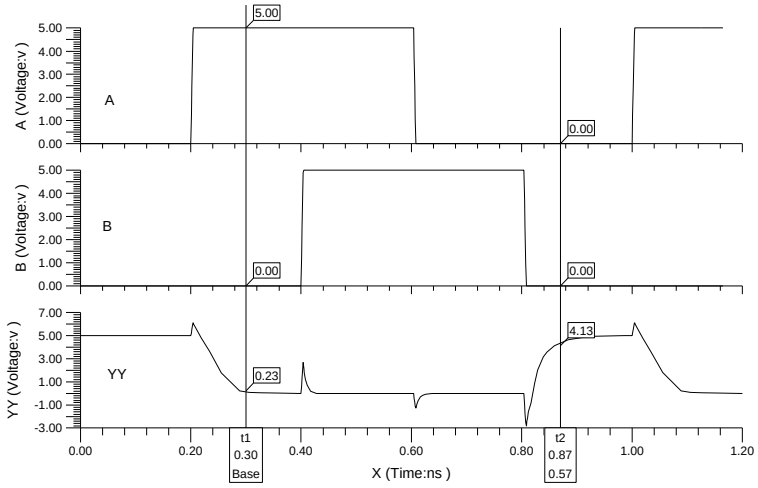


Figure 5.13: The post-layout simulation results of a 2-input NOR gate in the AMI0.5 technology.

From the plots we can estimate that the propagation delay is in the order of 50ps.

5.5 Layout Matrices

The structure of the circuit layout can be conveniently described in the textual form by means of the **layout matrices**.
The layout matrices describe primarily the structure of transistors and contacts along the diffusion paths as in the AMPLE device generator presented in sec. 5.3.
Specifically, the first and the last rows of a layout matrix describe the interconnection of the pMOS and nMOS transistors, respectively, in the basic form:

```
{<source_node>(<transistor_name>)<drain_node>[!]}
```

The middle row(s) contain gate signals of the respective transistors. If the interconnected

```
<drain_node> <source_node>
```

represent the same electric node the node name is used only once. If the electric nodes are different, the names are separated with the exclamation mark.

A 2-input NOR gate of the structure as shown in Figure 5.10 can be describe by the following layout matrix:

$$L_{\mathrm{NOR2}} = \begin{bmatrix} V_{DD} & (pa) & h & (pb) & y \\ & a & & b & \\ \mathrm{GND} & (na) & y & (nb) & \mathrm{GND} \end{bmatrix} \tag{5.1}$$

To emphasize a serial connection of two pMOS transistors the node "h" can be omitted.

5.6 A three-input NAND gate

As previously we re-write the logic equation for the 3-input NAND gate, namely

$$y = \overline{a \cdot b \cdot c}$$

into two equations, one for generation of ones, the other for generation of zeros, namely

ones when	$y_p = \bar{a} + \bar{b} + \bar{c}$	is true	(parallel connection of pMOSs)
zeros when	$y_n = a \cdot b \cdot c$	is true	(serial connection of nMOSs)

these equations result in a schematic as in Figure 5.14.

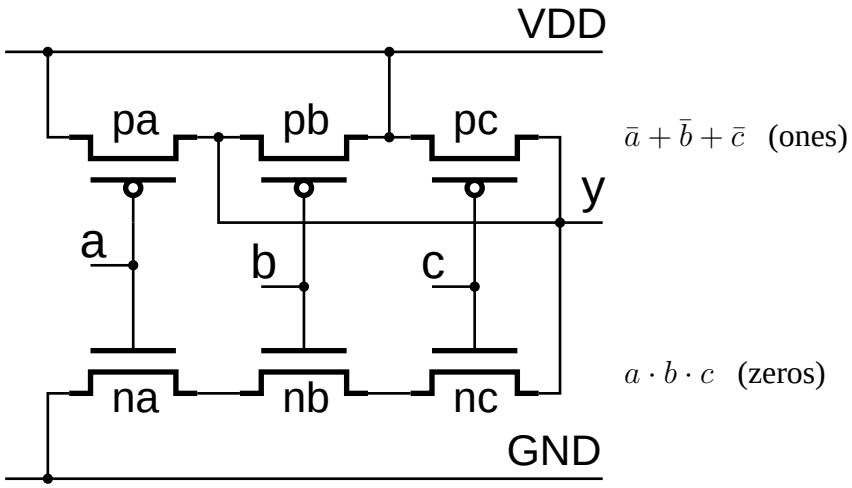


Figure 5.14: Schematic of a 3-input NAND gate

From Figure 5.14 note three pMOS transistors connected in parallel between the V_{DD} and the y nodes, and three nMOS transistors connected in series between the GND and the y nodes.

The schematic can be easily converted into an equivalent stick diagram shown in Figure 5.15.

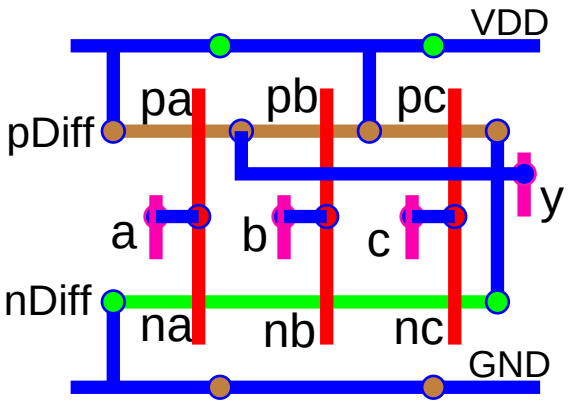


Figure 5.15: Stick diagram of a 3-input NAND gate

Note that the stick diagram follows the “principle of the good layout” described in sec. 5.2.

The equivalent layout matrix is of the following form:

$$L_{\mathrm{NAND3}} = \begin{bmatrix} V_{DD} & (pa) & y & (pb) & V_{DD} & (pc) & y \\ & a & & b & & c & \\ \mathrm{GND} & (na) & & (nb) & & (nc) & y \end{bmatrix} \tag{5.2}$$

Note that names of the electric nodes between the serially connected nMOS transistors are omitted.

Finally, a possible circuit layout generated by Mentor Graphics tools is presented in Figure 5.16.

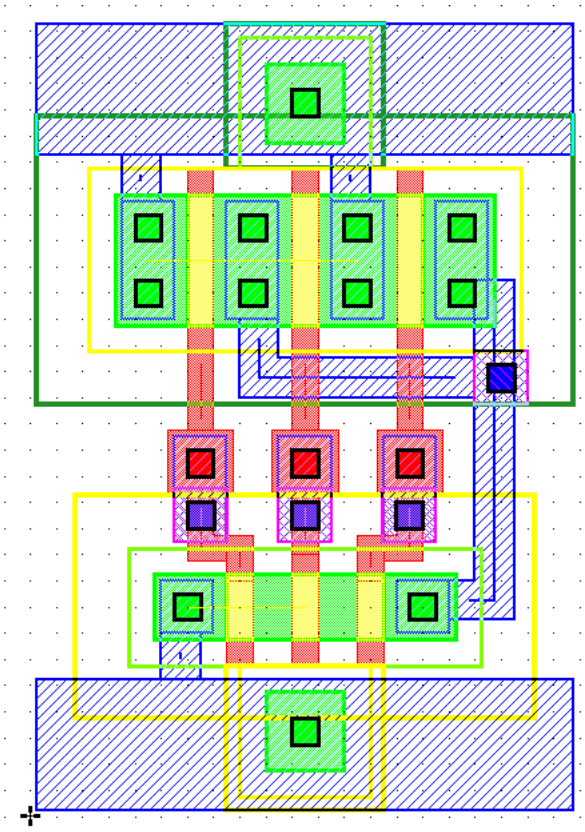


Figure 5.16: Layout of a 3-input NAND gate in the AMI0.5 technology.

