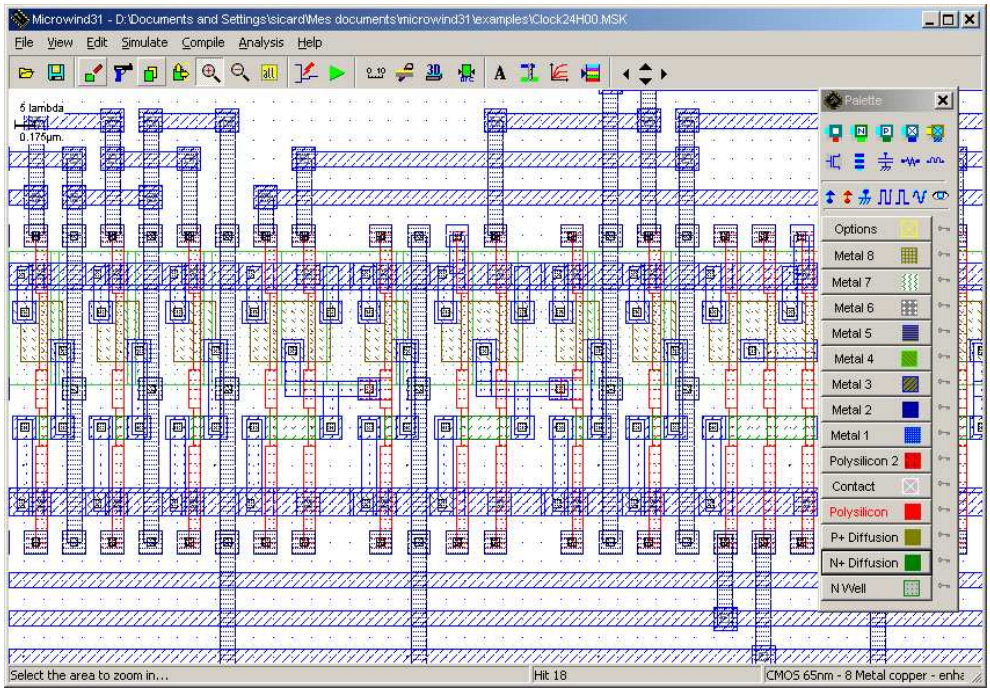
Microwind & Dsch
Version 3.1



User's Manual Lite Version

Etienne Sicard
www.microwind.org

August 2006

About the author

Etienne SICARD was born in Paris, June 1961. He received the B.S degree in 1984 and the PhD in Electrical Engineering from the University of Toulouse, in 1987, in the laboratory LAAS of Toulouse. He was granted a Monbusho scholarship and stayed 18 months at Osaka University, Japan (1988-1989). Previously a professor of electronics in the Department of Physics, University of Balearic Islands, SPAIN (1990), Etienne SICARD is currently a professor at INSA of Toulouse, Department of Electrical and Computer Engineering. He is a visiting professor at the electronic department of Carleton University, Ottawa, in 2004. His research interest include several aspects of CAD Tools for the design of integrated circuits, including signal integrity in deep sub-micron CMOS ICs and electromagnetic compatibility.
Etienne SICARD is the author of books and several software in the field of micro-electronics, sound processing as well as technical papers concerning electromagnetic compatibility of CMOS integrated circuits. He is a member of French SEE and the IEEE EMC society. He was elected in 2006 distinguished IEEE lecturer for EMC of ICs. Email: etienne.sicard@insa-toulouse.fr

About ni2designs <to be verified by vinay>

Ni2designs is founded on a culture based on the principles of open communication, empowerment, trust, integrity, and giving back to the community, and these same values thrive at ni logic today. Ni2designs family brings together a rich experience across all domains of EDA. Ni2designs offer services to companies across the spectrum of nano VLSI, microelectronics, embedded systems and digital design.

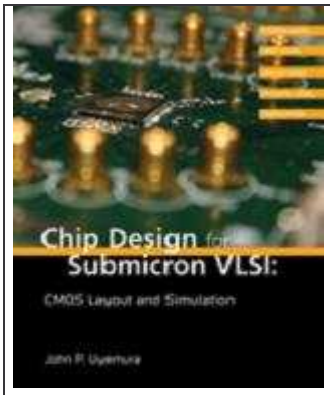
Ni2designs Pvt. Ltd.,
B-5/25, Bandal Complex,
Bhusari Colony, Paud Road, Kothrud, Pune - 411 038
Maharashtra, India
Tele-Fax 91 - 20 - 2528 6948, Phone-WLL 91 - 20 - 3090 4378
Website : www.ni2designs.com, Email info@ni2designs.com

Web information

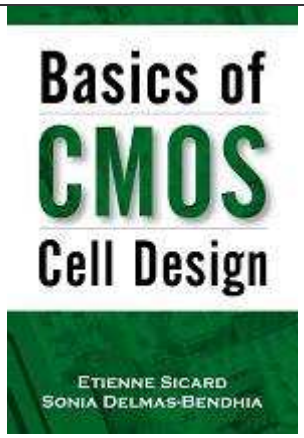
- www.microwind.org for general information about MICROWIND
- www.microwind.net to download the lite version and order the professional version
- www.ni2designs.com for more information about ni2designs company

Copyright

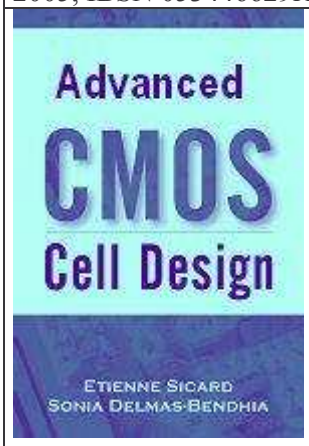
Books Using Microwind



"Chip Design for Submicron VLSI: CMOS Layout and Simulation" by John Uyemura - Georgia Institute of Technology, USA - based on Microwind & Dsch http://www.engineering.thomsonlearning.com 2005, IBSN 053446629X



"Basic CMOS cell Design" by Etienne Sicard and Sonia Bendhia, McGraw-Hill India, 2005, ISBN 0-07-059933-5 (international edition to appear 2007)



"Advanced CMOS cell design" by Etienne Sicard and Sonia Ben Dhia, McGraw-Hill India, end 2006, ISBN pending (international edition to appear 2007)

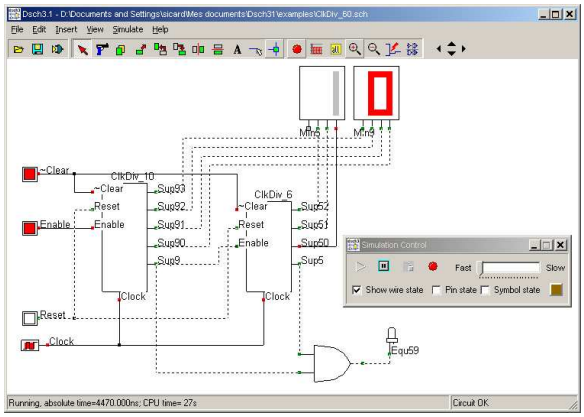
ISBN 2-87649-xxx-x
Published by INSA Toulouse
135 Av de Rangueil
31077 Toulouse France
First print : September 2006

Table of Contents

Introduction

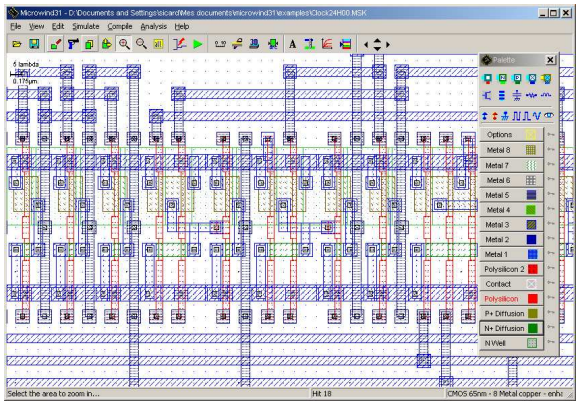
The present document introduces the design and simulation of CMOS integrated circuits, in an attractive way thanks to user-friendly PC tools DSCH and MICROWIND. The *lite* version of these tools only includes a subset of available commands. The *lite* version is freeware, available on the web site www.microwind.net. The complete version of the tools is available through *ni2designs* India (www.ni2designs.com).

About DSCH



The DSCH program is a logic editor and simulator. DSCH is used to validate the architecture of the logic circuit before the microelectronics design is started. DSCH provides a user-friendly environment for hierarchical logic design, and fast simulation with delay analysis, which allows the design and validation of complex logic structures. Some techniques for low power design are described in the manual. DSCH also features the symbols, models and assembly support for 8051 and 18f64. DSCH also includes an interface to SPICE.

About MICROWIND



The MICROWIND program allows the student to design and simulate an integrated circuit at physical description level. The package contains a library of common logic and analog ICs to view and simulate. MICROWIND includes all the commands for a mask editor as well as original tools never gathered before in a single module (2D and 3D process view, Verilog compiler, tutorial on MOS devices). You can gain access to *Circuit Simulation* by pressing one single key. The electric extraction of your circuit is automatically performed and the analog simulator produces voltage and current curves immediately.

The chapters of this manual have been summarized below. Chapter 2 is dedicated to the presentation of the single MOS device, with details on the device modeling, simulation at logic and layout levels.

Chapter 3 presents the CMOS Inverter, the 2D and 3D views, the comparative design in micron and deep-submicron technologies. Chapter 4 concerns the basic logic gates (AND, OR, XOR, complex gates), Chapter 5 the arithmetic functions (Adder, comparator, multiplier, ALU). The latches and memories are detailed in Chapter 6.

As for Chapter 7, analog cells are presented, including voltage references, current mirrors, operational amplifiers and phase lock loops. Chapter 8 concerns analog-to-digital, digital to analog converter principles. and radio-frequency circuit. The input/output interfacing principles are illustrated in Chapter 9.
The detailed explanation of the design rules is in Chapter 10. The program operation and the details of all commands are given in the help files of the programs.

INSTALLATION

Connect to the web page www.microwind.net for the latest information about how to download the lite version of the software. Once installed, two directories are created, one for MICROWIND31, one for DSCH31, as illustrated below.

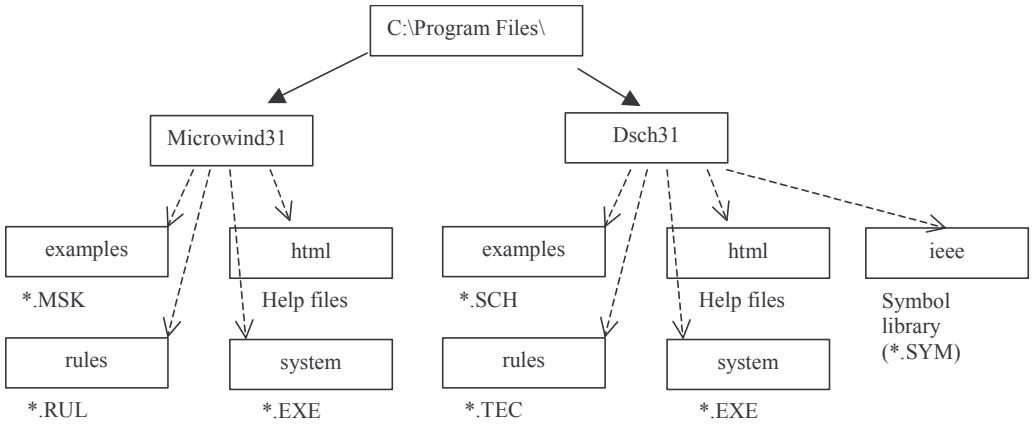


Figure 0-1: The architecture of Microwind and Dsch

Once installed, two directories are created, one for MICROWIND31, one for DSCH31. In each directory, a sub-directory called `html` contains help files. In MICROWIND31, other sub-directories include example files (*.MSK), design rules (*.RUL) and system files (mainly microwind31.exe). In DSCH31, other sub-directories include example files (*.SCH and *.SYM), design rules (*.TEC) and system files (mainly dsch31.exe).

1 Technology Scale Down

The Moore's Law

Recognizing a trend in integrated circuit complexity, Intel co-founder Gordon Moore extrapolated the tendency and predicted an exponential growth in the available memory and calculation speed of microprocessors which, he said in 1965, would double every year [Moore]. With a slight correction (i.e. doubling every 18 months, see figure 1-1), *Moore's Law* has held up to the Itanium® 2 processor which has around 400 million transistors.

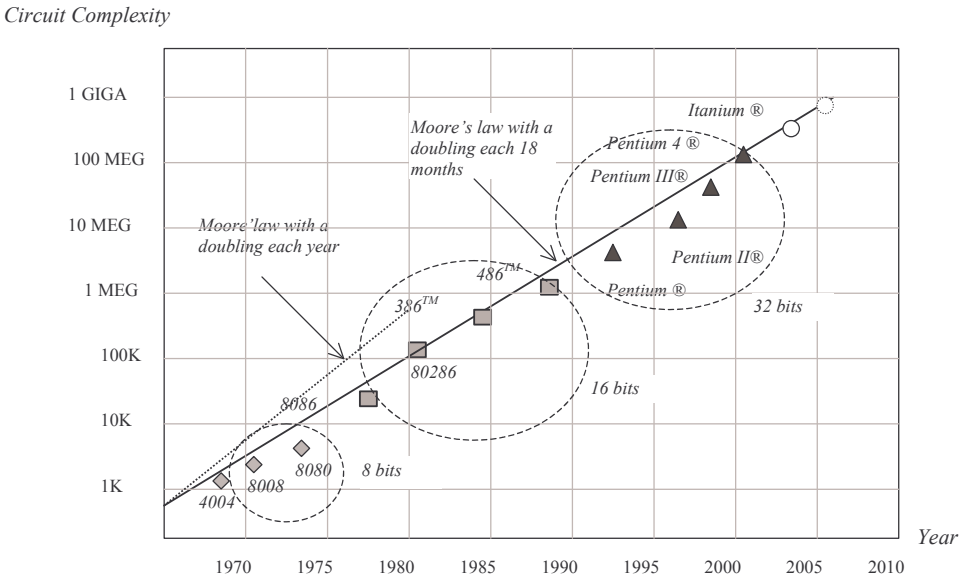


Figure 1-1 : Moore's law compared to Intel processor complexity from 1970 to 2010.

Scaling Benefits

The trend of CMOS technology improvement continues to be driven by the need to integrate more functions in a given silicon area. At each lithography scaling, the linear dimensions are approximately reduced by a factor of 0.7, and the areas are reduced by factor of 2. Smaller cell sizes lead to higher integration density which has risen to nearly 1 million gates per mm^2 in 90 nm technology.

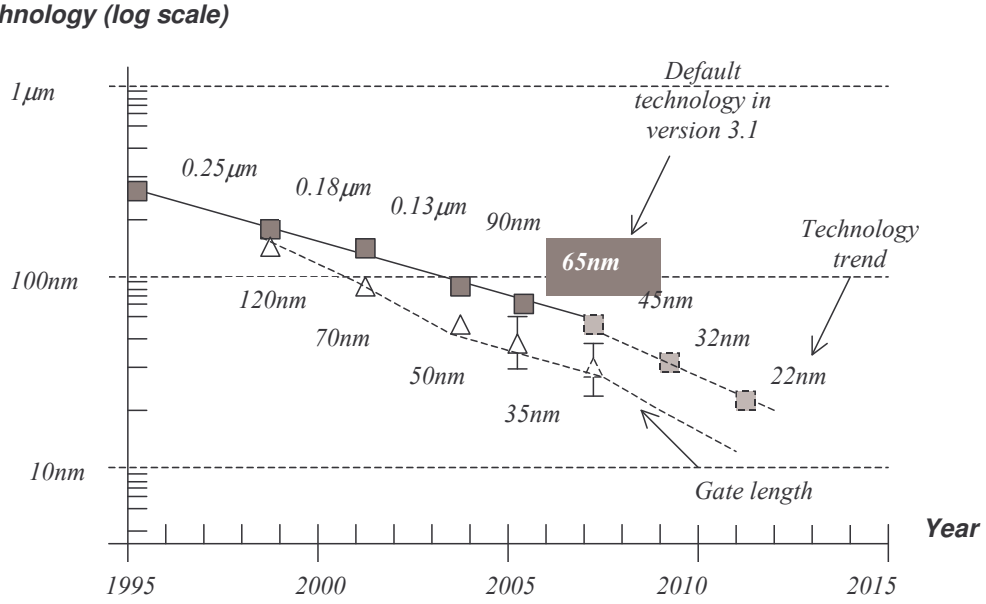


Figure 1-2 : The technology scale down towards nano-scale devices

Market Growth

The integrated circuit market has been growing steadily since many years, due to ever-increased demand for electronic devices. The production of integrated circuits for various technologies is illustrated over the years in Figure 1-3. It can be seen that a new technology has appeared regularly each two years, with a ramp up close to three years [Ghani]. The production peak is constantly increased, and similar trends should be observed for novel technologies such as 65nm (forecast peak in 2009).