The back-annotated schematic is given in Figure 5.17. This time only connection capacitances are shown in the schematics. The values are in aF (attoFarads = 10^{-18}F).
Post-layout simulation waveforms are given in Figure 5.18.
Note that a longer propagation delay occurs for the high-to-low transition. This is due to the fact that in this case the output node is being connected to GND node through three serially connected nMOS transistors.

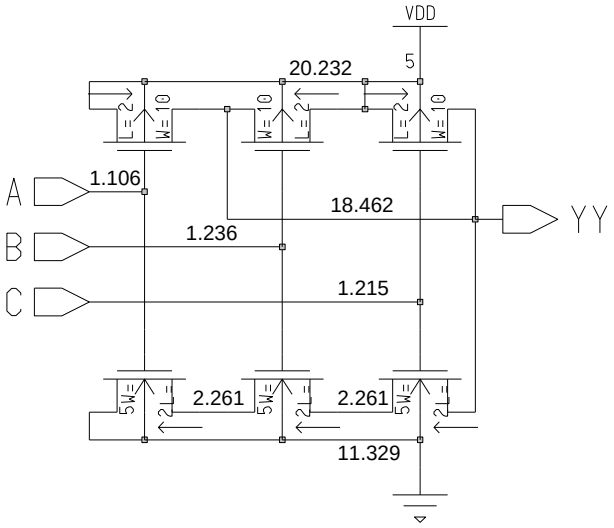


Figure 5.17: Back-annotated schematic of a 3-input NAND gate

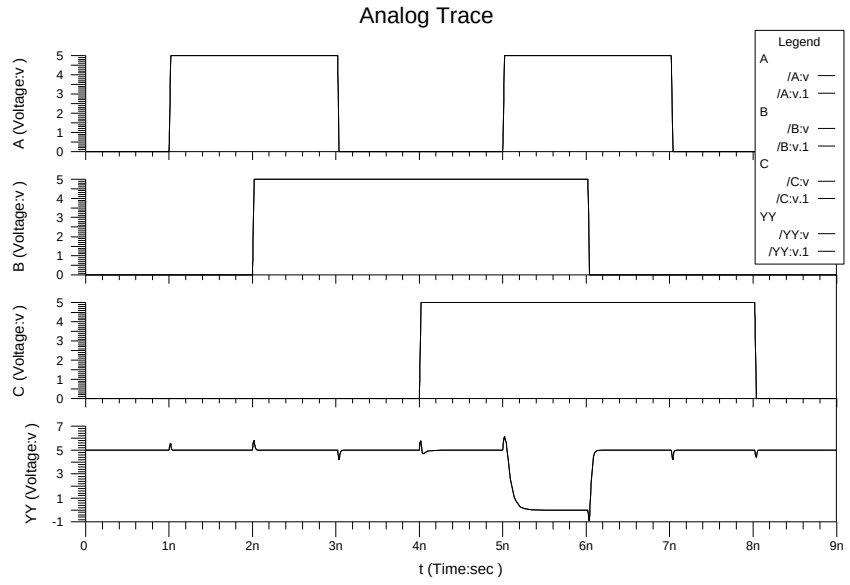


Figure 5.18: Post-layout simulation waveforms of a 3-input NAND gate

5.7 A composite CMOS gate

A **composite** CMOS gate is a generalization of simple NAND and NOR gates and consists of two complementary switching circuits:

- an nMOS pull-down circuit connected between the GND node and the gate output, y, and
- a pMOS pull-up circuit connected between the V_{DD} node and the gate output, y, as Figure 5.19.

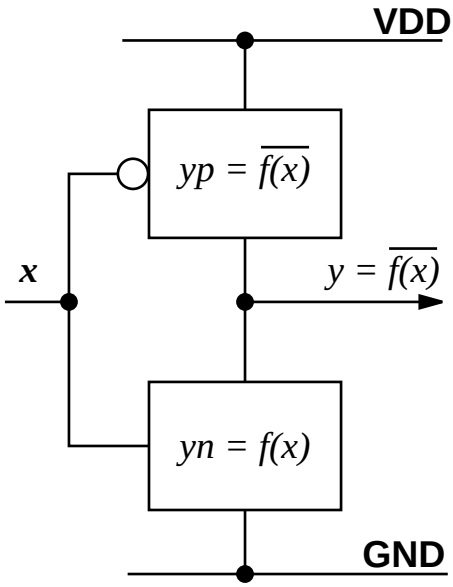


Figure 5.19: A generalised schematic of a composite CMOS gate

The **logic function** which is to be implemented by the composite gate should be represented in the **complemented form** as:

$$y = \overline{f(\mathbf{x})} \tag{5.3}$$

where $\mathbf{x}$ is an n-bit input vector, and y is a 1-bit output signal.

The **nMOS pull-down** circuit specifies **zeros** of the function $\bar{f}$, that is, the cases when the output y should be connected to the GND node. Therefore, the nMOS pull-down circuit should implement the **complement** of the original function, namely:

nMOS:

$$y_n = \overline{y} = f(\mathbf{x}) \tag{5.4}$$

The **pMOS pull-up** circuit specifies **ones** of the function $\bar{f}$, that is, the cases when the output y should be connected to the V_{DD} node. Therefore the pMOS pull-up circuit should implement the **original** function. However, because the input signals are complemented at the pMOS gates, we have to use the Morgan's rule and complement the original function, namely:

pMOS:

$$y_p = y = \overline{f(\mathbf{x})} = g(\overline{\mathbf{x}}) \tag{5.5}$$

where g is the complement of f.

Two **switching circuits**, namely, the pMOS pull-up specified in eqn (5.5) and nMOS pull-down specified in eqn (5.4) are designed following the simple rules of the switch logic, namely:

- There is a transistor switch associated with each variable and controlled by this variable
- The parallel connection of transistor switches represents the OR function.
- The serial connection of transistor switches represents the AND function.

5.7.1 Example: An Inverted-sum-of-products composite gate

Consider a **composite CMOS gate** which implements the following logic function given in the inverted Sum-of-Products (SoP) form (AND-OR-Inverter, aoi):

$$y = \overline{a \cdot b + c} \tag{5.6}$$

The above logic function is in an inverted form as required by eqn (5.4).

The first step is to design the **nMOS switching circuit** implementing the complement of the function (5.6), that is, zeros of the logic function, namely:

$$y_n = \overline{y} = a \cdot b + c$$

The resulting switching circuit is a parallel connection of the nMOS transistor controlled by the c signal with the serial connection of the a and b nMOS transistors as in Figure 5.20.

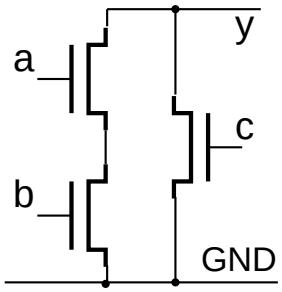


Figure 5.20: The nMOS switching circuit implementing zeros of the logic function specified in eqn (5.6).

The **pMOS switching circuit** should implement the original function as in eqn (5.6), that is ones of the logic function, but after De Morgan's transformation, namely:

$$y_p = y = \overline{a \cdot b + c} = (\overline{a} + \overline{b}) \cdot \overline{c}$$

The resulting switching circuit is a serial connection of the c pMOS transistor with the parallel connection of the a and b pMOS transistors as in Figure 5.21.

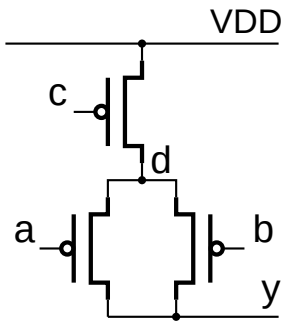


Figure 5.21: The pMOS switching circuit implementing ones of the logic function specified in eqn (5.6).

Connecting the pMOS and nMOS parts we obtain the schematic of the complete composite CMOS gate as in Figure 5.22.

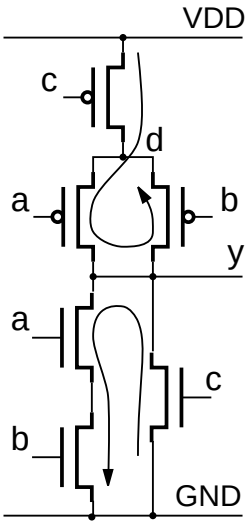


Figure 5.22: A schematic of a composite CMOS gate resulting from assembling pMOS and nMOS switching circuits which implements the logic function of eqn (5.6).

The next step is to convert the schematic into a “linear” form, with pMOS and nMOS transistors connected in a serial-looking way.

We do this by “walking” along the branches of the circuit as indicated in Figure 5.22. We try to walk synchronously along the corresponding pMOS and nMOS transistor, that is, transistors driven by the same signal.

In Figure 5.22 we start with transistors driven by the signal c, an then walking along transistors driven by the signals a and b.

Resulting pairs of the transistors are placed horizontally as in Figure 5.23.

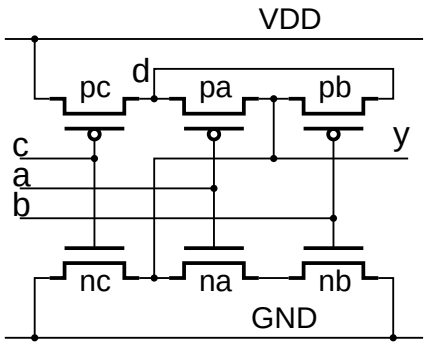


Figure 5.23: A “linear” version of the schematic from Figure 5.22.

The equivalent layout matrix can be written in the following form:

$$L_{\mathrm{NAND3}} = \begin{bmatrix} V_{DD} & (pc) & d & (pa) & y & (pb) & d \\ & c & & a & & b & \\ \mathrm{GND} & (nc) & y & (na) & & (nb) & \mathrm{GND} \end{bmatrix} \tag{5.7}$$

Note that each name of the electric node in the diffusion rows entails a contact between diffusion and metal 1 layer.

With reference to Figure 5.23 a natural question arises if such a transformation to a “linear” schematic is always possible. We will study this problem in the next section.

From the “linear” schematic given in Figure 5.23 the corresponding stick diagram can be easily obtained, as in Figure 5.24.

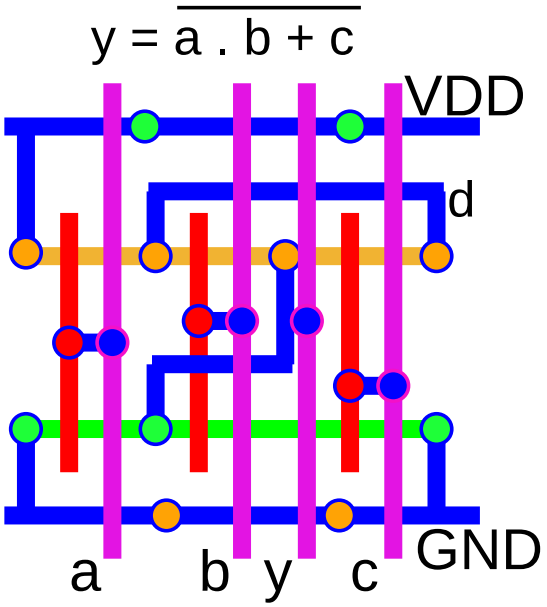


Figure 5.24: A stick diagram of a composite gate derived from the “linear” schematic

Note similarity in the topology of the linear schematic and the stick diagram.
However, in the stick diagram, apart from the location of transistors we specify the type of interconnecting material and associated contacts between layers.

As it was explained previously the input output signals should be available on the metal 2 layer, therefore suitable vias between metal 1 and metal 2 are needed.

In the stick diagram of Figure 5.24 identify:

- paths: n-diffusion, p-diffusion, polysilicon, metal 1 and metal 2,
- 4 p-diffusion contacts (pdiffcut), 3 n-diffusion contacts (ndiffcut), 3 polysilicon contacts (polycut), 4 metal2 contacts (via, m2cut) and 2 well contacts, nWellcut, pWellcut.

The complexity of the CMOS circuitry can be evaluated by the number of transistors and contacts need to build the circuit.

For the above example, from the stick diagram of Figure 5.24 we can find out that the circuit consists of 6 transistors and 14 contacts, plus the well/substrate contacts.

The number of contacts is an approximate measure of the complexity of interconnections in the circuit.

5.8 Switch logic

5.8.1 Switch logic versus gate logic

The **CMOS gate logic** is composed of cells of the general structure as presented in Figure 5.19. The principal features of the gate logic are as follows:

- Input signals are connected only to gates of transistors.
- Input signals control the process of connecting the output signal either to VDD (logic ”1”), or to GND (logic ”0”)
- No power is drawn from the inputs.
- Output signals are re-generated thanks to power amplification.

The **CMOS switch logic** is composed of MOS transistors connected between input and output signals as in Figure 5.25.

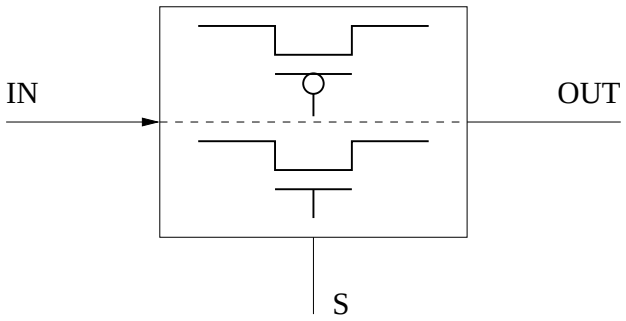


Figure 5.25: A general structure of the switch logic circuitry

The defining features of the switch logic are as follows:

- Input signals “IN” are passed through transistors to the output “OUT”. The term “**pass transistor**” is used in this context.
- Signal level degradation may occur.
- No power amplification to re-generate signals is present,
- Current is drawn from the inputs.
- Transistors are controlled by signals “S” connected to the gates

The main attraction of the switch logic is a smaller size of circuits which use it.

5.8.2 Signal degradation in switch logic

When a MOS transistor is switched on (by a suitable gate voltage) and an input signal is transfer to the output, a **degradation of the signal levels** occurs as demonstrated in Figure 5.26.

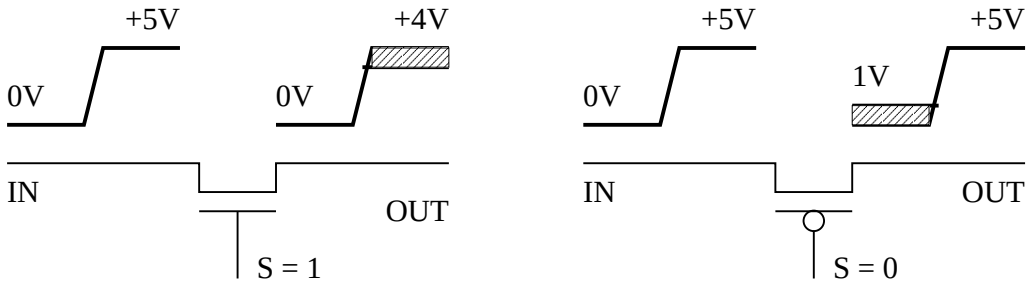


Figure 5.26: Signal level degradation in pass transistors.