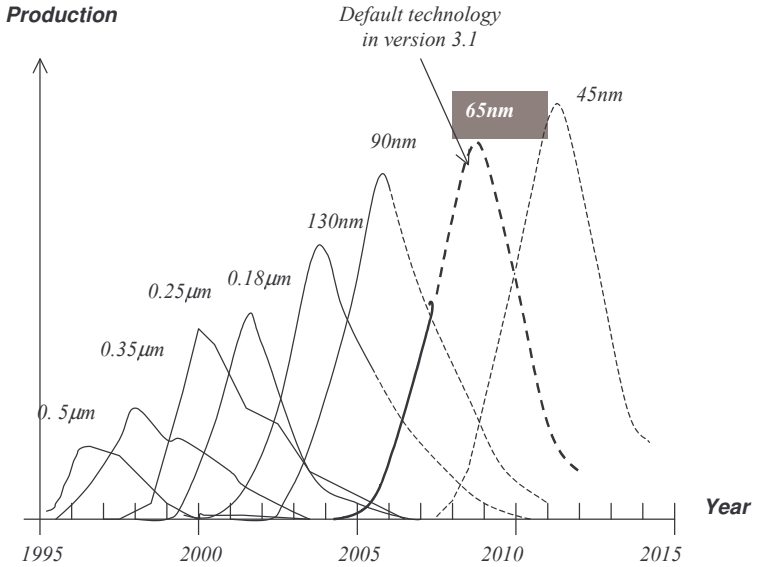


Figure 1-3: Technology ramping every two years

About 65 nm technology

Industrial 65-nm processes have been introduced in 2003-2004 time frame. The industrial 65-nm process features the introduction of uniaxial tensile strain in the NMOS transistor channel and uniaxial compressive strain in the PMOS transistor channel resulting in dramatic improvements in channel mobility and consequently drive currents. More information about the implementation of the 65-nm technology may be found in the MICROWIND application node [Sicard 2006].

65nm technology variants

There may exist several variants of the 65-nm process technology. One corresponds to the highest possible speed, at the price of a very high leakage current. This technology is called "High speed" as it is dedicated to applications for which the highest speed is the primary objective: fast microprocessors, fast DSP, etc. The second technological option called "General Purpose" (Figure 1-4) is targeted to standard products where the speed factor is not critical. The leakage current is one order of magnitude lower than for the high-speed variant, with gate switching decreased by 50%. This technology option has been addressed in Microwind's 65nm rule file [Sicard 2006a].

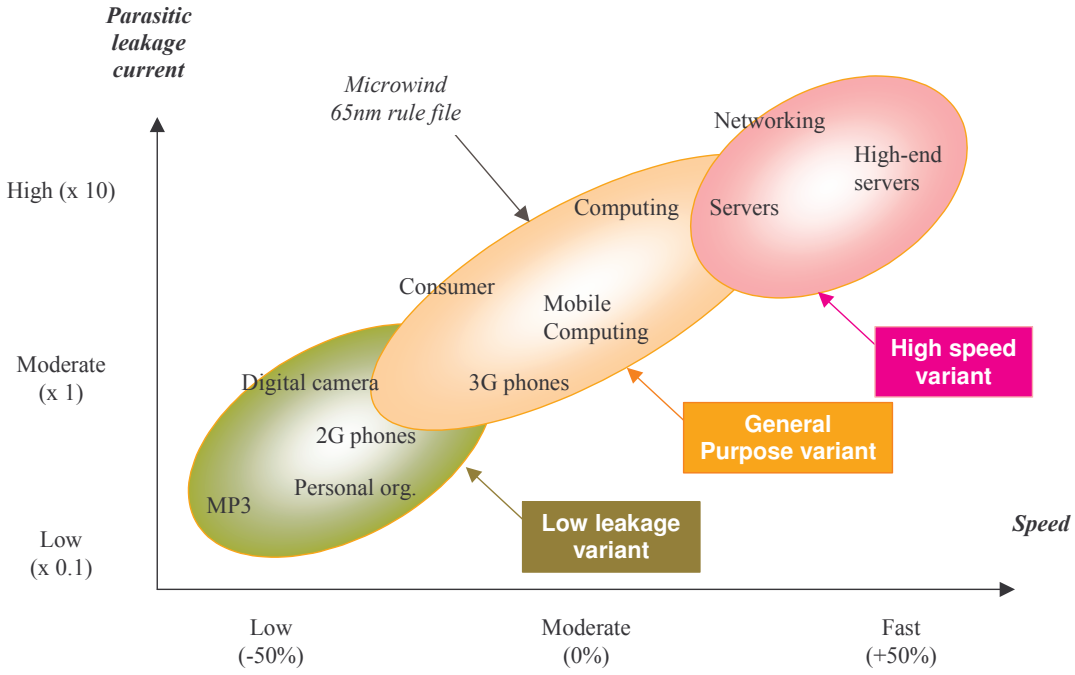


Figure 1-4 : Introducing three variants of the 65nm technology

There may also exist a third variant called low leakage (bottom left of figure 1-4). This variant concerns integrated circuits for which the leakage must remain as low as possible, a criterion that ranks first in applications such as embedded devices, mobile phones or personal organizers.

The operational voltage is usually from 0.85V to 1.2V, depending on the technology variant. In Microwind, we decided to fix VDD at 1.0V in the cmos65nm.RUL rule file, which represents a compromise between all possible technology variations available for this 65-nm node.

Gate Dielectric

Continued thickness reduction of conventional oxides such as silicon dioxide (SiO_2) results in reliability degradation and unacceptable current leakage. New dielectric materials with high permittivity (High-"K") are needed to replace SiO2, both for the MOS device itself and the embedded capacitors. Silicon oxynitride (SiOxNy) has been proposed in 65nm technologies as an effective replacement to SiO2 as the dielectric material in metal-oxide-semiconductor devices (figure 1-5).

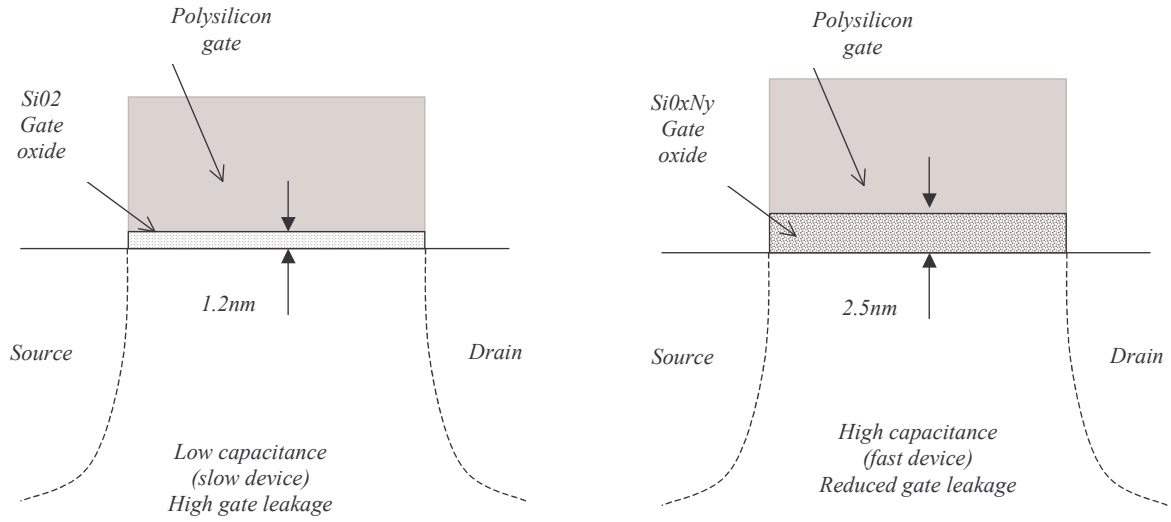


Figure 1-5 : The new gate oxide material enhances the MOS device performances in terms of switching speed and leakage

Strained Silicon

The main novelty related to the 90 and 65 nm technology is the introduction of strained silicon to speed-up the carrier mobility, which boosts both the n-channel and p-channel transistor performances. In Intel's the 65 nm technology [Bai] the channel strains in both NMOS and PMOS devices have been improved over 90 nm technology [Ghani]. PMOS transistor channel strain has been enhanced by increasing the Ge content in the compressive SiGe film. Both transistors employ ultra shallow source-drains to further increase the drive currents.

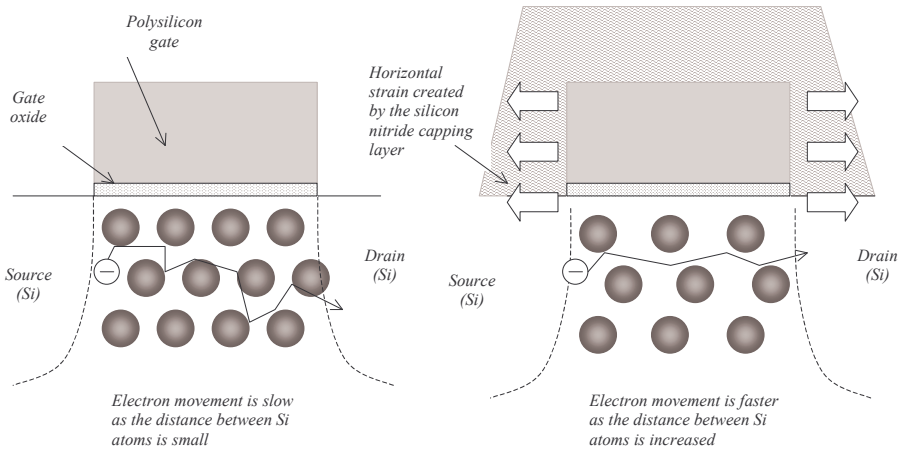


Figure 1-6 : Strain generated by a silicon-nitride capping layer which increases the distance between atoms underneath the gate, which speeds up the electron mobility of n-channel MOS devices

Let us the silicon atoms forming a regular lattice structure, inside which the electrons participating to the device current have to flow. In the case of electron carriers, stretching the lattice allows the charges to flow faster from the drain to the source, as depicted in Fig. 1-6.

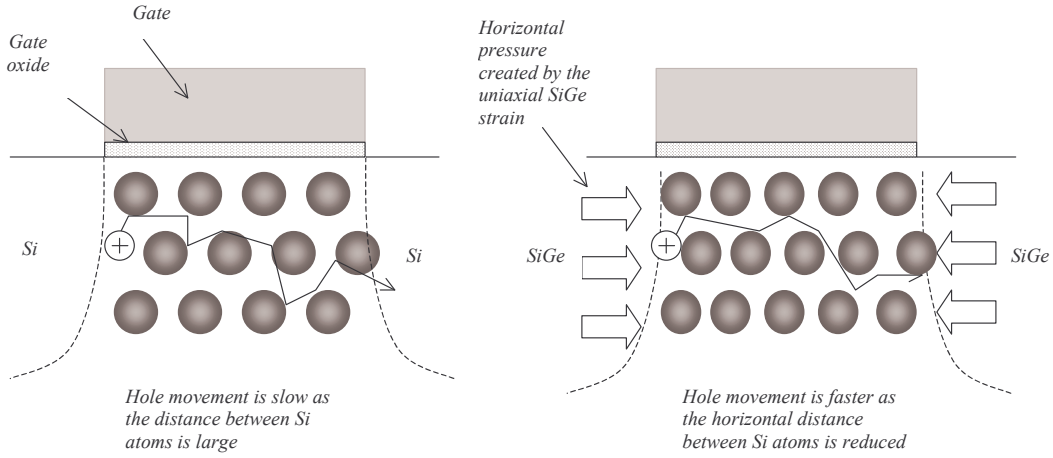


Figure 1-7 : Compressive stain to reduce the distance between atoms underneath the gate, which speeds up the hole mobility of p-channel MOS devices

The mobility improvement exhibits a linear dependence with the tensile film thickness. A 80 nm film has resulted in a 10% saturation current improvement in Intel's 90nm technology [2]. The strain may also be applied from the bottom with a uniform layer of an alloy of silicon and germanium (SiGe).
In a similar way, compressing the lattice slightly speeds up the p-type transistor, for which current carriers consist of holes (Fig. 6). The combination of reduced channel length, decreased oxide thickness and strained silicon achieves a substantial gain in drive current for both nMOS and pMOS devices.

2 The MOS device

This chapter presents the CMOS transistor, its layout, static characteristics and dynamic characteristics. The vertical aspect of the device and the three dimensional sketch of the fabrication are also described.

Logic Levels

Three logic levels 0,1 and X are defined as follows:

Logical value	Voltage	Name	Symbol in DSCH	Symbol in MICROWIND
0	0.0V	VSS	(Green in logic simulation)	(Green in analog simulation)
1	1.0V in cmos 65nm	VDD	(Red in logic simulation)	(Red in analog simulation)
X	Undefined	X	(Gray in simulation)	(Gray in simulation)

The MOS as a switch

The MOS transistor is basically a switch. When used in logic cell design, it can be *on* or *off*. When *on*, a current can flow between drain and source. When off, no current flow between drain and source. The MOS is turned on or off depending on the gate voltage. In CMOS technology, both n-channel (or nMOS) and p-channel MOS (or pMOS) devices exist. The nMOS and pMOS symbols are reported below. The symbols for the ground voltage source (0 or VSS) and the supply (1 or VDD) are also reported in figure 2-1.

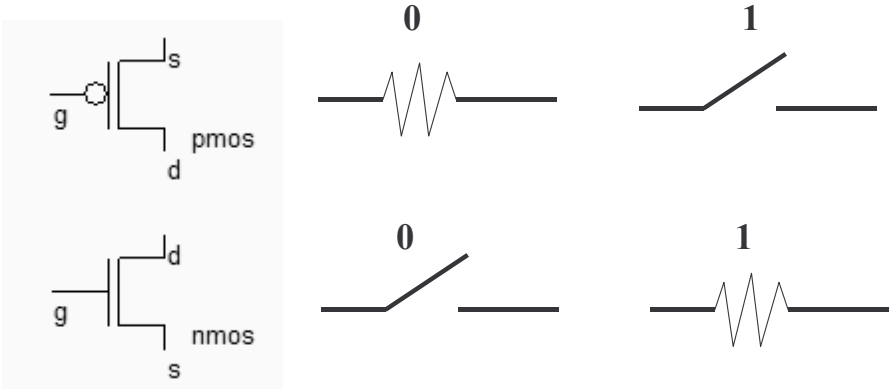


Figure 2-1 : the MOS symbol and switch

The n-channel MOS device requires a logic value 1 (or a supply VDD) to be on. In contrary, the p-channel MOS device requires a logic value 0 to be on. When the MSO device is on, the link between the source and drain is equivalent to a resistance. The order of range of this 'on' resistance is 100Ω-5KΩ. The 'off' resistance is considered infinite at first order, as its value is several MΩ.

MOS layout

We use MICROWIND to draw the MOS layout and simulate its behavior. Go to the directory in which the software has been copied (By default `Microwind31`). Double-click on the MICROWIND icon.
The MICROWIND display window includes four main windows: the main menu, the layout display window, the icon menu and the layer palette. The layout window features a grid, scaled in lambda (λ) units. The lambda unit is fixed to half of the minimum available lithography of the technology. The default technology is a CMOS 8-metal layers 65nm technology. In this technology, lambda is 0.035µm (35nm).

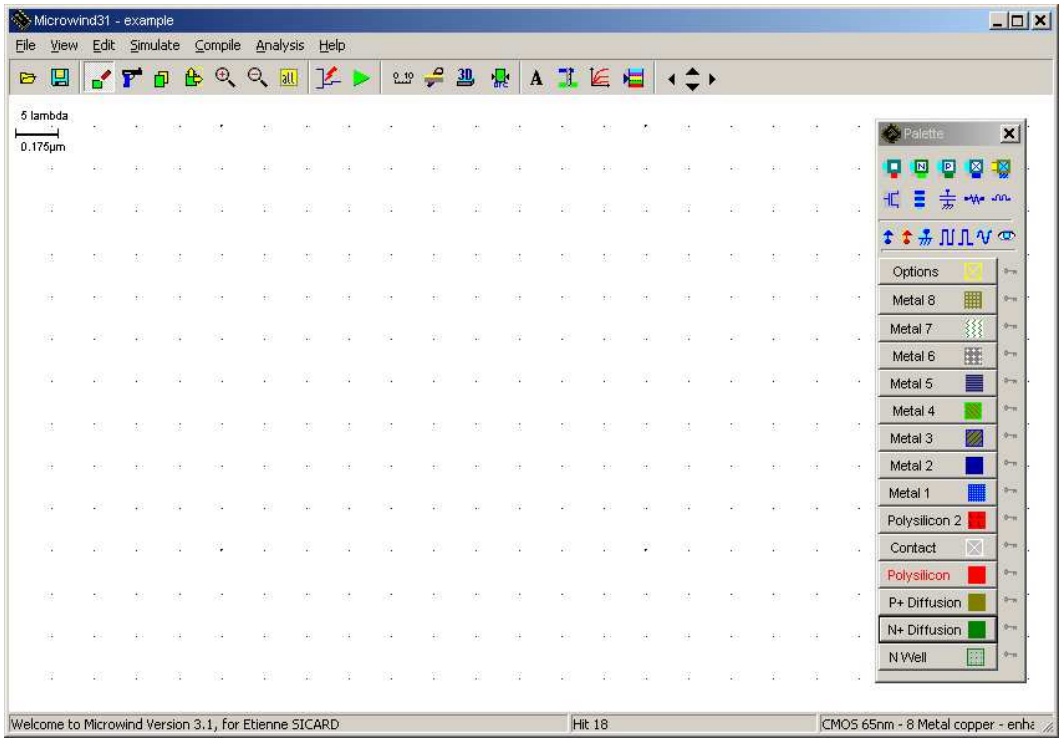


Figure 2-2 :The MICROWIND window as it appears at the initialization stage..

The palette is located in the lower right corner of the screen. A red color indicates the current layer. Initially the selected layer in the palette is polysilicon. By using the following procedure, you can create a manual design of the n-channel MOS.

❶ Fix the first corner of the box with the mouse. While keeping the mouse button pressed, move the mouse to the opposite corner of the box. Release the button. This creates a box in polysilicon layer as shown in Figure 2-3. The box width should not be inferior to 2 λ, which is the minimum width of the polysilicon box.