- nMOS passes the **low signal level** (say 0V) **without degradation**, but reduces the high level (say, +5V) by the threshold voltage (say, +1V)
- pMOS passes the **high signal level** (say +5V) **without degradation**, but increase the low level (say, 0V) by the treshold voltage (say, +1V)

When the gate signal S is low for nMOS, or high for p-MOS, the transistor is switched off, that is, **output is disconnected** from input.

5.8.3 A weak multiplexer.

A 2-to-1 multiplexer can be specified by the following logic and functional equations:

$$y = \bar{s} \cdot x_0 + s \cdot x_1 \,, \quad \text{or} \quad y = \begin{cases} x_0 & \text{if} \quad s = 0 \\ x_1 & \text{if} \quad s = 1 \end{cases} \tag{5.8}$$

The above equation entails the NAND-based logic diagram as presented in Figure 5.27.

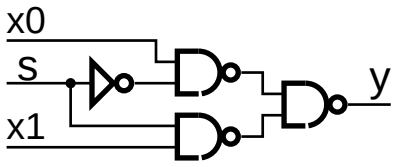


Figure 5.27: The logic diagram of the NAND-based 2-to-1 multiplexer.

If we implement this equation using NAND gates we need total of **14 transistors**! (four to implement a 2-input NAND gate, and two for an inverter).

This number can be reduced to 12 if we employ the concept of the composite gate.

However, the functional equation (5.8) suggests the amazingly simple and elegant **2-transistor solution** known as a "weak" multiplexer presented in Figure 5.28.

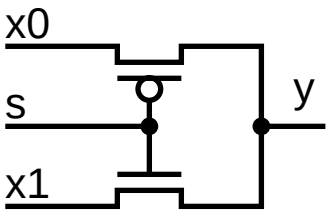


Figure 5.28: The 2-to-1 weak multiplexer

Clearly, when the signal s is **low**, only the **pMOS** pass transistor is "on" and $y = x_0$, and, conversely, when the signal s is **high**, only the **nMOS** pass transistor is "on" and $y = x_1$. We discussed the weak multiplexer already in sec. 3.4 and we know that the problem with the above circuit is that when an input signal, x_0 or x_1, is being passed to the output, the circuit degrades:

- the low level of the x_0 signal,
- the high level of the x_1 signal.

5.8.4 A CMOS Transmission Gate

A transmission gate is a complementary switch formed from a parallel connection of an n-MOS, p-MOS pair. Both transistors are switched on or off synchronously, as in Figure 5.29.

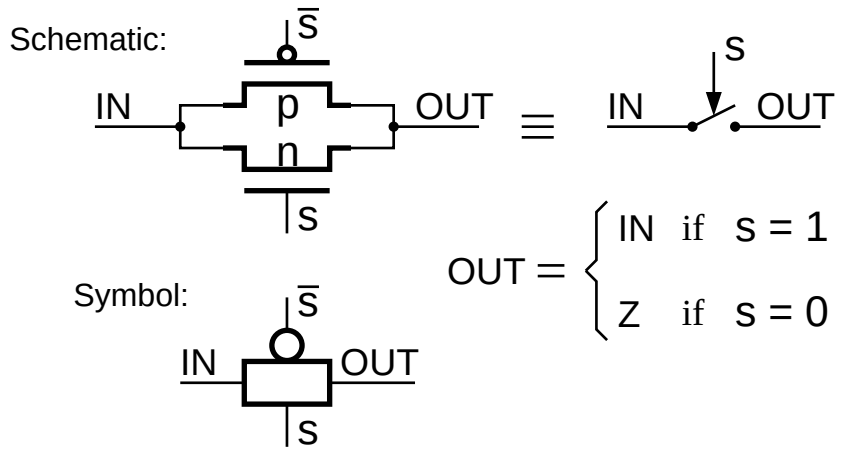


Figure 5.29: A schematic diagram and a symbol of the CMOS transmission gate.

Note that:

- The gates of the transistors are driven by a pair of complementary signals, s and $\bar{s}$.
- Thanks to the parallel connection of two complementary transistors, both signal levels are now transferred through the transmission gate **without degradation**.

The transmission gate is a perfect switch which can be used in place of a pass transistor to prevent the signal level degradation.

Typically, a transmission gate is used together with an inverter used to generate a complementary control signal required to control the transmission gate as in Figure 5.30.

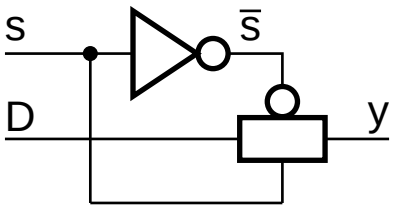


Figure 5.30: The logic diagram of the complementary switch (transmission gate.

Note that the complementary switch is “on” when the control signal s is high and “off” when $s = 0$.

A stick diagram of a pair: an inverter and transmission gate is given in Figure 5.31.

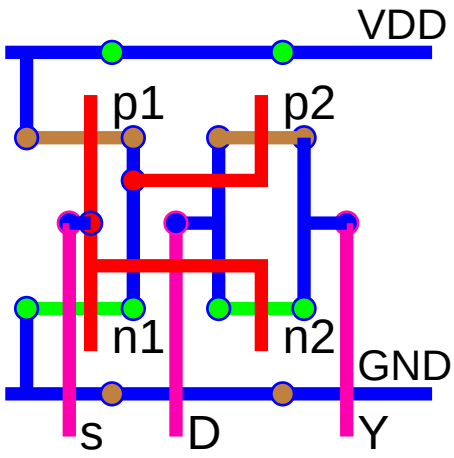


Figure 5.31: The stick diagram of the transmission gate with a control signal inverter.

Note the relative complexity of the circuit which contains **four transistors** and **17 contacts**. One pair of the transistors, (p1, n1), is used to implement an inverter whereas the pair (p2, n2) implements the transmission gate.

The layout matrix equivalent to the stick diagram of Figure 5.31 is of the following form:

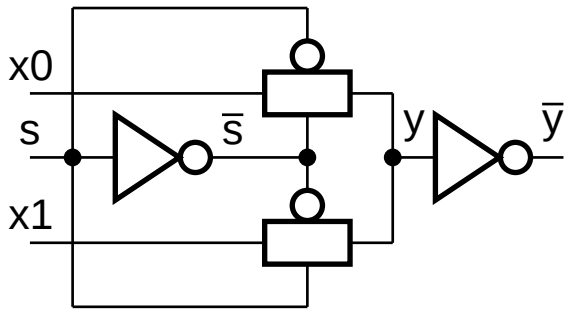
$$L_{\mathrm{TG}} = \begin{bmatrix} V_{DD} & (p1) & \bar{s}\,!\,D & (p2) & y \\ & s & & \bar{s} & \\ & s & & s & \\ \mathrm{GND} & (n1) & \bar{s}\,!\,D & (n2) & y \end{bmatrix} \tag{5.9}$$

Note that there are two middle rows describing the gate signals, one for pMOS transistors and one for nMOS, respectively. The exclamation mark is equivalent to a **gap** in diffusion. The gap exists because different electrical nodes cannot be connected together.

The transmission gate together with the weak multiplexer are fundamental building blocks of digital circuits based on the switch logic.

5.8.5 A 2-to-1 multiplexer using transmission gates

A schematic of a 2-to-1 multiplexer built using transmission gates is shown below:



The circuit includes two inverters: one to generate the complement of the control signal, s, and the other to give power amplification to the output signal in order to increase its fan-out.