❷ Change the current layer into N+ diffusion by a click on the palette of the Diffusion N+ button. Make sure that the red layer is now the N+ Diffusion. Draw a n-diffusion box at the bottom of the drawing as in Figure 2-3. N-diffusion boxes are represented in green. The intersection between diffusion and polysilicon creates the channel of the nMOS device.

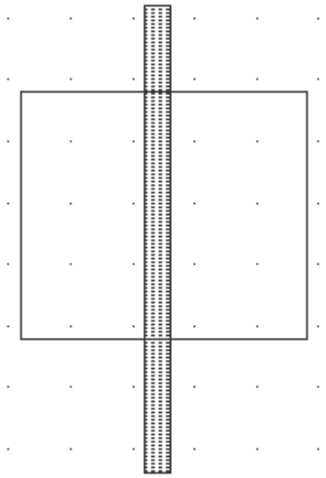


Figure 2-3 : Creating the N-channel MOS transistor

Vertical aspect of the MOS

Click on this icon to access *process simulation* (Command ***Simulate → Process section in 2D***). The cross-section is given by a click of the mouse at the first point and the release of the mouse at the second point.

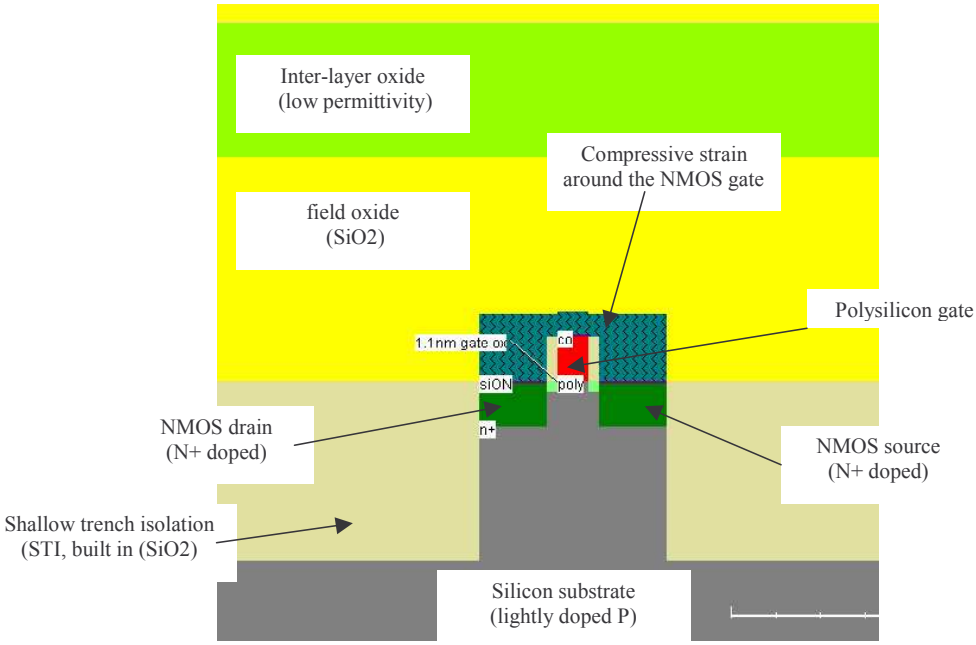


Figure 2-4 : The cross-section of the nMOS devices.

In the example of Figure 2-4, three nodes appear in the cross-section of the n-channel MOS device: the gate (red), the left diffusion called *source* (green) and the right diffusion called *drain* (green), over a substrate (gray). A thin oxide called the gate oxide isolates the gate. Various steps of oxidation have lead to stacked oxides on the top of the gate.

The physical properties of the source and of the drain are exactly the same. Theoretically, the source is the origin of channel impurities. In the case of this nMOS device, the channel impurities are the electrons. Therefore, the source is the diffusion area with the lowest voltage. The polysilicon gate floats over the channel, and splits the diffusion into 2 zones, the source and the drain. The gate controls the current flow from the drain to the source, both ways. A high voltage on the gate attracts electrons below the gate, creates an electron channel and enables current to flow. A low voltage disables the channel.

Static Mos Characteristics

Click on the *MOS characteristics* icon. The screen shown in Figure 2-5 appears. It represents the *Id/Vd* static characteristics of the nMOS device. The MOS size (width and length of the channel situated at the intersection of the polysilicon gate and the diffusion) has a strong influence on the value of the current. In Figure 2-5, the MOS width is 350 nm and the length is 70 nm. A high gate voltage (*Vg* = 1.0V) corresponds to the highest *Id/Vd* curve. For *Vg*=0, no current flows. You may change the voltage values of *Vd, Vg, Vs* by using the voltage cursors situated on the right side of the window.

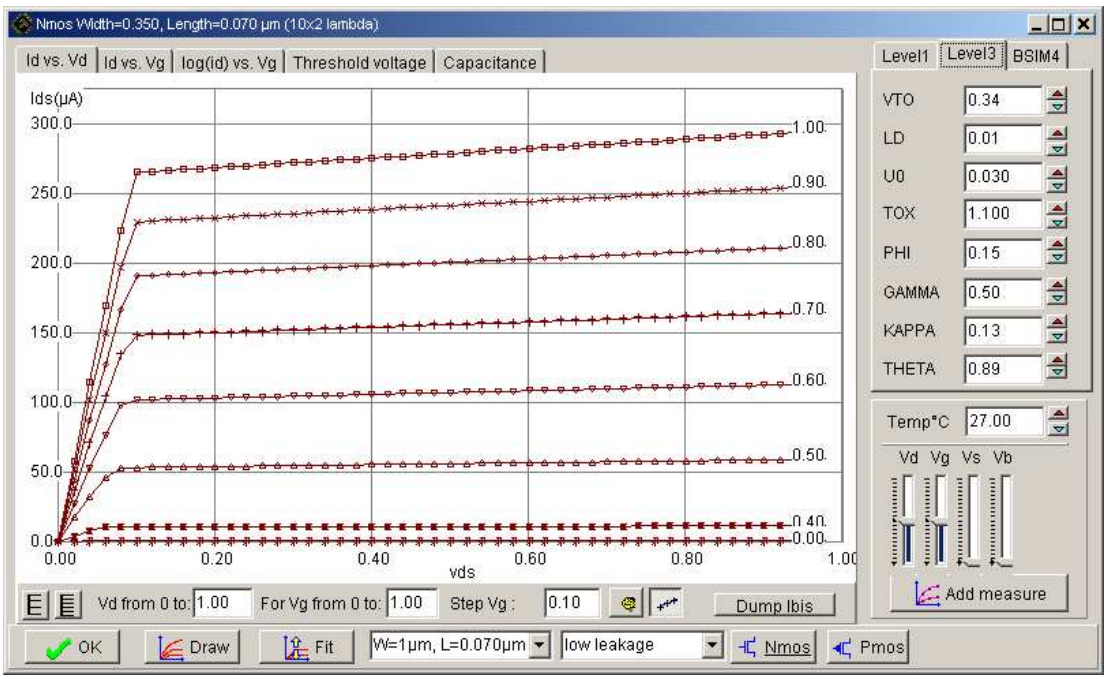
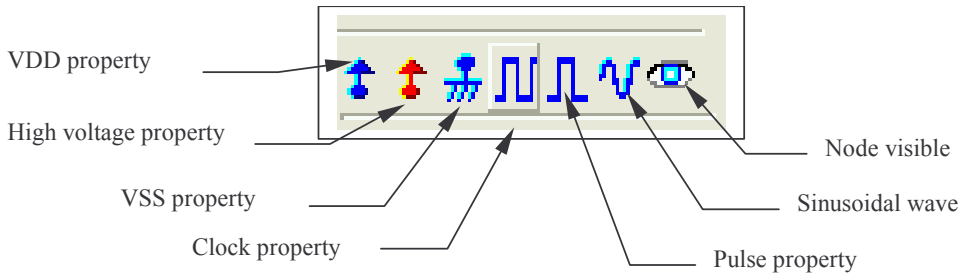


Figure 2-5 : N-Channel MOS characteristics.

A maximum current around 0.3 mA is obtained for *Vg*=1.0 V, *Vd*=1.0 V, with *Vs*=0.0. The MOS parameters correspond to SPICE Level 3. A tutorial on MOS model parameters is proposed later in this chapter.

Dynamic MOS behavior

This paragraph concerns the dynamic simulation of the MOS to exhibit its switching properties. The most convenient way to operate the MOS is to apply a clock to the gate, another to the source and to observe the drain. The summary of available properties that can be added to the layout is reported below.



❶ Apply a clock to the gate. Click on the *Clock* icon and then, click on the polysilicon gate. The clock menu appears again. Change the name into `Vgate` and click on **OK** to apply a clock with 0.2 ns period (90 ps at "0", 10 ps rise, 90 ps at "1", 10 ps fall).

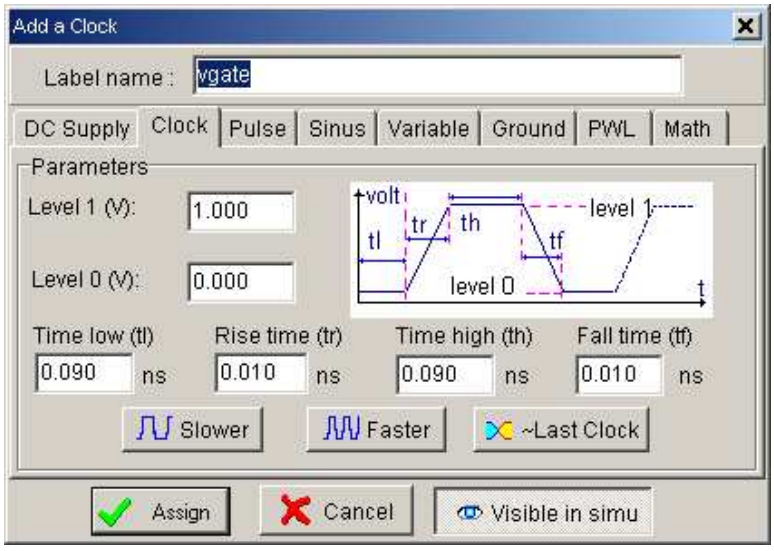


Figure 2-6 : The clock menu.

❷ Apply a clock to the drain. Click on the Clock icon, click on the left diffusion. The *Clock* menu appears. Change the name into `Vdrain` and click on **OK**. A default clock with 1ns period is generated. The *Clock* property is sent to the node and appears at the right hand side of the desired location with the name `Vdrain`.

❸ Watch the output: Click on the *Visible* icon and then, click on the right diffusion. Click **OK**. The Visible property is then sent to the node. The associated text `s1` is in italic, meaning that the waveform of this node will appear at the next simulation.

Always save BEFORE any simulation. The analog simulation algorithm may cause run-time errors leading to a loss of layout information. Click on **File → Save as**. A new window appears, into which you enter the design name. Type for example `myMos`. Then click on **Save**. The design is saved under that filename.

Analog Simulation

Click on **Simulate → Start Simulation**. The timing diagrams of the nMOS device appear, as shown in Figure 2-7.

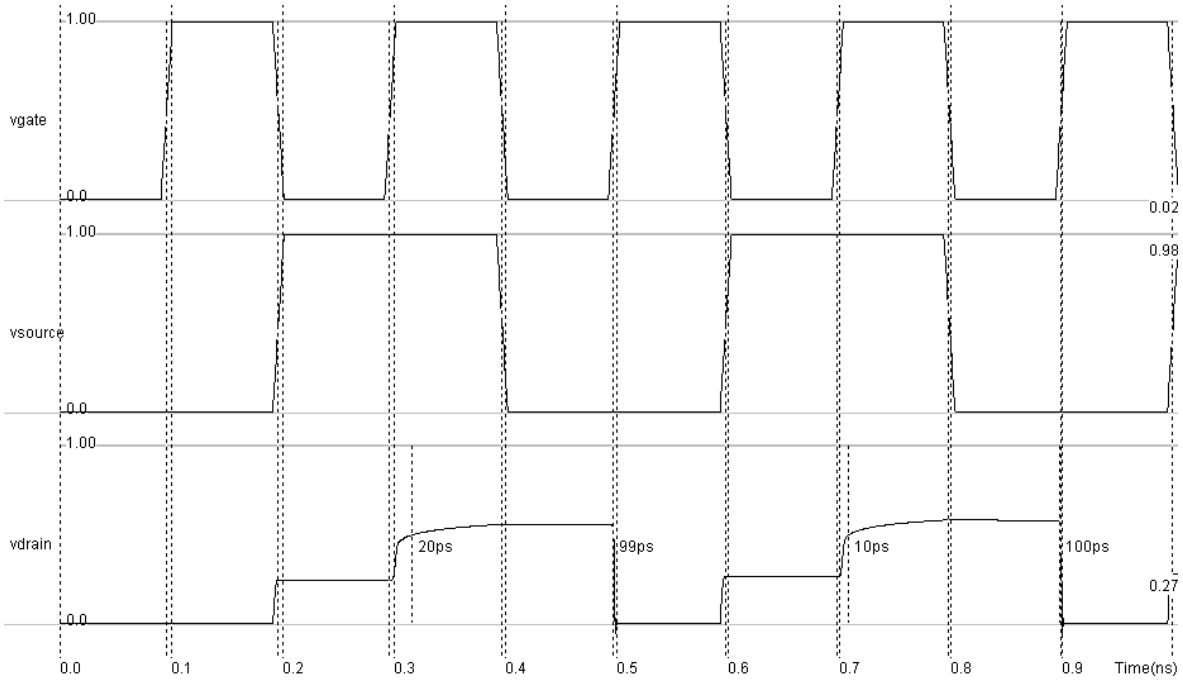


Figure 2-7 : Analog simulation of the MOS device.

When *vgate* is at zero, no channel exists so the node *vsource* is disconnected from the drain. When the gate is on (*vgate*=1.0 V), the source copies the drain. It can be observed that the nMOS device drives well at zero but poorly at the high voltage. The highest value of *vsource* is around 0.6 V, that is VDD minus the threshold voltage. This means that the n-channel MOS device do not drives well logic signal 1, as summarized in figure 2-8. Click on **More** in order to perform more simulations. Click on **Close** to return to the editor.

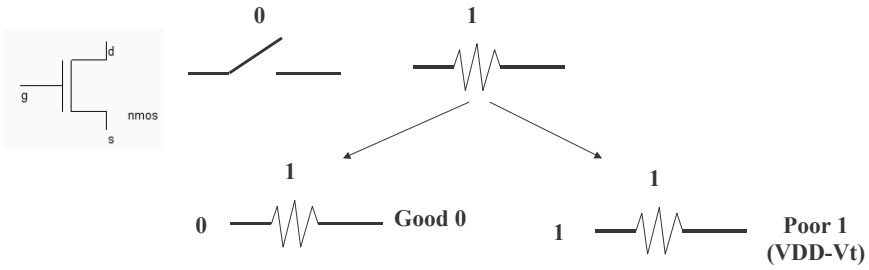


Figure 2-8 : The nMOS device behavior summary

The MOS Models

Mos Level 1

For the evaluation of the current *Ids* between the drain and the source as a function of *Vd,Vg* and *Vs*, you may use the old but nevertheless simple LEVEL1 described below. The parameters listed in table 2-1 correspond to "low leakage" MOS option, which is the default MOS option in 65nm technology. When dealing with sub-micron technology, the model LEVEL1 is more than 4 times too optimistic regarding current prediction, compared to real-case measurements.

$\varepsilon_0 = 8.85\ 10^{-12}$ F/m is the absolute permittivity

ε_r = relative permittivity, equal to 3.9 in the case of SiO2 (no unit)

Mode	Condition	Expression for the current *Ids*
CUT-OFF	Vgs<0	$Ids = 0$
LINEAR	Vds<Vgs-Vt	$Ids = \mathrm{UO}\frac{\varepsilon_0\varepsilon_r}{\mathrm{TOX}}.\frac{\mathrm{W}}{\mathrm{L}}((V_{gs} - vt).V_{ds} - \frac{(V_{ds})^2}{2}))$
SATURATED	Vds>Vgs-Vt	$Ids = \mathrm{UO}\frac{\varepsilon_0\varepsilon_r}{\mathrm{TOX}}.\frac{\mathrm{W}}{\mathrm{L}}(V_{gs} - vt)^2$

Mos Level1 parameters			
Parameter	Definition	Typical Value 65nm NMOS	PMOS
VTO	Threshold voltage	0.34 V	-0.34 V
U0	Carrier mobility	0.03 m^2/V-s	0.01 m^2/V-s
TOX	Gate oxide thickness	1.1 nm	1.1 nm
PHI	Surface potential at strong inversion	0.15 V	0.15 V
GAMMA	Bulk threshold parameter	0.5 $V^{0.5}$	0.4 $V^{0.5}$
W	MOS channel width	400 nm	400 nm
L	MOS channel length	70 nm	70 nm

Table 2-1: Parameters of MOS level 1 implemented into Microwind

The MOS Level 3

For the evaluation of the current *Ids* as a function of *Vd,Vg* and *Vs* between drain and source, we commonly use the following equations, close from the SPICE LEVEL 3 formulations [Lee]. The formulations are derived from the LEVEL1 and take into account a set of physical limitations in a semi-empirical way.