It is a good exercise to **draw a stick diagram** of the above circuit.

5.9 Exclusive-OR gate in gate and switch logic

The Exclusive-OR (XOR) function:

$$f = a \oplus b = \bar{a} \cdot b + a \cdot \bar{b} = \overline{a \cdot b + \bar{a} \cdot \bar{b}} \tag{5.10}$$

is a basic building block of a 1-bit adder therefore it is fundamental to all arithmetic circuits. In addition XOR is used in data coding and error detection, the parity circuit being the simplest example.

We will review a number of possible implementations of the XOR gate what will give us an opportunity to employ different design methods considered in previous sections.

First we briefly consider a popular **NAND-based implementation** which is given in Figure 5.32.

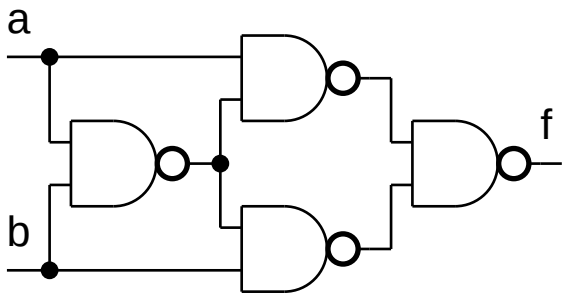


Figure 5.32: The logic diagram of a NAND-based XOR gate.

The NAND-based implementation requires $4 \times 4 = 16$ transistors, which seems to be a rather excessive number for a basic logic gate.

Next we consider an implementation based on a composite gate. One of the possible solution is based on the following modification of eqn (5.10):

$$f = \overline{a \cdot b + c}\,, \quad c = \overline{a + b} \tag{5.11}$$

Eqn (5.11) entails a NOR gate generating the signal c, and a complex gate as discussed in sec. 5.7.1 generating the signal f. We can adopt the schematic of such a composite gate given in Figure 5.23 and proceed it a 2-input NOR gate.

The resulting schematic and the stick diagram is presented in Figure 5.33.

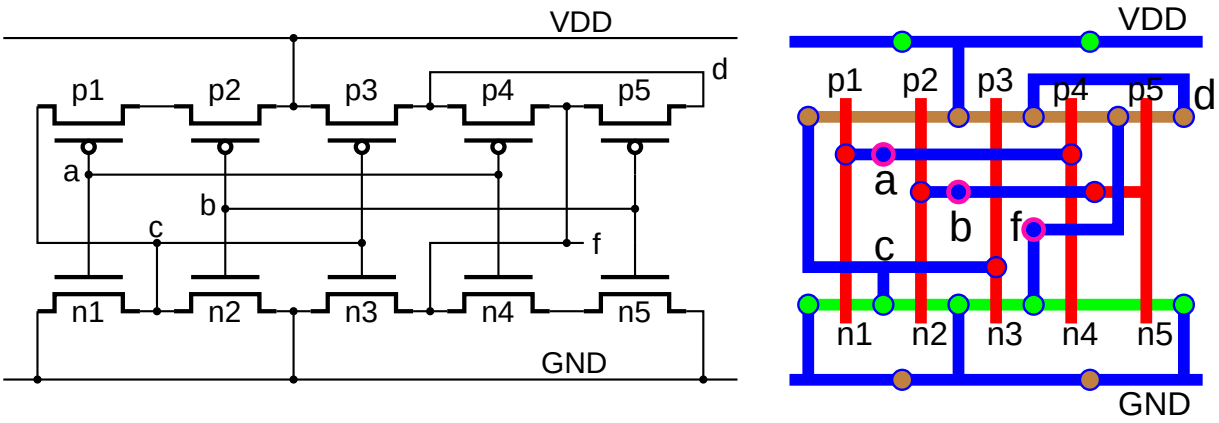


Figure 5.33: An XOR gate implementation based on a composite gate and a 2-input NOR gate.

This implementation has 5 pairs of transistors, which is an improvement on the 16 transistor NAND-based design presented in Figure 5.32. The equivalent layout matrix has the following form:

$$L_{\mathrm{XORa}} = \begin{bmatrix} c & (p1) & & (p2) & V_{DD} & (p3) & d & (p4) & f & (p5) & d \\ & a & & b & & c & & a & & b & \\ \mathrm{GND} & (n1) & c & (n2) & \mathrm{GND} & (n3) & f & (n4) & & (n5) & \mathrm{GND} \end{bmatrix} \tag{5.12}$$

Note a relative simplicity of the layout matrix and related circuit, in particular, the absence of gaps in diffusion paths.

The layout of the XOR cell based on the composite gate concept equivalent to the stick diagram of Figure 5.33 is shown in Figure 5.34. Identify the input/output ports and the way these ports are connected to

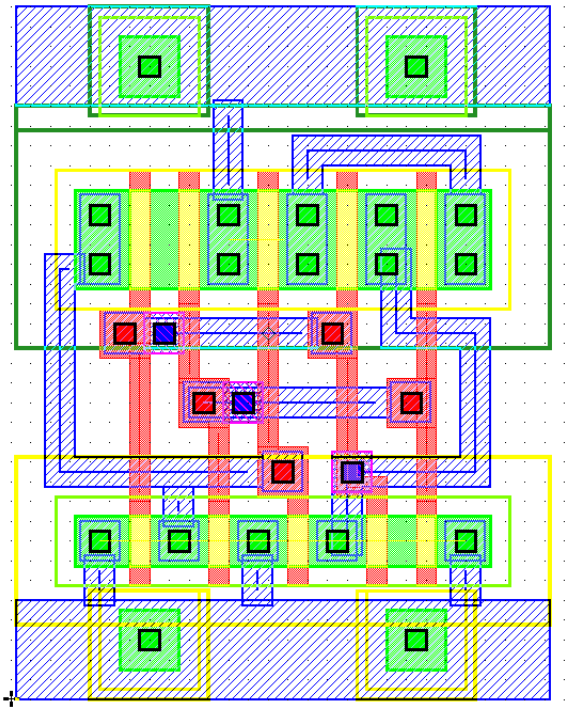


Figure 5.34: Layout

polysilicon and metal 1 layers.

The schematic with the back annotated parasitic capacitances (in fF) extracted from the circuit layout is shown in Figure 5.35.

Finally, the post-layout simulation of the XOR cell is shown in Figure 5.36. The four vertical cursors shows the approximated end of transition of the output signal either from L→H or from H→L. The transition time varies from 100 ps for the 3rd transition to 280 ps for the 4th transition.

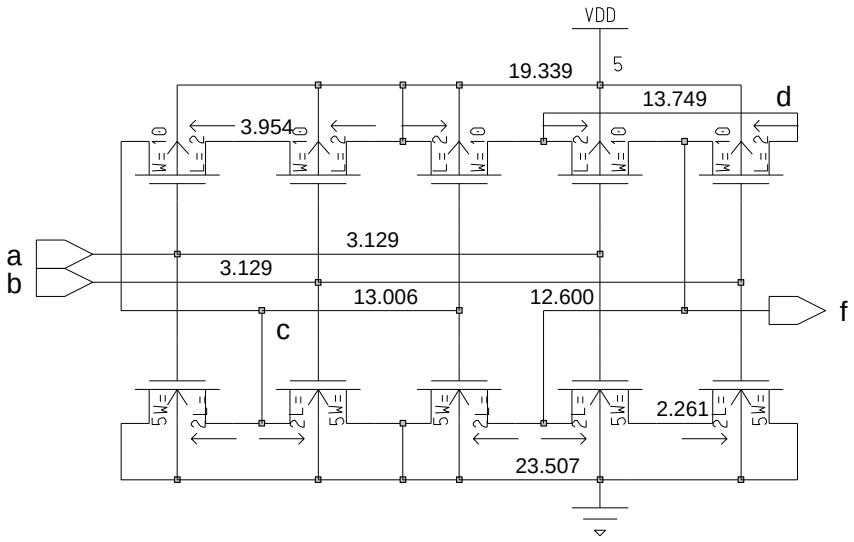


Figure 5.35: Schematic with extracted lumped capacitances.