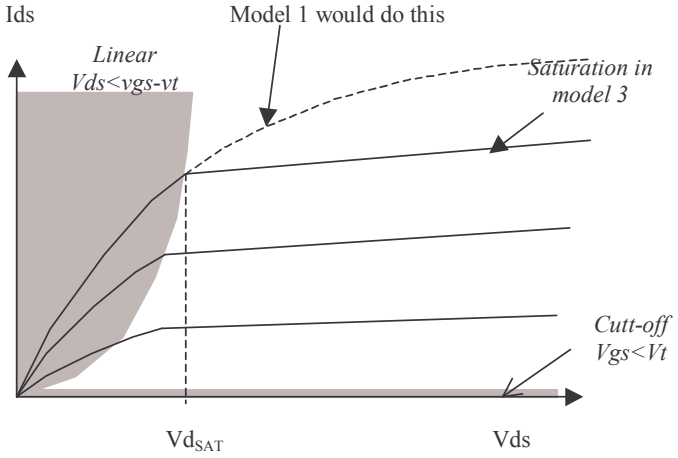


Figure 2-9 : Introduction of the saturation voltage VdSat which truncates the equations issued from model 1

One of the most important change is the introduction of Vd_{SAT}, a saturation voltage from which the current saturates and do not rise as the LEVEL1 model would do (figure 2-9). This saturation effect is significant for small channel length.

The BSIM4 MOS Model

A new MOS model, called BSIM4, has been introduced in 2000 [Liu]. A simplified version of this model is supported by MICROWIND in its full version and recommended for nanoscale technology simulation. BSIM4 still considers the operating regions described in MOS level 3 (linear for low *Vds*, saturated for high *Vds*, subthreshold for *Vgs<Vt*), but provides a perfect continuity between these regions. BSIM4 introduces a new region where the impact ionization effect is dominant.

The number of parameters specified in the official release of BSIM4 is as high as 300. A significant portion of these parameters is unused in our implementation. We concentrate on the most significant parameters, for educational purpose. The set of parameters is reduced to around 20, shown in the right part of figure 2-10.

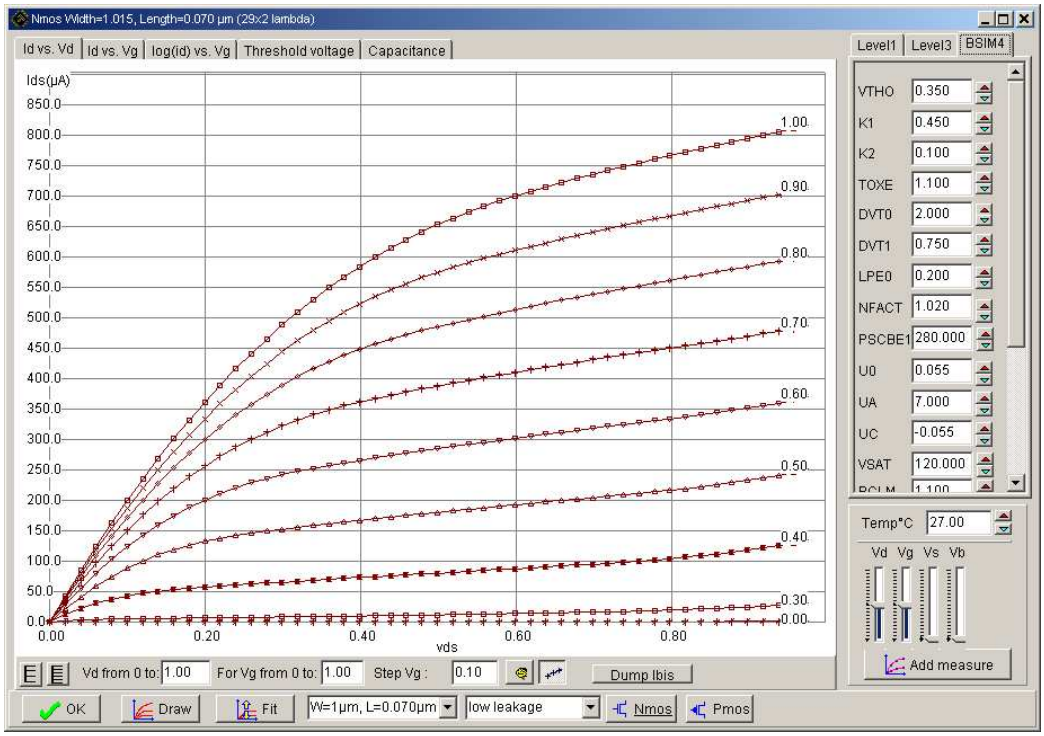


Figure 2-10 : Implementation of BSIM4 within Microwind (full version only)

The PMOS Transistor

The p-channel transistor simulation features the same functions as the n-channel device, but with opposite voltage control of the gate. For the nMOS, the channel is created with a logic 1 on the gate. For the pMOS, the channel is created for a logic 0 on the gate. Load the file `pmos.msk` and click the icon **MOS characteristics**. The p-channel MOS simulation appears, as shown in Figure 2-11.

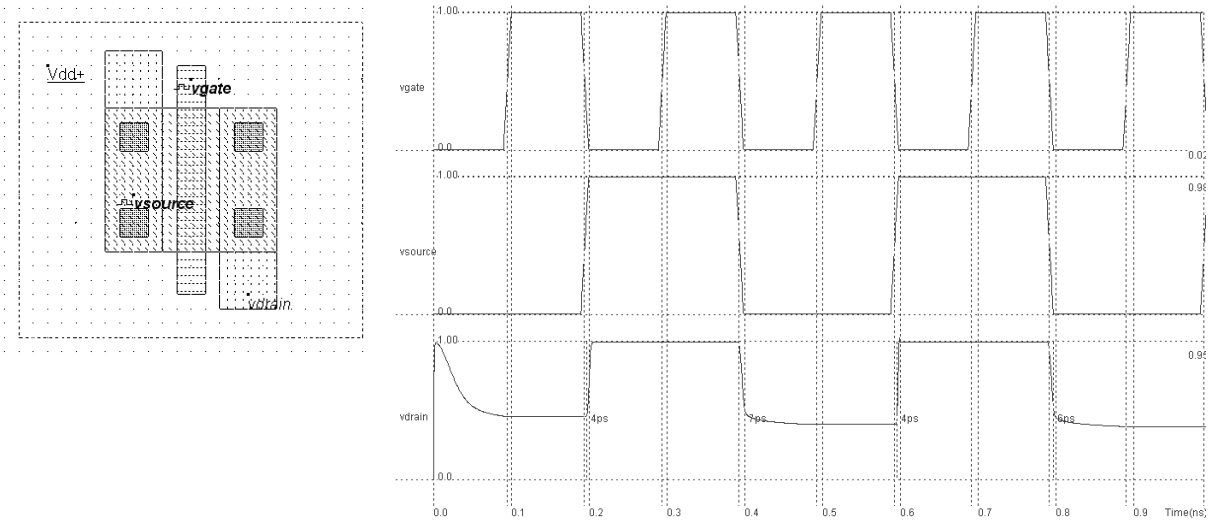


Figure 2-11 : Layout and simulation of the p-channel MOS (mypmos.MSK)

Note that the pMOS gives approximately half of the maximum current given by the nMOS with the same device size. The highest current is obtained with the lowest possible gate voltage, that is 0. From the simulation of figure 2-11, we see that the pMOS device is able to pass well the logic level 1. But the logic level 0 is transformed into a positive voltage, equal to the threshold voltage of the MOS device (0.35 V). The summary of the p-channel MOS performances is reported in figure 2-12.

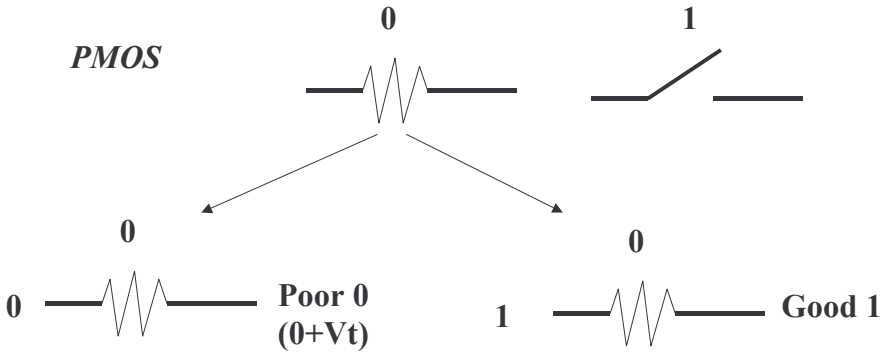


Figure 2-12 : Summary of the performances of a pMOS device

The Transmission Gate

Both NMOS devices and PMOS devices exhibit poor performances when transmitting one particular logic information. The nMOS degrades the logic level 1, the pMOS degrades the logic level 0. Thus, a perfect pass gate can be constructed from the combination of nMOS and pMOS devices working in a complementary way, leading to improved switching performances. Such a circuit, presented in figure 2-13, is called the transmission gate. In DSCH , the symbol may be found in the **Advance** menu in the palette. The transmission gate includes one inverter, one nMOS and one pMOS.

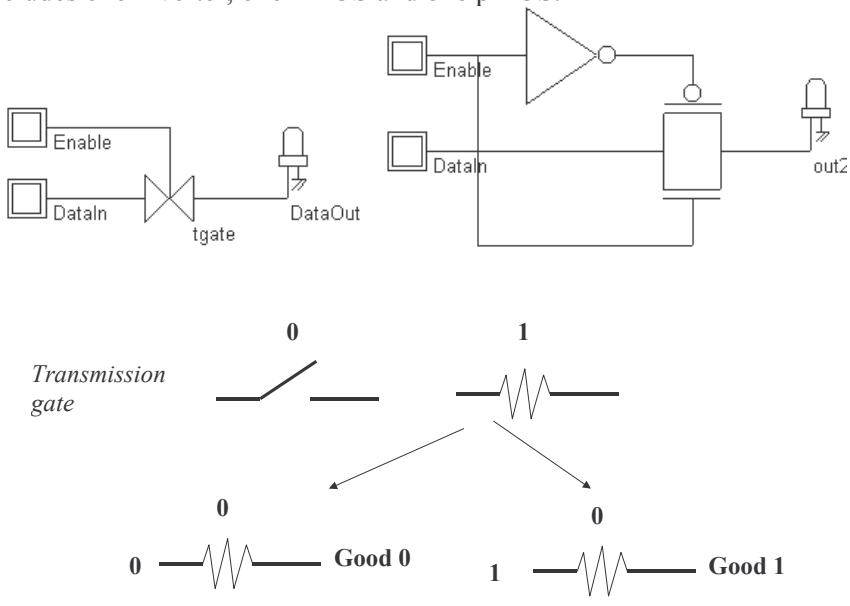


Figure 2-13 : Schematic diagram of the transmission gate (Tgate.SCH)

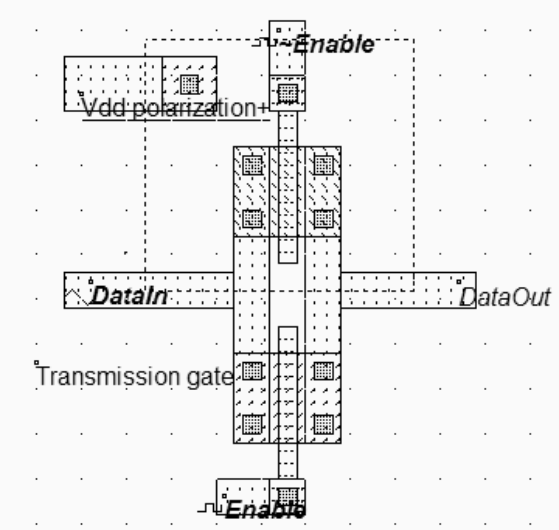


Figure 2-14 : Layout of the transmission gate (TGATE.MSK)

The layout of the transmission gate is reported in figure 2-14. The n-channel MOS is situated on the bottom the p-channel MOS on the top. Notice that the gate controls are not connected, as *~Enable* is the opposite of *Enable*.

Added Features in the full version

BSIM4	The state-of-the art MOS model for accurate simulation of nano-scale technologies, including a tutorial on key parameters of the model.
High Speed Mos	A new kind of MOS device has been introduced in deep submicron technologies, starting the 0.18µm CMOS process generation. The new MOS, called high speed MOS (HS) is available as well as the normal one, recalled Low leakage MOS (LL).
High Voltage MOS	For I/Os operating at high voltage, specific MOS devices called "High voltage MOS" are used. The high voltage MOS is built using a thick oxide, two to three times thicker than the low voltage MOS, to handle high voltages as required by the I/O interfaces..
Temperature Effects	Three main parameters are concerned by the sensitivity to temperature: the threshold voltage `VTO`, the mobility `U0` and the slope in sub-threshold mode. The modeling of the temperature effect is described and illustrated .
Process Variations	Due to unavoidable process variations during the hundreds of chemical steps for the fabrication of the integrated circuit, the MOS characteristics are never exactly identical from one device to another, and from one die to an other. Monte-carlo simulation, min/max/typ simulations are provided in the full version.

3 The Inverter

This chapter describes the CMOS inverter at logic level, using the logic editor and simulator DSCH , and at layout level, using the tool MICROWIND .

The Logic Inverter

In this section, an inverter circuit is loaded and simulated. Click **File→ Open** in the main menu. Select `INV.SCH` in the list. In this circuit are one button situated on the left side of the design, the inverter and a led. Click **Simulate→ Start simulation** in the main menu.

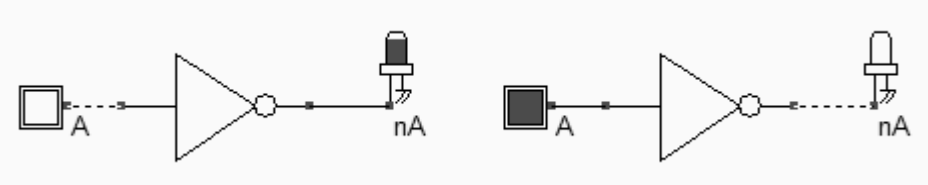


Figure 3-1 : The schematic diagram including one single inverter (Inverter.SCH)

Now, click inside the buttons situated on the left part of the diagram. The result is displayed on the leds. The red value indicates logic 1, the black value means a logic 0. Click the button **Stop simulation** shown in the picture below. You are back to the editor.

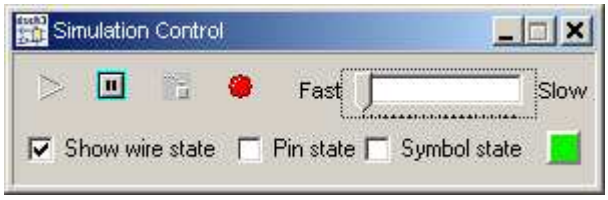


Figure 3-2 : The button Stop Simulation

Click the **chronogram** icon to get access to the chronograms of the previous simulation (Figure 3-3). As seen in the waveform, the value of the output is the logic opposite of that of the input.

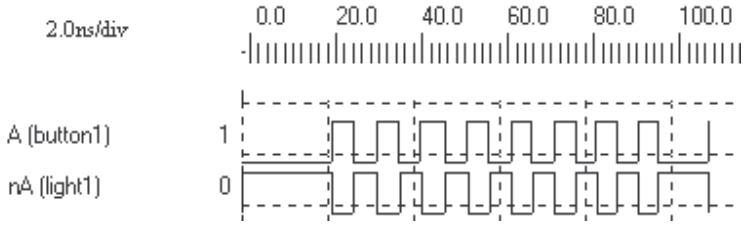


Figure 3-3 : Chronograms of the inverter simulation (CmosInv.SCH)

Double click on the INV symbol, the symbol properties window is activated. In this window appears the VERILOG description (left side) and the list of pins (right side). A set of drawing options is also reported in the same window. Notice the gate delay (3 pico-second in the 65-nm technology), the fanout that represents the number of cells connected to the output pin (1 cell connected), and the wire delay due to this cell connection (an extra 2 ps delay).

The CMOS inverter

The CMOS inverter design is detailed in the figure below. Here the p-channel MOS and the n-channel MOS transistors function as switches. When the input signal is logic 0 (Figure 3-4 left), the nMOS is switched off while PMOS passes VDD through the output. When the input signal is logic 1 (Figure 3-4 right), the pMOS is switched off while the nMOS passes VSS to the output.

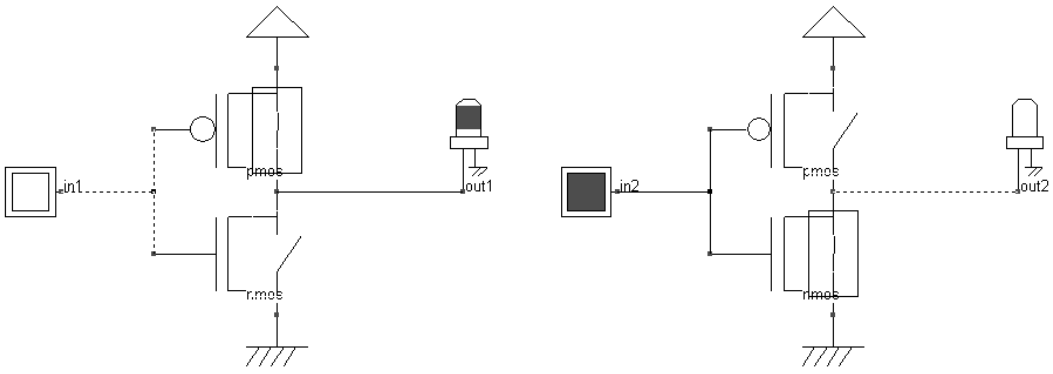


Figure 3-4 : The MOS Inverter (File CmosInv.sch)

The fanout corresponds to the number of gates connected to the inverter output. Physically, a large fanout means a large number of connections, that is a large load capacitance. If we simulate an inverter loaded with one single output, the switching delay is small. Now, if we load the inverter by several outputs, the delay and the power consumption are increased. The power consumption linearly increases with the load capacitance. This is mainly due to the current needed to charge and discharge that capacitance.

Manual Layout of the Inverter

In this paragraph, the procedure to create manually the layout of a CMOS inverter is described. Click the icon **MOS generator** on the palette. The following window appears. By default the proposed length is the minimum length available in the technology (2 lambda), and the width is 10 lambda. In 0.12µm technology, where lambda is 0.06µm, the corresponding size is 0.12µm for the length and 0.6µm for the width. Simply click **Generate Device**, and click on the middle of the screen to fix the MOS device.

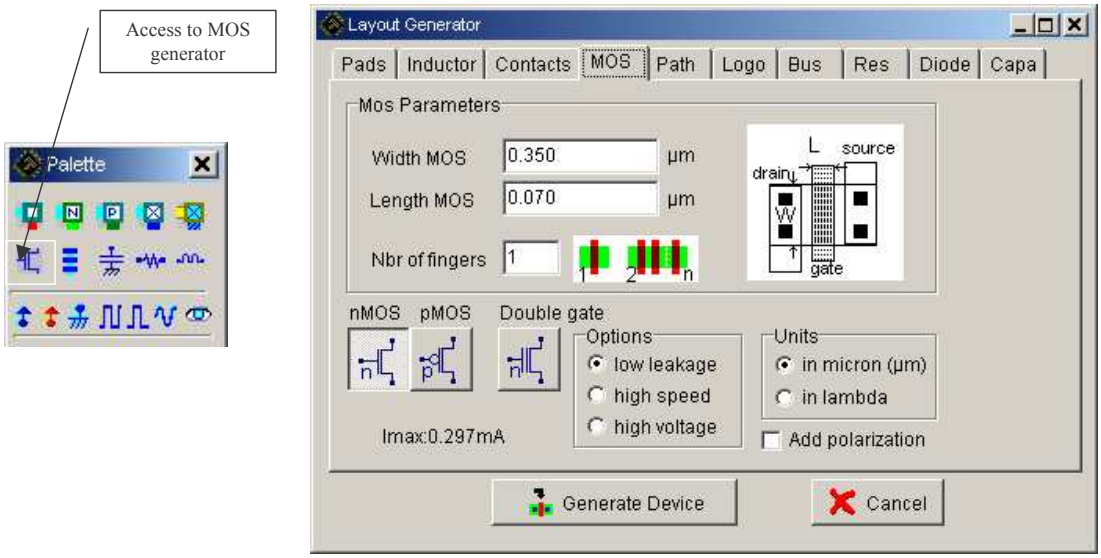


Figure 3-5 : Generating a nMOS device

Click again the icon **MOS generator** on the palette. Change the type of device by a tick on **p-channel**, and click **Generate Device**. Click on the top of the nMOS to fix the pMOS device.

The MOS generator is the safest way to create a MOS device compliant to design rules. The programmable parameters are the MOS width, length, the number of gates in parallel and the type of device (n-channel or p-channel). By default metal interconnects and contacts are added to the drain and source of the MOS. You may add a supplementary metal2 interconnect on the top of metal 1 for drain and source.

Connection between Devices

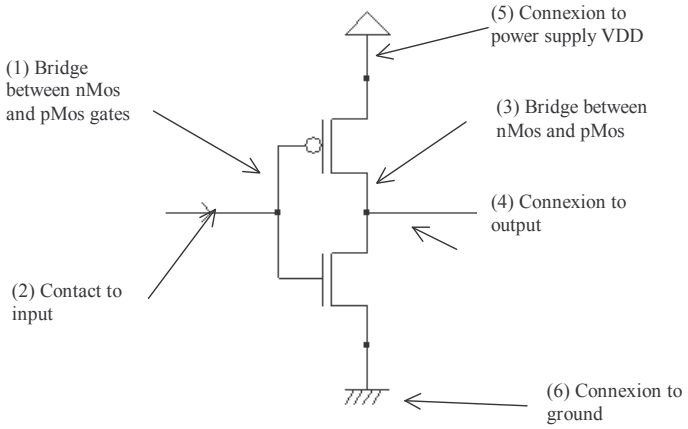


Figure 3-6 : Connections required to build the inverter (CmosInv.SCH)

Within CMOS cells, metal and polysilicon are used as interconnects for signals. Metal is a much better conductor than polysilicon. Consequently, polysilicon is only used to interconnect gates, such as the bridge (1) between pMOS and nMOS gates, as described in the schematic diagram of figure 3-6. Polysilicon is rarely used for long interconnects, except if a huge resistance value is expected.

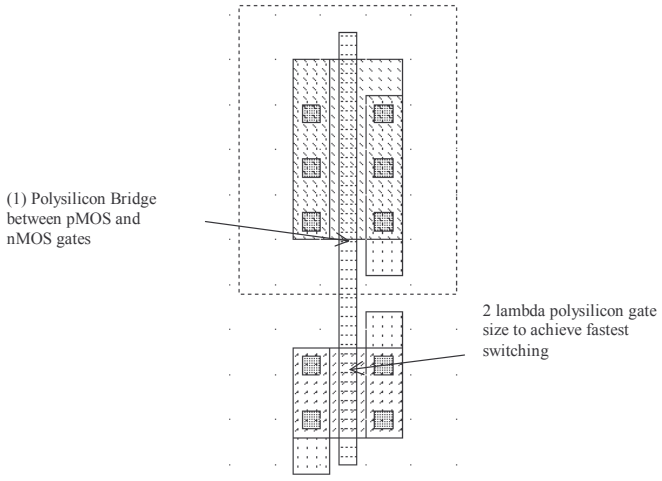


Figure 3-7 : Polysilicon bridge between nMOS and pMOS devices (InvSteps.MSK)

In the layout shown in figure 3-7, the polysilicon bridge links the gate of the n-channel MOS with the gate of the p-channel MOS device. The polysilicon serves as the gate control and the bridge between MOS gates.

Useful Editing Tools

The following commands may help you in the layout design and verification processes.

Command	Icon/Short cut	Menu	Description
UNDO	CTRL+U	Edit menu	Cancels the last editing operation
DELETE	 CTRL+X	Edit menu	Erases some layout included in the given area or pointed by the mouse.
STRETCH		Edit menu	Changes the size of one box, or moves the layout included in the given area.
COPY	 CTRL+C	Edit Menu	Copies the layout included in the given area.
VIEW ELECTRICAL NODE	 CTRL+N	View Menu	Verifies the electrical net connections.
2D CROSS-SECTION		Simulate Menu	Shows the aspect of the circuit in vertical cross-section.

Table 3-1: A set of useful editing tools

Metal-to-poly

As polysilicon is a poor conductor, metal is preferred to interconnect signals and supplies. Consequently, the input connection of the inverter is made with metal. Metal and polysilicon are separated by an oxide which prevents electrical connections. Therefore, a box of metal drawn across a box of polysilicon does not allow an electrical connection (Figure 3-8).

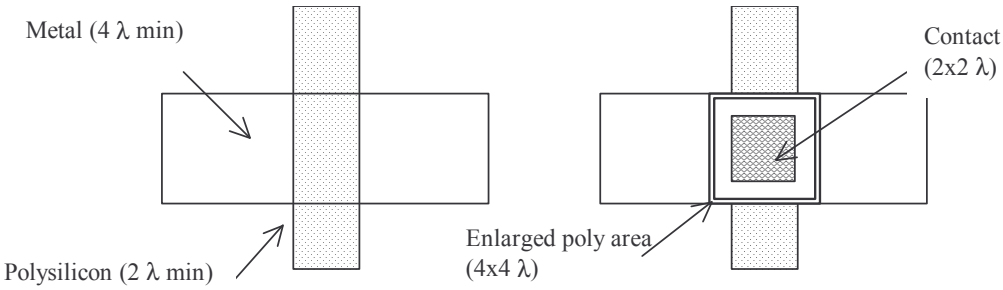


Figure 3-8 : Physical contact between metal and polysilicon

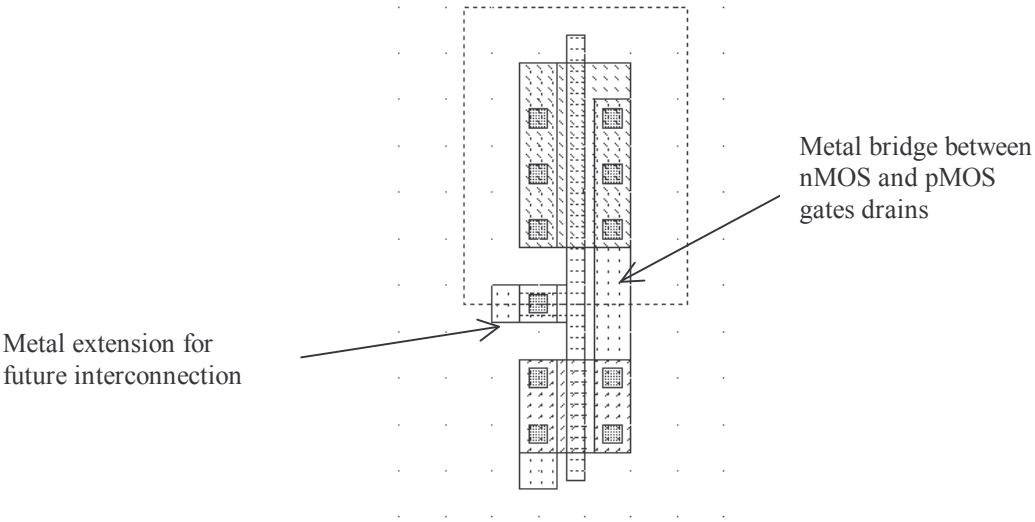


Figure 3-9 : Adding a poly contact, poly and metal bridges to construct the CMOS inverter (InvSteps.MSK)

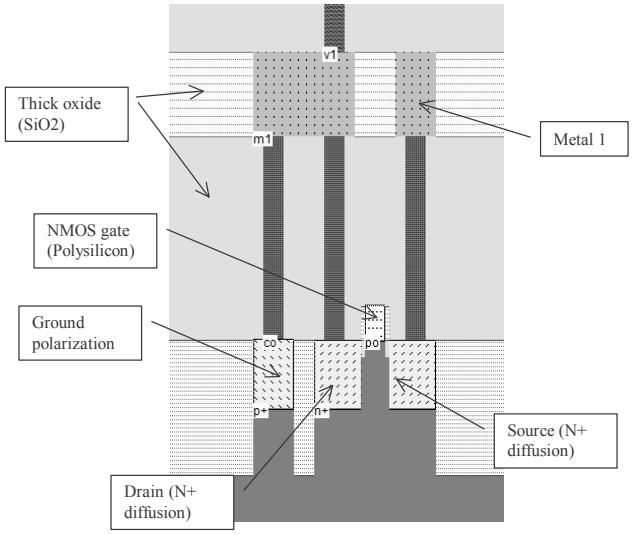


Figure 3-10 : The 2D process section of the inverter circuit near the nMOS device (InvSteps.MSK)

To build an electrical connection, a physical contact is needed. The corresponding layer is called "contact". You may insert a metal-to-polysilicon contact in the layout using a direct macro situated in the palette.

The *Process Simulator* shows the vertical aspect of the layout, as when fabrication has been completed. This feature is a significant aid to understand the circuit structure and the way layers are stacked on top of each other. A click of the mouse on the left side of the n-channel device layout and the release of the mouse at the right side give the cross-section reported in figure 3-10.

Supply Connections

The next design step consists in adding supply connections, that is the positive supply VDD and the ground supply VSS. We use the metal2 layer (Second level of metallization) to create horizontal supply connections. Enlarging the supply metal lines reduces the resistance and avoids electrical overstress. The simplest way to build the physical connection is to add a metal/Metal2 contact that may be found in the palette. The connection is created by a plug called "via" between metal2 and metal layers.
The final layout design step consists in adding polarization contacts. These contacts convey the VSS and VDD voltage supply close to the bulk regions of the device. Remember that the n-well region should always be polarized to a high voltage to avoid short-circuit between VDD and VSS. Adding the VDD polarization in the n-well region is a very strict rule.

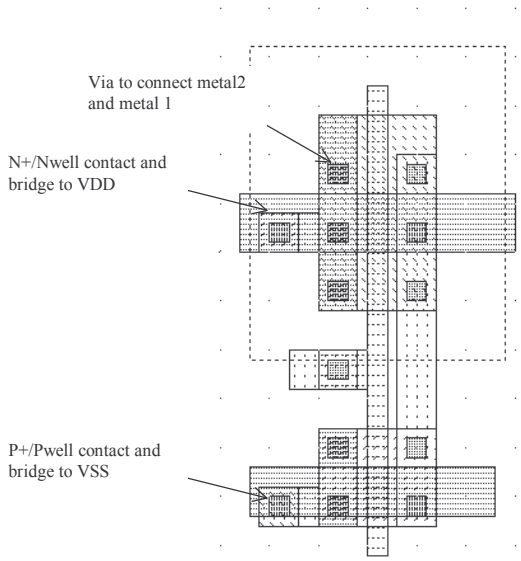


Figure 3-11 : Adding polarization contacts

Process steps to build the Inverter

At that point, it might be interesting to illustrate the steps of fabrication as they would sequence in a foundry. MICROWIND includes a 3D process viewer for that purpose. Click **Simulate → Process steps in 3D**. The simulation of the CMOS fabrication process is performed, step by step by a click on **Next Step**.

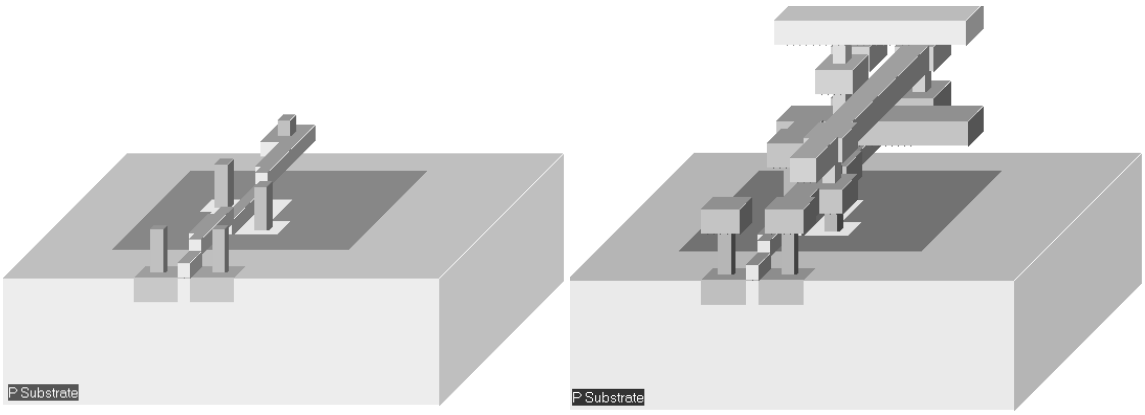


Figure 3-12 : The step-by-step fabrication of the Inverter circuit (InvSteps.MSK)

On figure 3-12, the picture on the left represents the nMOS device, pMOS device, common polysilicon gate and contacts. The picture on the right represents the same portion of layout with the metal layers stacked on top of the active devices.

Inverter Simulation

The inverter simulation is conducted as follows. Firstly, a VDD supply source (1.0 V) is fixed to the upper metal2 supply line, and a VSS supply source (0.0 V) is fixed to the lower metal2 supply line. The properties are located in the palette menu. Simply click the desired property , and click on the desired location in the layout. Add a clock on the inverter input node (The default node name *clock1* has been changed into *Vin*)and a visible property on the output node *Vout*.

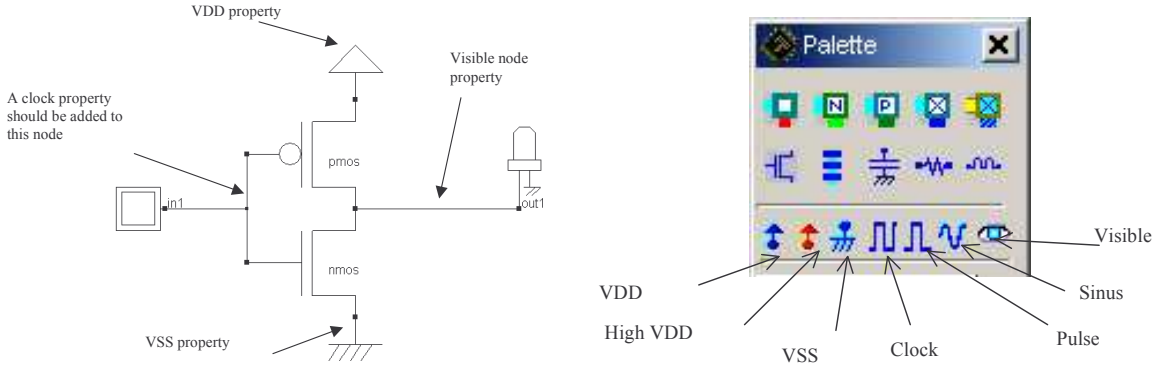


Figure 3-13 : Adding simulation properties (InvSteps.MSK)

The command **Simulate → Run Simulation** gives access to the analog simulation. Select the simulation mode **Voltage vs. Time**. The analog simulation of the circuit is performed. The time domain waveform, proposed by default, details the evolution of the voltages *in1* and *out1* versus time. This mode is also called transient simulation, as shown in figure 3-14.