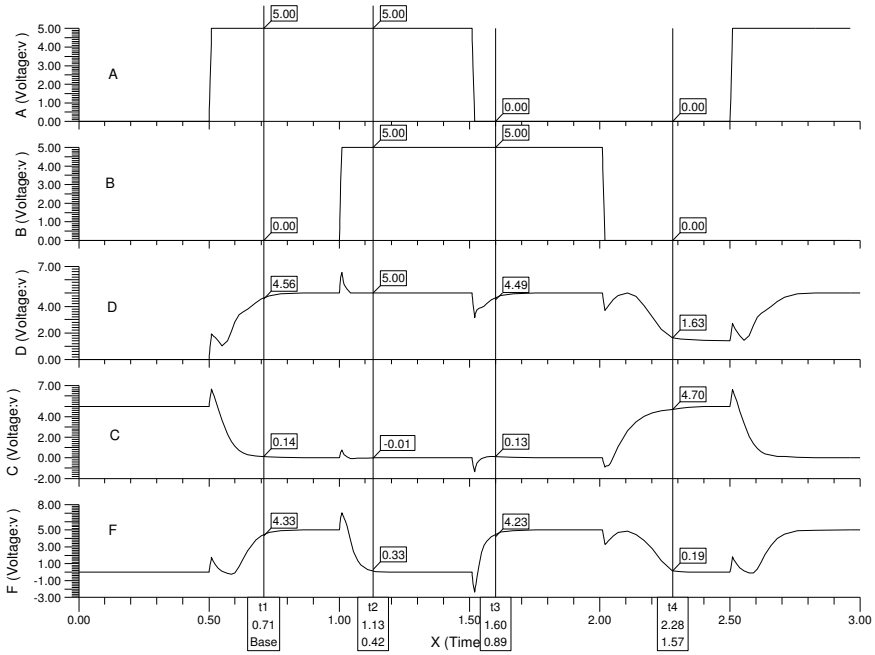


Figure 5.36: Simulation waveforms.

Now we will investigate XOR structures based on the concept of switch logic. We start with a generic implementation based on a **2-to-1 multiplexer**. The relevant truth and function tables and a circuit block-diagram is presented in Figure 5.37.

a	b	f
0	0	0
0	1	1
1	0	1
1	1	0

a	f
0	b
1	$\bar{b}$

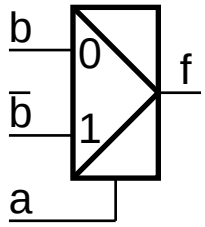


Figure 5.37: The truth and function tables and a block-diagram of an XOR implementation based on a 2-to-1 multiplexer.

An equivalent lower-level schematic with a 2-to-1 multiplexer built from transmission gates is given in Figure 5.38.

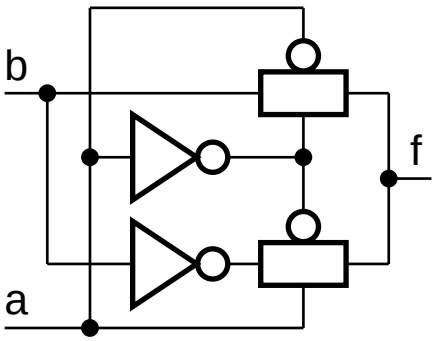


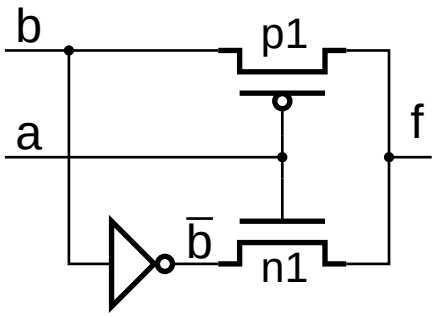
Figure 5.38: An XOR gate based on the transmission gates

This solution requires 4 pairs of transistors (2 per inverter and transmission gate), which is less than the composite gate implementation.

In general, however, an inverter providing power amplification will be required bringing the number of transistor to 2×5.

As an exercise draw a stick diagram of the above XOR circuit.

As it might be expected the smallest number of transistors can be achieved using a **weak multiplexer**. Such a **"weak" XOR** can be build using a multiplexing pair of complementary MOS transistors and an inverter as in Figure 5.39.



a	b	"on"	f	level
0	0	p1	b	weak zero
0	1	p1	b	strong one
1	0	n1	$\bar{b}$	weak one
1	1	n1	$\bar{b}$	strong zero

Figure 5.39: A "weak" XOR.

In the table in Figure 5.39 we indicate when a specific transistor is switched on and what is the level of the output signal, f.

In order to restore signal levels and provide power amplification we can add an output inverter. Such a **buffered weak XOR** has 3 pairs of transistors and its schematic is presented in Figure 5.40.

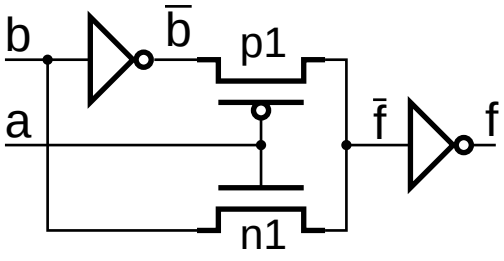


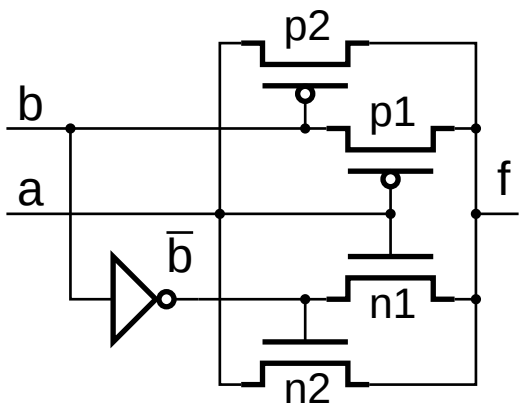
Figure 5.40: A buffered weak XOR

Note that the multiplexer implements now the XNOR function, $\overline{a \oplus b}$, which after inversion will form the required XOR function.

An interesting modification of the “weak” XOR is achieved by adding a transmission gate to avoid signal level degradation.

From the table in Figure 5.39 note that the weak level of signals corresponds to the value of the signal a. These levels can be passed on to the output through a transmission gate controlled by the signal $\bar{b}$.

This results in the following **switch logic XOR gate** implementation where all signal levels are “strong”, as in Figure 5.41.



a	b	“on”	f
0	0	p1, p2, n2	b
0	1	p1	b
1	0	n1, p2, n2	$\bar{\text{b}}$
1	1	n1	$\bar{\text{b}}$

Figure 5.41: An XOR gate based on a weak multiplexer and a transmission gate

The transmission gate is implemented by the CMOS pair (p2, n2). The pair (p1, n1) forms, as previously, a weak multiplexer.

The above solution delivers proper signal levels, however, in realistic circuits, we would still need a buffering output inverter.