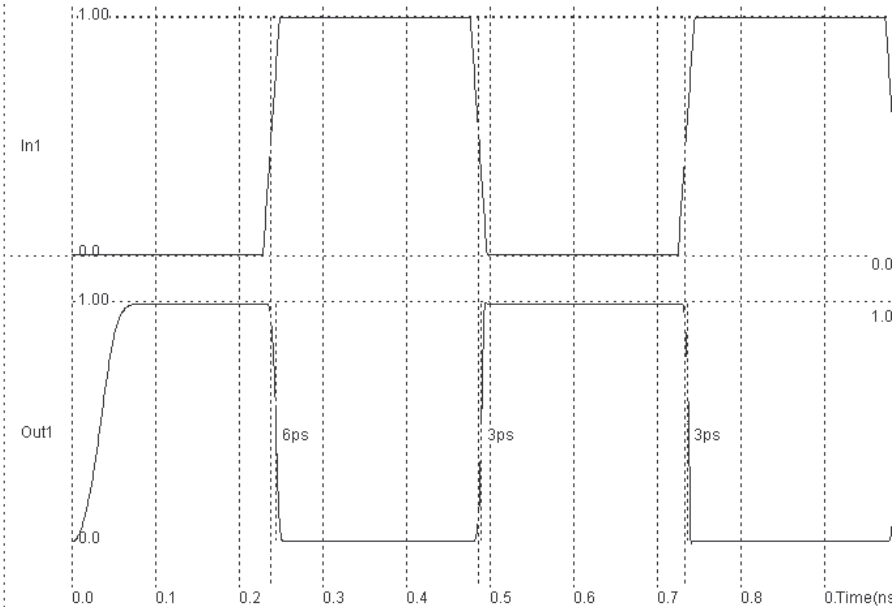


Figure 3-14 : Transient simulation of the CMOS inverter (InvSteps.MSK)

The truth-table is verified as follows. A logic “0” corresponds to 0 V a logic “1” to a 1. 0 V. When the input rises to “1”, the output falls to “0”, with a 3 pico-second delay (6.10^{-12} second).

Added Features in the Full version

Power estimation	Analysis of the inverter consumption, the leakage, etc…
3-state inverter	A complete description of the 3-state circuits, with details on the structure, behavior.
Inverter sizing effects	Impact of the width and length of MOS devices on the inverter characteristics.
Exercises	Some basic exercises related to the inverter design and its static/dynamic performances.

4 Basic Gates

Introduction

Table 4-1 gives the corresponding symbol to each basic gate as it appears in the logic editor window as well as the logic description. In this description, the symbol & refers to the logical AND, | to Or, ~to INVERT, and ^ to XOR. A complete description of basic gate implementation may be found in [Baker].

Name	Logic symbol	Logic equation
INVERTER		Out=~in;
AND		Out=a&b;
NAND		Out=~(a.b);
OR		Out=(a|b);
NOR		Out=~(a|b);
XOR		Out=a^b;
XNOR		Out=~(a^b);

Table 4-1. The list of basic gates

The Nand Gate

The truth-table and logic symbol of the NAND gate with 2 inputs are shown below. In DSCH , select the NAND symbol in the palette, add two buttons and one lamp as shown above. Add interconnects if necessary to link the button and lamps to the cell pins. Verify the logic behavior of the cell.

in1	in2	Out
0	0	1
0	1	1
1	0	1
1	1	0

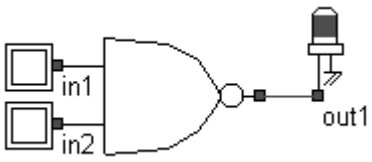


Figure 4-1 : The truth table and symbol of the NAND gate

In CMOS design, the NAND gate consists of two nMOS in series connected to two pMOS in parallel. The schematic diagram of the NAND cell is reported below. The nMOS in series tie the output to the ground for one single combination A=1, B=1.

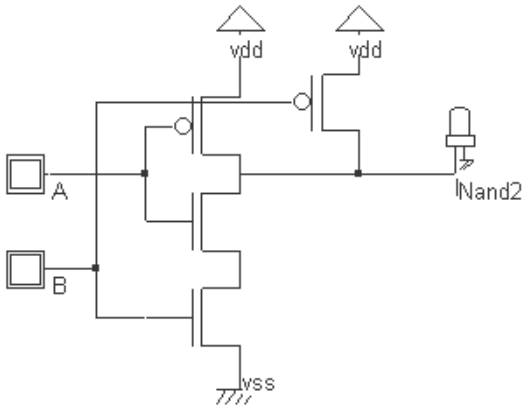
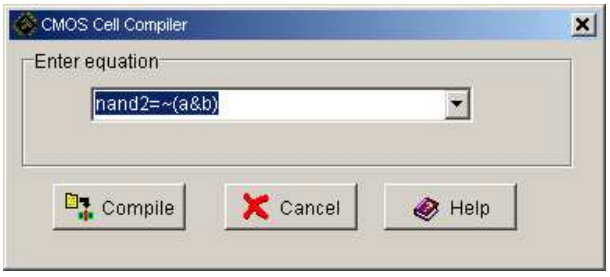


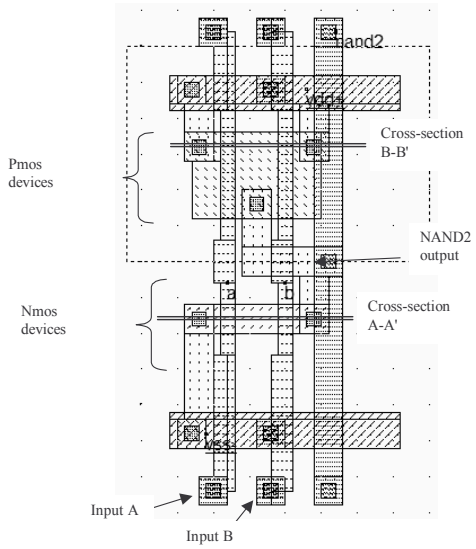
Figure 4-2 : The truth table and schematic diagram of the CMOS NAND gate design (NandCmos.SCH)

For the three other combinations, the nMOS path is cut, but a least one pMOS ties the output to the supply VDD. Notice that both nMOS and pMOS devices are used in their best regime: the nMOS devices pass "0", the pMOS pass "1".

You may load the NAND gate design using the command **File → Read→NAND.MSK**. You may also draw the NAND gate manually as for the inverter gate. An alternative solution is to compile directly the NAND gate into layout with MICROWIND . In this case, complete the following procedure:



In MICROWIND , click on **Compile→Compile One Line.** Select the line corresponding to the 2-input NAND description as shown above. The input and output names can be by the user modified.



Click **Compile**. The result is reported above.

The compiler has fixed the position of VDD power supply and the ground VSS. The texts *A, B,* and *S* have also been fixed to the layout. Default clocks are assigned to inputs *A* and *B*.

Figure 4-3 : A NAND cell created by the CMOS compiler.

The cell architecture has been optimized for easy supply and input/output routing. The supply bars have the property to connect naturally to the neighboring cells, so that specific effort for supply routing is not required. The input/output nodes are routed on the top and the bottom of the active parts, with a regular spacing to ease automatic channel routing between cells.

The AND gate

As can be seen in the schematic diagram and in the compiled results, the AND gate is the sum of a NAND2 gate and an inverter. The layout ready to simulate can be found in the file `AND2.MSK`. In CMOS, the negative gates (NAND, NOR, INV) are faster and simpler than the non-negative gates (AND, OR, Buffer). The cell delay observed in the figure 4-4 are significantly higher than for the NAND2 gate alone, due to the inverter stage delay.

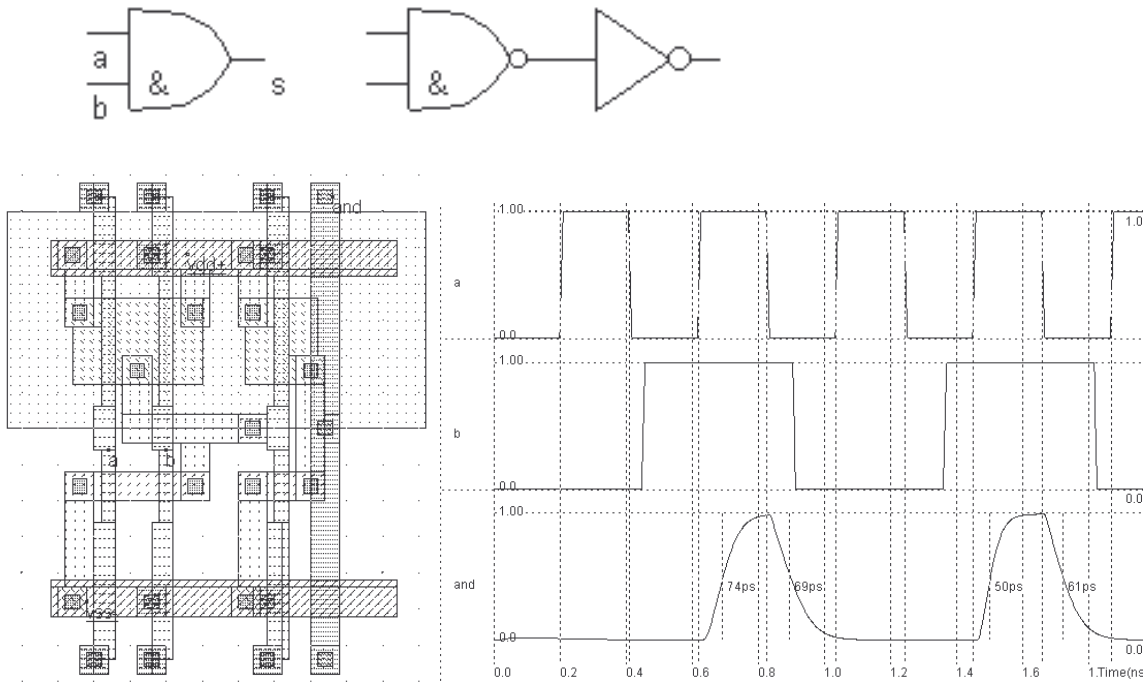


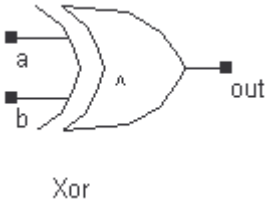
Figure 4-4 : Layout and simulation of the AND gate

The XOR Gate

The truth-table and the schematic diagram of the CMOS XOR gate are shown above. There exist many possibilities for implementing the XOR function into CMOS. The least efficient design, but the most forward, consists in building the XOR logic circuit from its Boolean equation.

XOR 2 inputs

A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	0



The proposed solution consists of a transmission-gate implementation of the XOR operator. The truth table of the XOR can be read as follow: *IF B=0, OUT=A, IF B=1, OUT = Inv(A).* The principle of the circuit presented below is to enable the A signal to flow to node *N1* if *B*=1 and to enable the *Inv(A)* signal to flow to node *N1* if *B=0*. The node *OUT* inverts *N1*, so that we can find the XOR operator. Notice that the nMOS and pMOS devices situated in the middle of the gate serve as pass transistors.

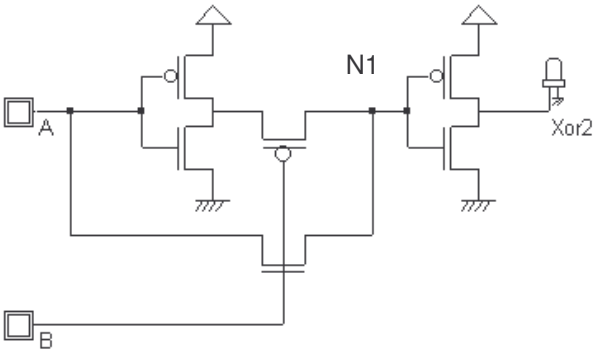


Figure 4-5 : The schematic diagram of the XOR gate (XORCmos.SCH)

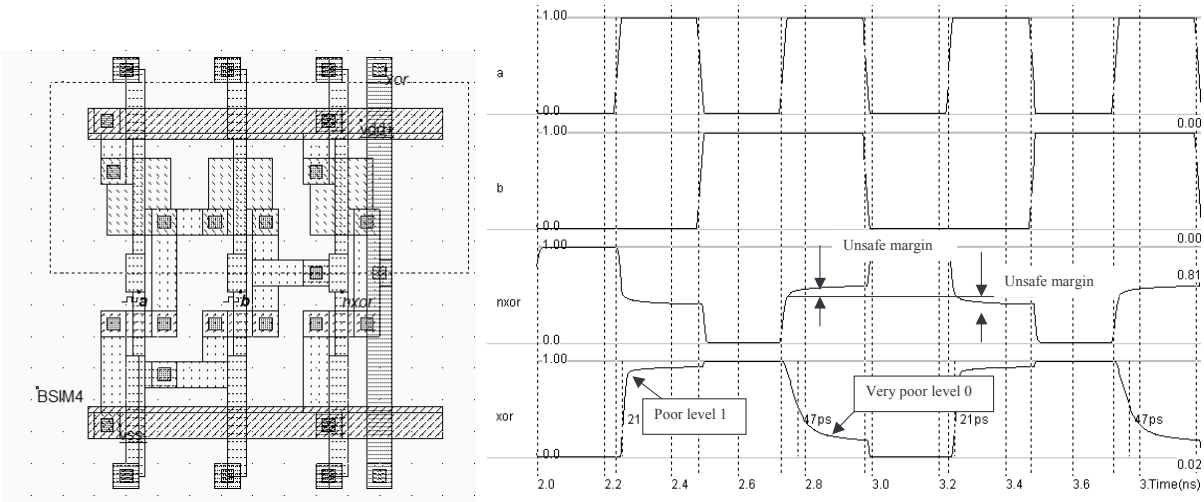


Figure 4-6 : Layout and simulation of the XOR gate (XOR.MSK).

You may use DSCH to create the cell, generate the Verilog description and compile the resulting text. In MICROWIND , the Verilog compiler is able to construct the XOR cell as reported in Figure 4-6. You may add a visible property to the intermediate node which serves as an input of the second inverter.

See how the signal, called *internal*, is altered by *Vtn* (when the nMOS is ON) and *Vtp* (when the pMOS is ON). Fortunately, the inverter regenerates the signal.

Multiplexor

Multiplexing means transmitting a large amount of information through a smaller number of connections. A digital multiplexor is a circuit that selects binary information from one of many input logic signals and directs it to a single input line. The main component of the multiplexor is a basic cell called the transmission gate. The transmission gate let a signal flow if *Enable* is asserted.

Sel	In0	In1	f
0	x	0	0
0	x	1	1
1	0	x	0
1	1	x	1

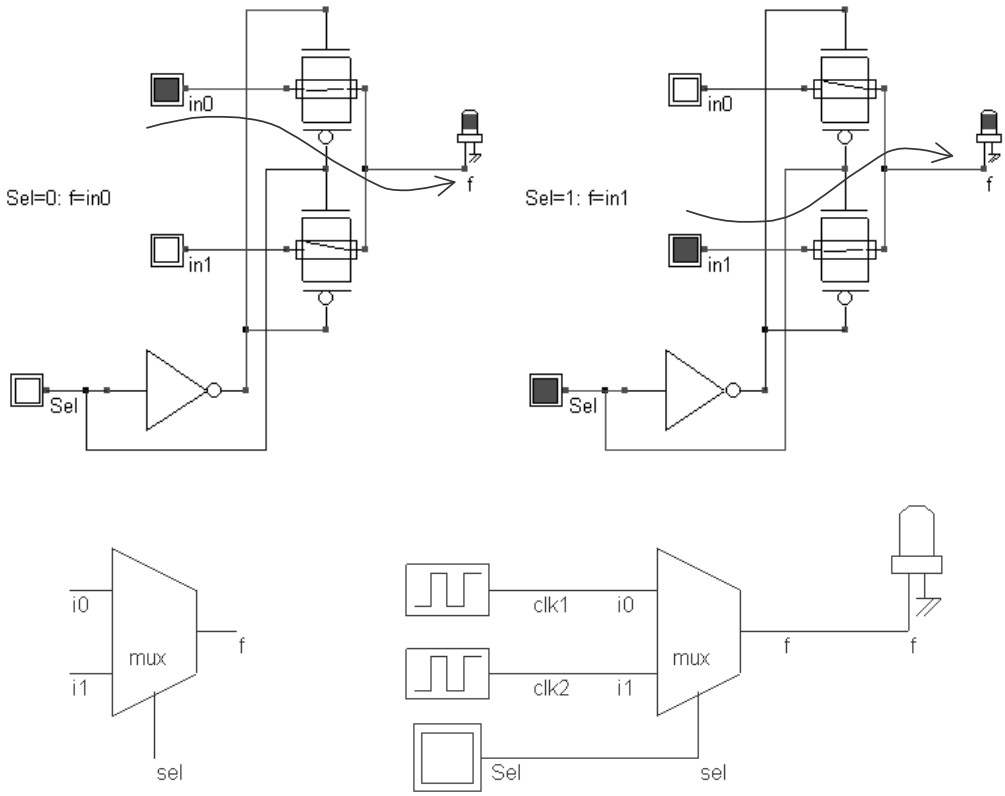


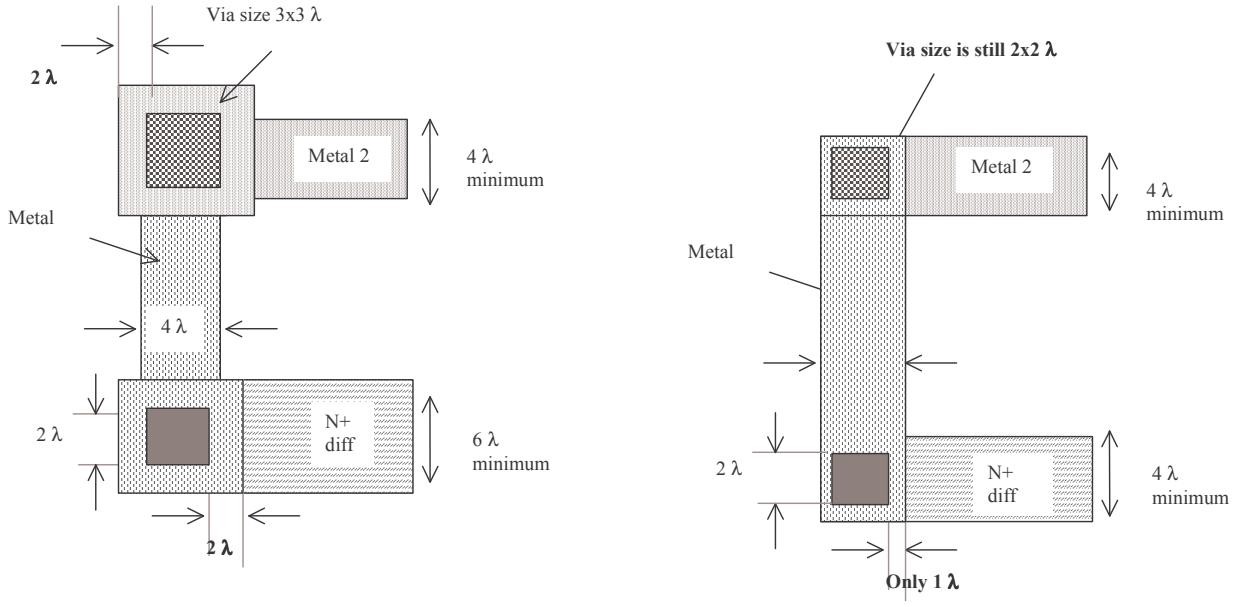
Figure 4-7 : The transmission gate used as a multiplexor (MUX.SCH)

In DSCH , a transmission gate symbol exists (Figure 4-7). It includes the nMOS, pMOS and inverter cells. Concerning the layout, the channel length is usually the minimum length available in the technology, and the width is set large, in order to reduce the parasitic 'on' resistance of the gate.