The schematic of Figure 5.41 can now be converted into a stick diagram and ultimately into a circuit layout. One possible solution is presented in Figure 5.42. This implementation requires 3 pairs of

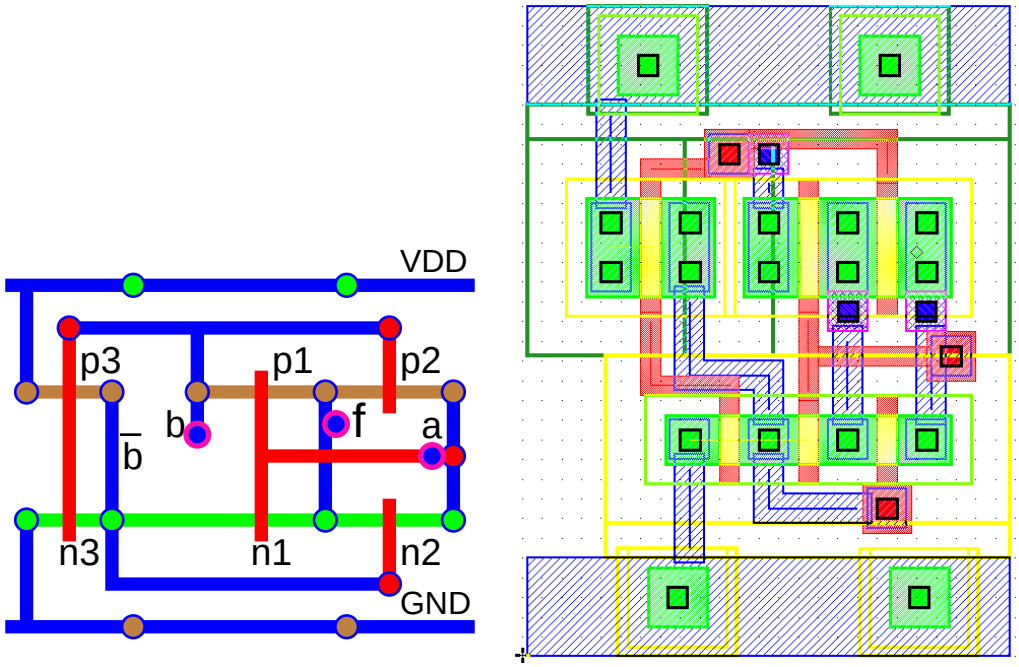


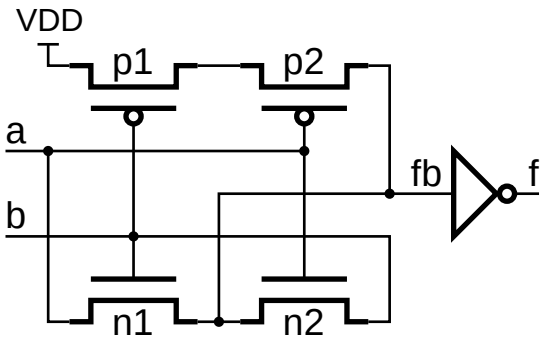
Figure 5.42: A stick diagram and a circuit layout of an XOR cell based on a weak multiplexer and a transmission gate

transistors. An output inverter might be needed for power amplification. The equivalent layout matrix has the following form:

$$L_{\text{XORs}} = \begin{bmatrix} V_{DD} & (p3) & \bar{b} & ! & b & (p1) & f & (p2) & a \\ & b & & & & a & & b & \\ & b & & & & a & & \bar{b} & \\ \text{GND} & (n3) & \bar{b} & & & (n1) & f & (n2) & a \end{bmatrix} \tag{5.13}$$

Comparing this solution with the one based on the composite gate (Figure 5.33) it seems that the switch logic solution gives a smaller XOR cell even if the output inverter is added. For the final assessment we need to compare the speed of both solutions.

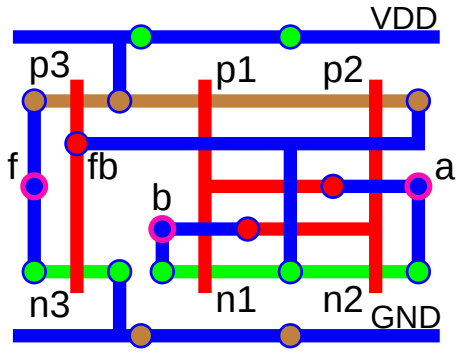
Finally, we consider a switch logic XOR cell which is the candidate for the **best XOR** award from the point of view of its size, speed and signal levels, must be however carefully assessed. The power amplification is provided by the output inverter. The schematic and operation table are given in Figure 5.43.



a	b	"on"	fb	f
0	0	p1, p2	V_{DD}	0
0	1	p2, n1	a	1
1	0	p1, n1	b	1
1	1	n1, n2	a, b	0

Figure 5.43: The "best" switch logic XOR cell.

The high level at the fb output comes either from both inputs when $a = 1$ and $b = 1$ through transistors n1 and n2, or from the $V_(DD)$ terminal through the serial connection of p1 and p2 transistors. The low level comes from one input signal, say $b = 0$ through a n2 transistor, or the other way around.



$$\begin{bmatrix} \text{f} & (p3) & V_{DD} & (p1) & & (p2) & \text{fb} \\ & \text{fb} & & \text{a} & & \text{b} & \\ \text{f} & (n3) & \text{GND} \;!\; \text{b} & (n1) & \text{fb} & (n2) & \text{a} \end{bmatrix}$$

Figure 5.44: A stick diagram and the equivalent layout matrix of the "best" XOR cell.

In Figure 5.45 we present a possible layout of the “best” XOR cell and related schematic annotated with values of the parasitic capacitances extracted from the circuit layout.

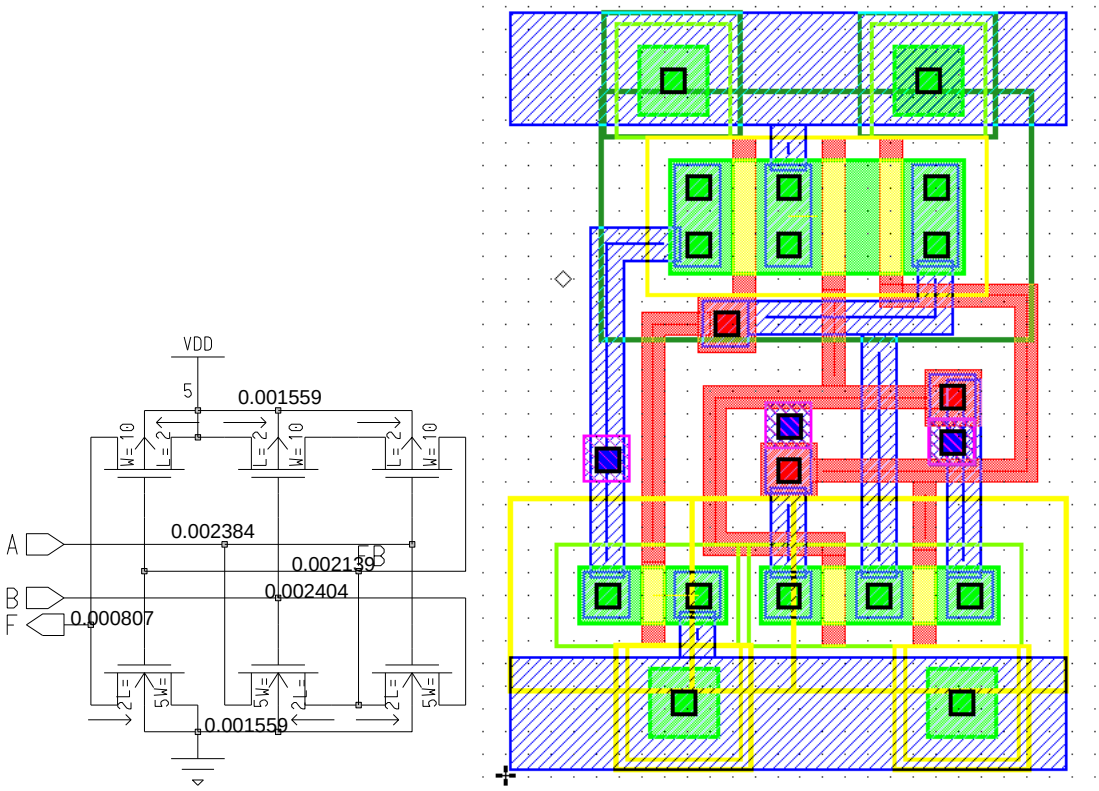


Figure 5.45: The “best” XOR: a schematic back annotated with parasitic capacitances (in fF) and a related circuit layout

Simulation results are presented in Figure 5.46.

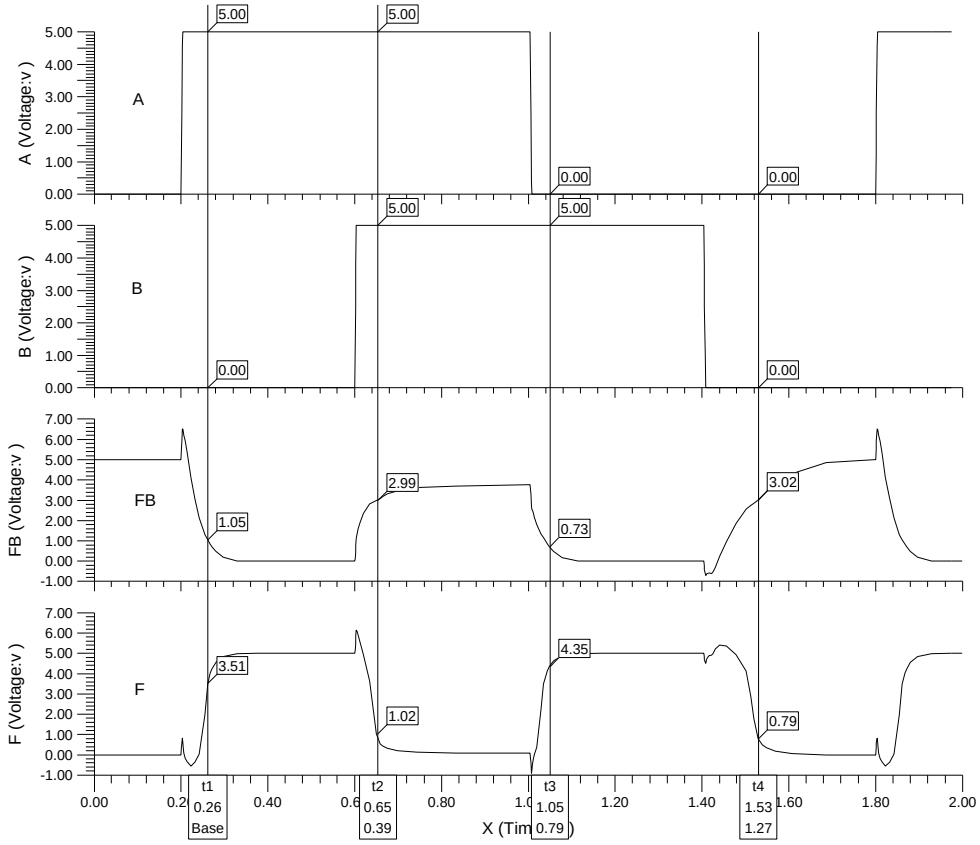


Figure 5.46: The “best” XOR: simulation waveforms

Note that the XOR circuit is being toggled every 400ps, which is equivalent to the switching frequency of 2.5GHz. The longest propagation delay occurs when both serially connected pMOS transistors are switched on and the high level at the FB net comes from V_{DD}. This delay is approximately equal to 100ps.

5.10 A Tri-state inverter

The tri-state inverter has an additional **enable** input, e, which controls the mode of operation:

- for e = 1, the inverting mode, $y = \bar{a}$,
- for e = 0, the high impedance mode, $y = Z$.

The tri-state inverter can be build from a standard inverter followed by a transmission gate as illustrated in Figure 5.47

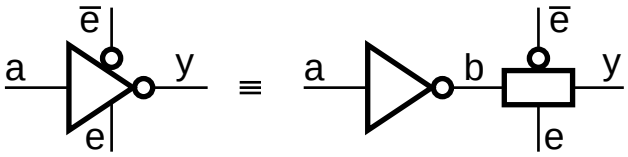


Figure 5.47: A symbol of a tri-state inverter and its equivalent implementation using the transmission gate.

For e = 0, when the transmission gate is open the output, y, is disconnected from the inverter and is in the high impedance state.

The generic tri-state inverter with the transmission gate can be improved by rearranging its schematic as illustrated in Figure 5.48.

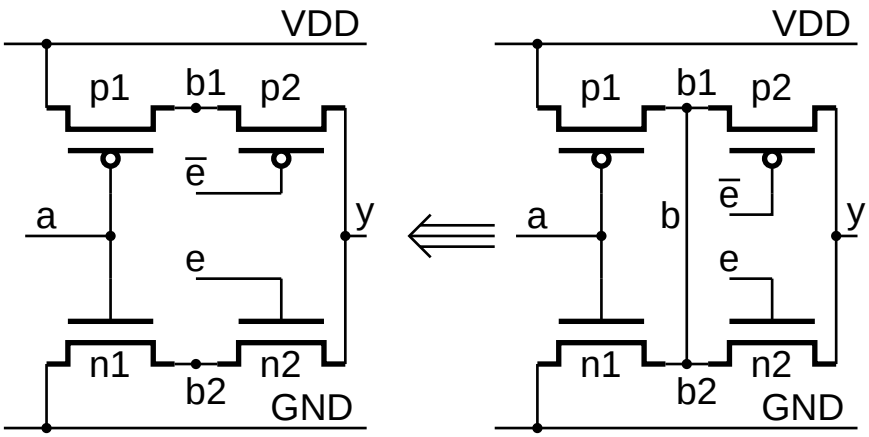


Figure 5.48: Schematic diagrams of an improved tri-state inverter (left) and its predecessor with the transmission gate (right).

The only difference is that the electrical net (connection) b has been removed. However it also removes two diffusion contacts from the circuit layout as illustrated in Figure 5.49.

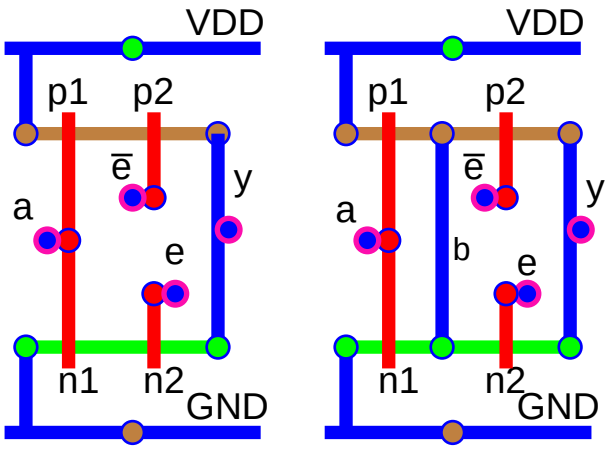


Figure 5.49: Stick diagrams of an improved tri-state inverter (left) and its predecessor with the transmission gate (right).

As a result the layout of an improved tri-state inverter is more compact comparing with its transmission gate based predecessor.