Interconnects

Up to 8 metal layers are available in CMOS 65-nm technology for signal connection and supply purpose. A significant gap exists between the 0.7 µm 2-metal layer technology and the 65-nm technology in terms of interconnect efficiency. Firstly, the contact size is 6 lambda in 0.7µm technology, and only 4 lambda in 65-nm.

This features a significant reduction of device connection to metal and metal2, as shown in figure 4-8. Notice that a MOS device generated using 0.7µm design rules is still compatible with 0.12µm technology. But a MOS device generated using 65-nm design rules would violate several rules if checked in 0.7µm technology.



(a) Contacts in 0.7µm technology

(b) Contacts in 65-nm technology

Figure 4-8 : Contacts in 0.7µm technology require more area than in 0.12µm technology

Secondly, the stacking of contacts is not allowed in micro technologies. This means that a contact from poly to metal2 requires a significant silicon area as contacts must be drawn in a separate location. In deep-submicron technology (Starting 0.35µm and below), stacked contacts are allowed.

Added Features in the Full version

Basic Gates	Truth-table and schematic diagram of the three-input OR gate. AND 4 inputs. Generalization.
Complex Gates	The technique produces compact cells with higher performances in terms of spacing and speed than conventional logic circuits. The concept of complex gates is illustrated through concrete examples. The logic implementation of complex gates in DSCH is also described.
Multiplexor	Description of a 2^n input lines and n selection lines whose bit combinations determine which input is selected. Transmission gate implementation of the 8 to 1 multiplexor.
Interconnect layers and RC behavior	Description of the interconnect materials: metal1..metal6, supply metals, via, RC effects in interconnects, as well as basic formulations for the resistance, inductance and capacitance. Illustration of the crosstalk effect in interconnects.
Exercises	XOR, complex gates, design considerations.

5 Arithmetics

This chapter introduces basic concepts concerning the design of arithmetic gates. The adder circuit is presented, with its corresponding layout created manually and automatically. Then the comparator, multiplier and the arithmetic and logic unit are also discussed. This chapter also includes details on a student project concerning the design of binary-to-decimal addition and display.

Unsigned Integer format

The two classes of data formats are the integer and real numbers. The integer type is separated into two formats: unsigned format and signed format. The real numbers are also sub-divided into fixed point and floating point descriptions. Each data is coded in 8,16 or 32 bits. We consider here unsigned integers, as described in figure 5-1.

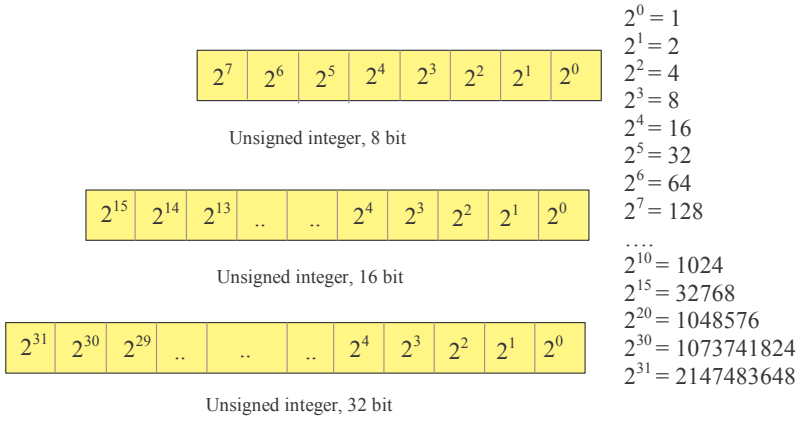


Figure 5-1 : Unsigned integer format

Half-Adder Gate

The Half-Adder gate truth-table and schematic diagram are shown in Figure 5-2. The SUM function is made with an XOR gate, the Carry function is a simple AND gate.

HALF ADDER

A	B	SUM	CARRY
0 0	0	0	
0 1	1	0	
1 0	1	0	
1 1	0	1	

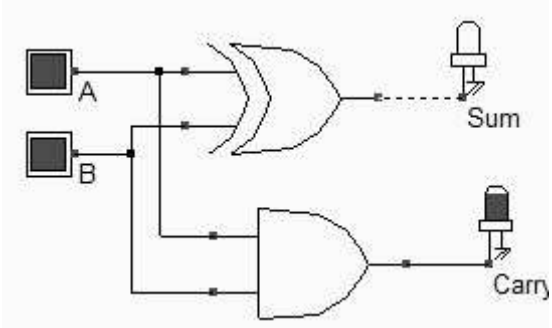


Figure 5-2 : Truth table and schematic diagram of the half-adder gate (HADD.MSK).

FULL CUSTOM LAYOUT	You may create the layout of the half-adder fully by hand in order to create a compact design. Use the polysilicon and metal1 layers for short connections only, because of the high resistance of these materials. Use Poly/Metal, Diff/Metal contact macros situated in the upper part of the Palette menu to link the layers together.
LAYOUT LIBRARY	Load the layout design of the Half-Adder using **File → Open** and loading the file `HalfAdder.MSK`.

VERILOG COMPILING. Use DSCH to create the schematic diagram of the half-adder. Verify the circuit with buttons and lamps. Save the design under the name `HalfAdder.sch` using the command **File → Save As.** Generate the Verilog text by using the command **File → Make Verilog File**. The text file `HalfAdder.v` is created. In MICROWIND , click on the command **Compile → Compile Verilog File**. Select the text file `HalfAdder.v`. Click **Compile.** When the compiling is complete, the resulting layout appears shown below. The XOR gate is routed on the left and the AND gate is routed on the right. Now, click on **Simulate →Start Simulation**. The timing diagrams of figure 5-3 appear and you should verify the truth table of the half-adder.

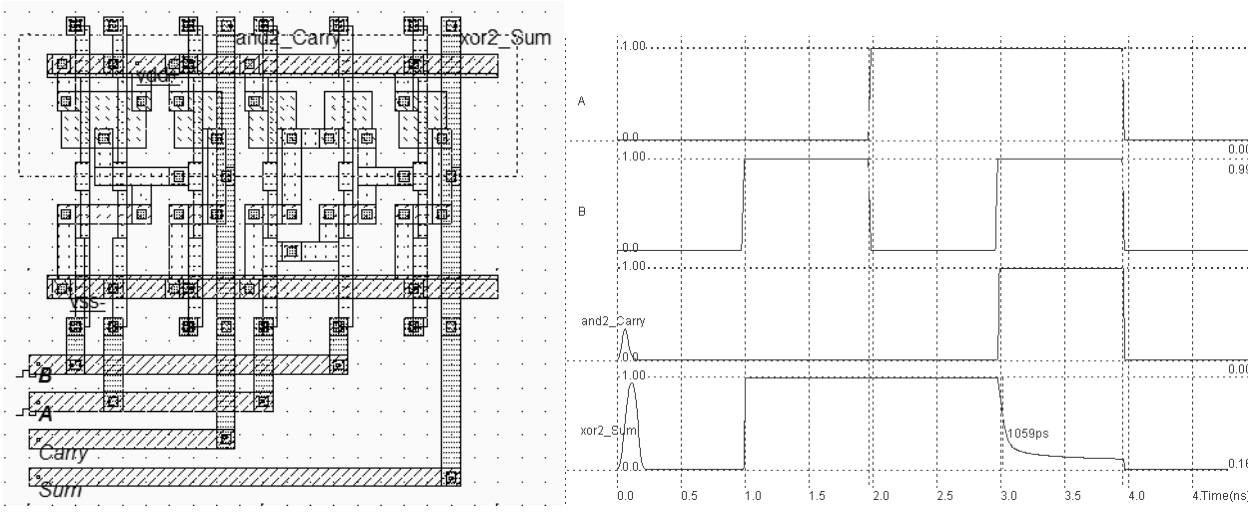


Figure 5-3 : Compiling and simulation of the half-adder gate (HalfAdder.MSK)

Full-Adder Gate

The truth table and schematic diagram for the full-adder are shown in Figure 5-4. The SUM is made with two XOR gates and the CARRY is a combination of NAND gates, as shown below. The most straightforward implementation of the CARRY cell is AB+BC+AC. The weakness of such a circuit is the use of positive logic gates, leading to multiple stages. A more efficient circuit consists in the same function but with inverting gates.

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

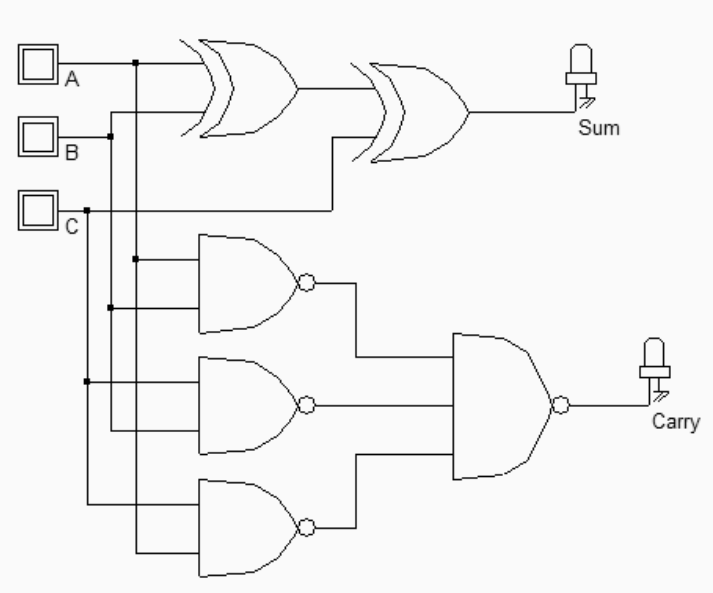


Figure 5-4 : The truth table and schematic diagram of a full-adder(FADD.SCH)

Full-Adder Symbol in DSCH

When invoking **File → Schema to new symbol**, the screen of figure 5-5 appears. Simply click **OK**. The symbol of the full-adder is created, with the name *fadd.sym* in the current directory. A Verilog description of the circuit is attached to the symbol.

We see that the XOR gates are declared as primitives while the complex gate is declared using the **Assign** command, as a combination of AND (&)and OR (|) operators. If we used AND and OR primitives instead, the layout compiler would implement the function in a series of AND and OR CMOS gates, loosing the benefits of complex gate approach in terms of cell density and switching speed.

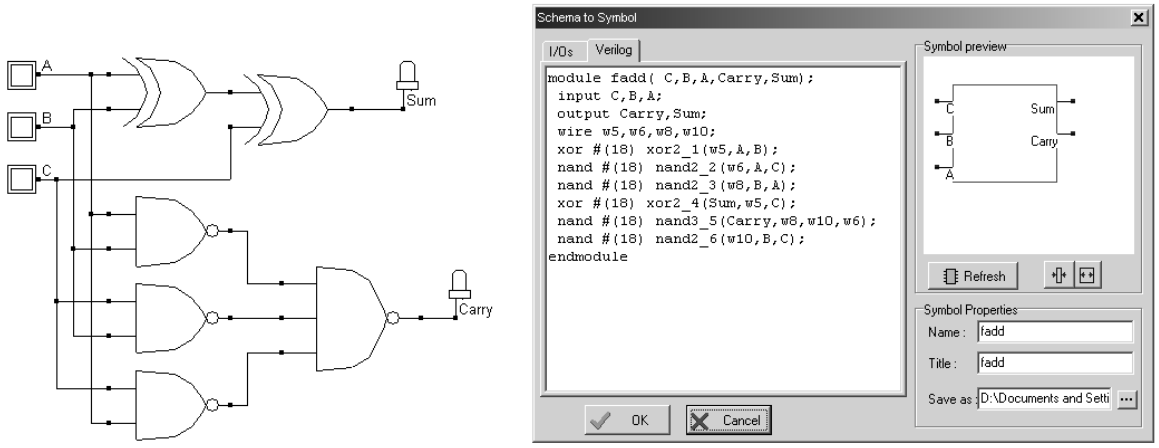


Figure 5-5 : Verilog description of the full adder (fadd.SYM)

Use the command **Insert → User Symbol** to include the full-adder symbol into a new circuit. For example, a 4-bit adder is proposed in figure 5-6. The two displays are connected to the identical data, but are configured in different mode: hexadecimal format for the right-most display, and integer mode for the left-most display.

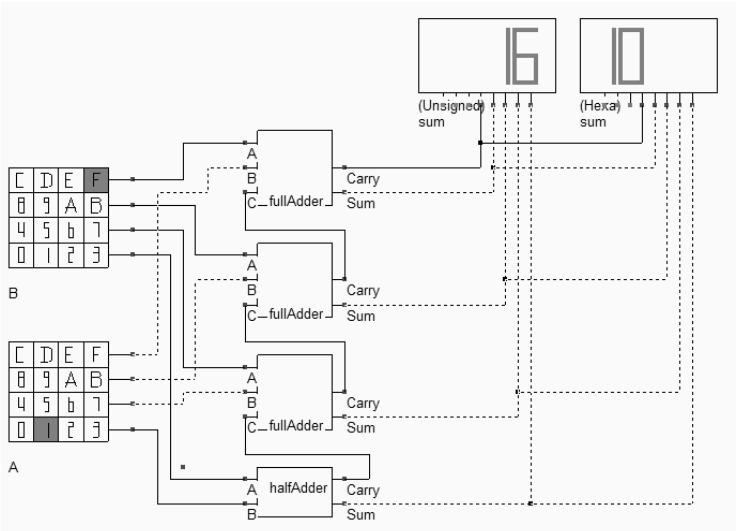


Figure 5-6 : Schematic diagram of the four-bit adder and some examples of results (Add4.SCH).

Comparator

The truth table and the schematic diagram of the comparator are given below. The A=B equality represents an XNOR gate, and A>B, A<B are operators obtained by using inverters and AND gates.

Comparator

A	B	A>B	A<B	A=B
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

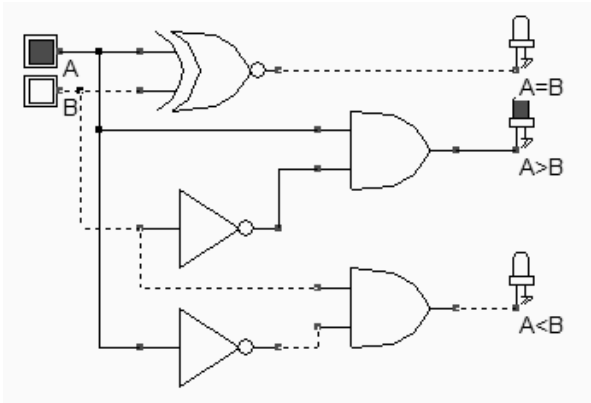


Figure 5-7 : The truth table and schematic diagram of the comparator (COMP.SCH).

Using DSCH , the logic circuit of the comparator is designed and verified at logic level. Then the conversion into Verilog is invoked (**File → Make verilog File**). MICROWIND compiles the verilog text into layout. The simulation of the comparator is given in Figure 5-8. The XNOR gate is located at the left side of the design. The inverter and NOR gates are at the right side. After the initialization, *A=B* rises to 1. The clocks *A* and *B* produce the combinations 00,01,10 and 11.

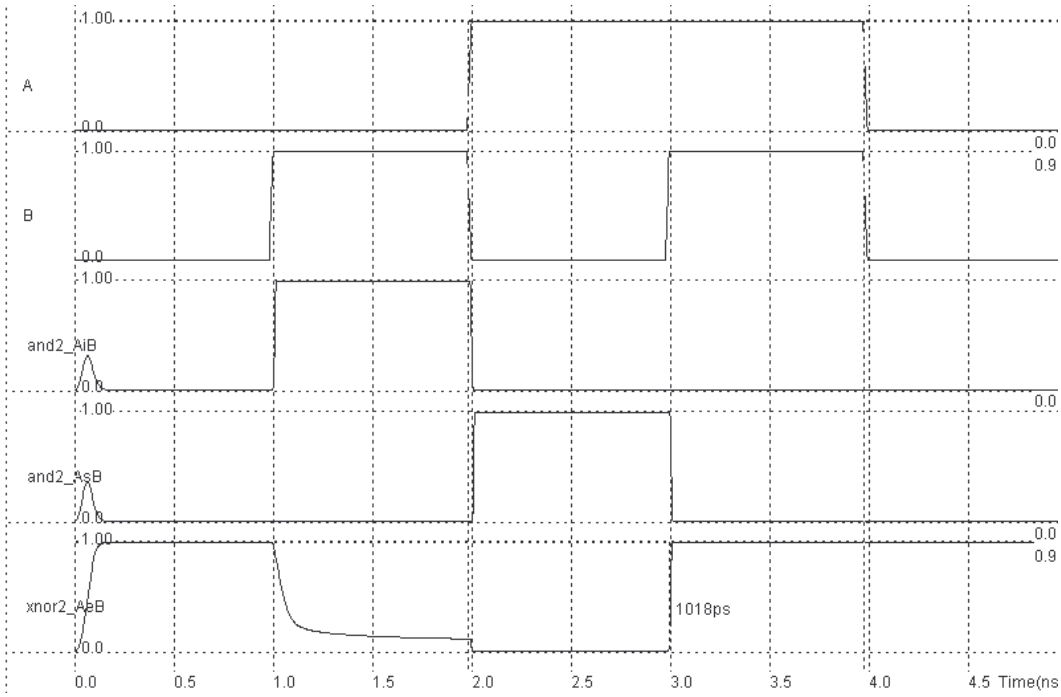


Figure 5-8 : Simulation of a comparator (COMP.MSK).

Micro-controller Models

In DSCH , a simplified model of the Intel 8051 and PIC 16f84 micro-controllers are included. The 8051 core includes an arithmetic and logic unit to support a huge set of instructions. You can add the corresponding symbol (8051.SYM) using the command **Insert → User Symbol**, as the symbol is not directly accessible through the symbol palette.

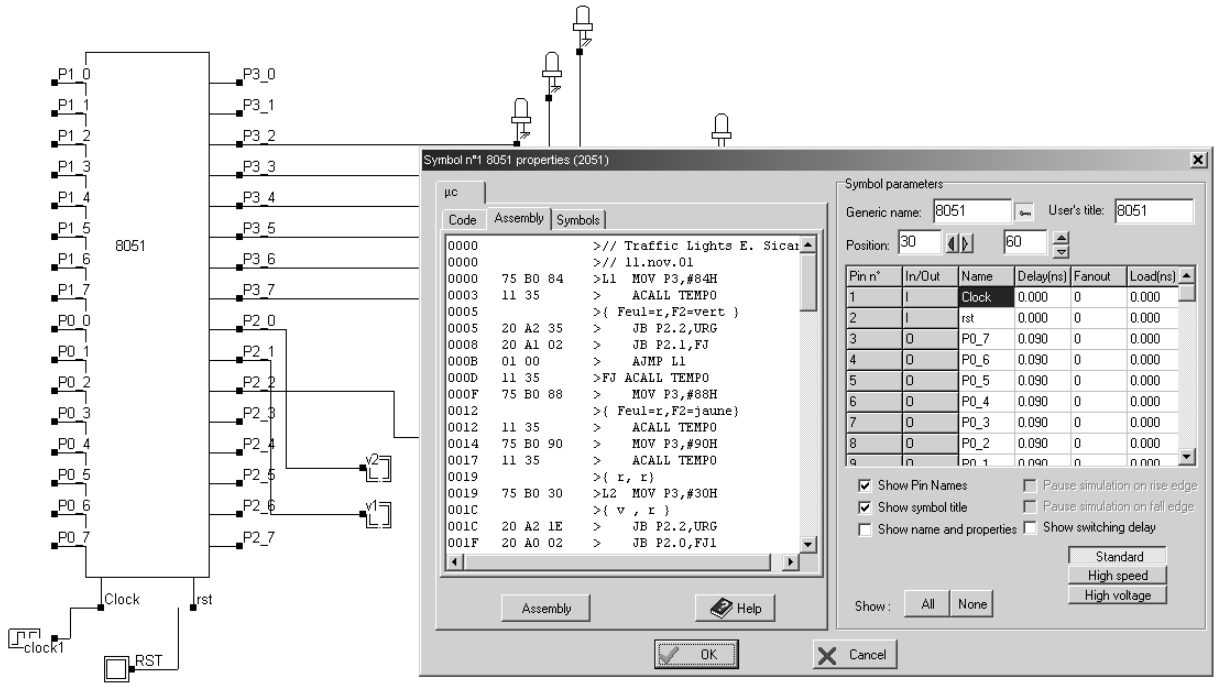


Figure 5-9 : The 8051 symbol and its embedded software (8051_traffic_lights.sch)

The symbol consists mainly of general purpose input/output ports (*P0,P1,P2* and *P3*), a *clock* and a *reset* control signals. The basic connection consists of a clock on the *Clock* input and a button on the *Reset* input. After a double-click in the symbol, the embedded code appears. That code may be edited and modified (Figure 5-9). When the button **Assembly** is pressed, the assembly text is translated into executable binary format. Once the logic simulation is running, the code is executed as soon as the reset input is deactivated. The value of the program counter, the accumulator A, the current *op_code* and the registers is displayed. In the chronograms, the accumulator variations versus the time are displayed. It can be noticed that this core operates with one single clock cycle per instruction, except for some instructions such as MOV (Move data) and AJMP (Jump to a specific address).

Added Features in the Full version

Adders	Full layout of the 4-bit adder. Structure of the carry look-ahead adder. Details on the routing and supply strategy.
Micro-controller	Model of the PIC16f84 micro-controller. Example files are provided, illustrating code execution.
Arithmetic and Logic Units	Basic principles of micro-operations on 8 bit format.

6 Latches

This chapter details the structure and behavior of latch circuits. The RS Latch, the D Latch, the edge-sensitive register and the counter are presented.

Basic Latch

The basis for storing an elementary binary value is called a latch. The simplest CMOS circuit is made from 2 inverters.

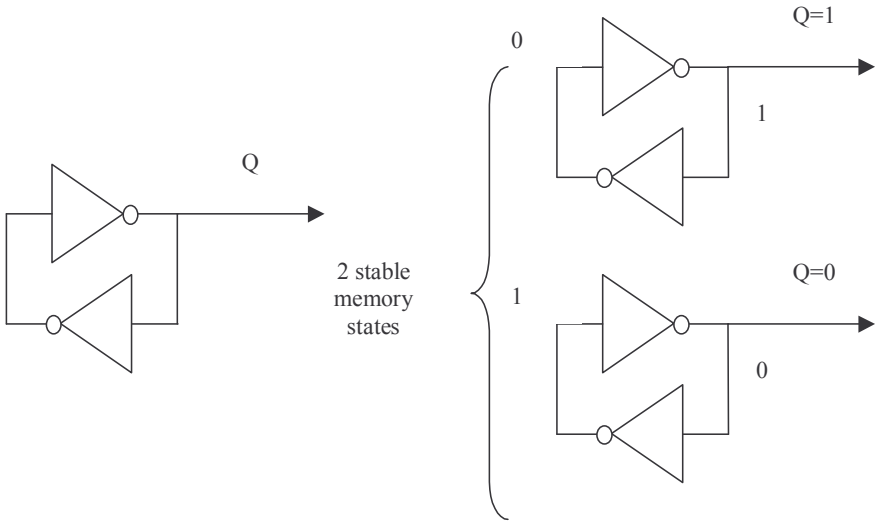


Figure 6-1 : Elementary memory cell based on an inverter loop

RS Latch

The RS Latch, also called Set-Reset Flip Flop (SR FF), transforms a pulse into a continuous state. The RS latch can be made up of two interconnected NOR or NAND gates, inspired from the two chained inverters of figure 6-2. In the case of RS-NOR, the *Reset* and *Set* inputs are active high. The memory state corresponds to *Reset=Set=0*. The combination *Reset=Set=1* should not be used, as it means that *Q* should be *Reset* and *Set* at the same time.

RS Latch (NOR)

R	S	Q	nQ
0	0	Q	nQ
0	1	1	0
1	0	0	1
1	1	1	1

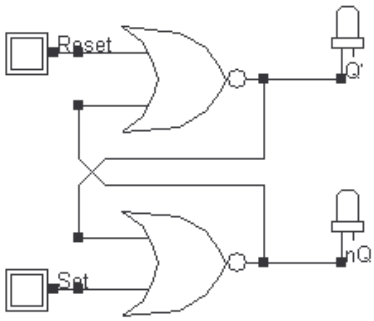


Figure 6-2 : The truth table and schematic diagram of a RS latch made (RSNor.SCH)

FULL CUSTOM LAYOUT. You may create the layout of RS latch manually. The two NOR gates may share the VDD and VSS supply achieving continuous diffusions.
LAYOUT COMPILING. Use DSCH to create the schematic diagram of the RS latch. Verify the circuit with buttons and lamps. Save the design under the name `RS.sch` using the command **File → Save As.** Generate the Verilog text file (.v appendix) by using the command **File → Make Verilog File**. In MICROWIND , click on the command **Compile → Compile Verilog File**. Select the text file `RS.v`. Click on **Compile.** When the compiling is complete, the resulting layout appears as shown below. The NOR implementation of the RS gate is completed.

```
module RSNor( Reset,Set,Q,nQ);
 input Reset,Set;
 output Q,nQ;
 nor nor1(Q,nQ,Reset);
 nor nor2(nQ,Set,Q);
endmodule
```

With the *Reset* and *Set* signals behaving like clocks, the memory effect is not easy to illustrate. A much better approach consists in declaring pulse signals with an active pulse on *Reset* followed by an active pulse on *Set*. Consequently, you must change the clock property into a pulse property. For NOR implementation, the pulse is positive.

1. Select the **Pulse** icon. Click on the node *Reset*.
2. Click the brush to clear the existing pulse properties of the pulse.
3. Enter the desired start time (0.48 ns in this example) and pulse duration, and click **Insert** (see figure 6-3).

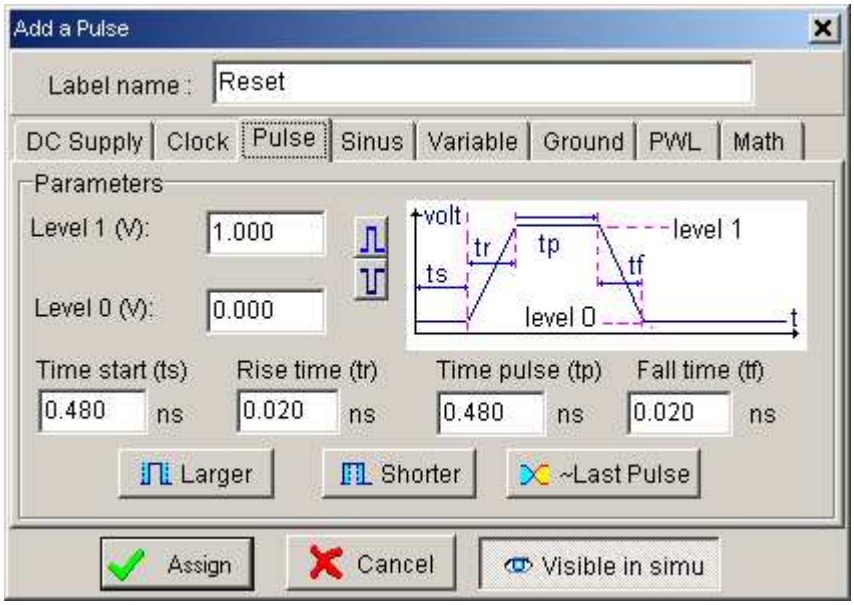


Figure 6-3 : The pulse property used to control the Reset of the latch (RsNor.MSK)

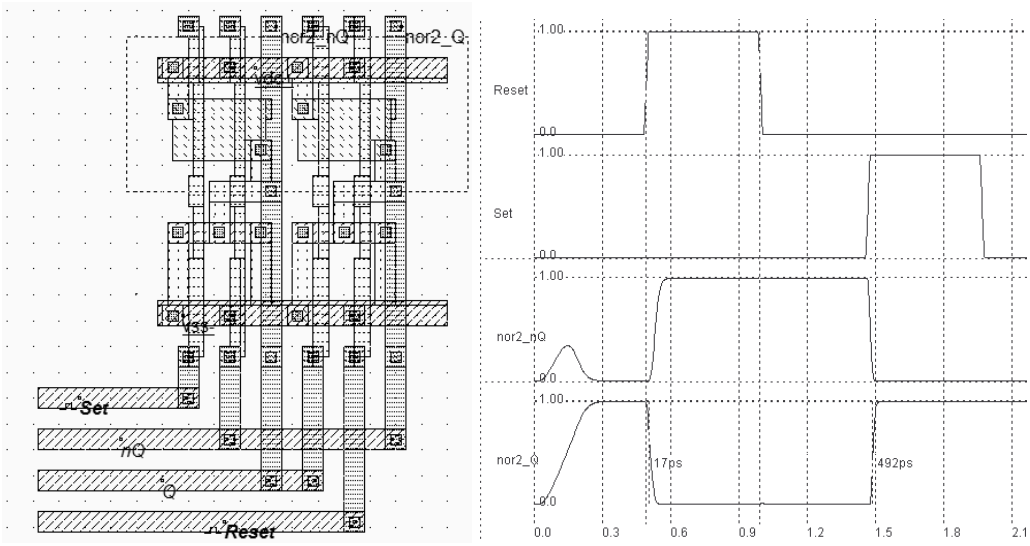


Figure 6-4 : Layout of the RS latch made (RSNor.MSK)

4. Repeat the same procedure to change the clock into a pulse for node *Set*. The start time is now fixed to 1.48 ns to generate a pulse later than for the *Reset* signal.
5. Click on **Simulate →Start Simulation**. The timing diagrams of figure 6-4 appear.

In the simulation of figure 6-4, a positive pulse on *Set* turns *Q* to a stable high state. Notice that when *Set* goes to 0, *Q* remains at 1, which is called the 'memory' state. When a positive pulse occurs on *Reset*, *Q* goes low, *nQ* goes high. In this type of simulation, the combination *Reset=Set=1* is not present.

Edge Trigged Latch

This edge-trigged latch is one of the most widely used cells in microelectronics circuit design. The cell structure comprises two master-slave basic memory stages. The most compact implementation of the edge-trigged latch is reported below (figure 6-5). The schematic diagram is based on inverters and pass-transistors. On the left side, the two chained inverter are in memory state when the pMOS loop transistor *P1* is on, that is when *Clk=0*. The two-chained inverters on the right side act in an opposite way. The reset function is obtained by a direct ground connection of the master and slave memories, using nMOS devices.

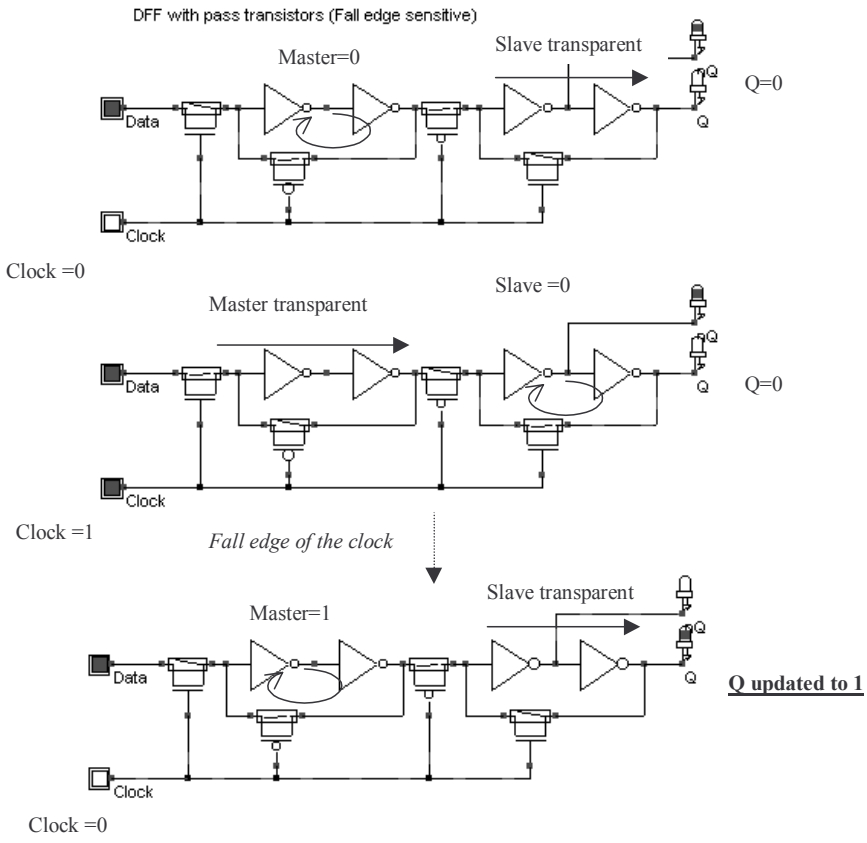


Figure 6-5 : The edge-trigged latch and its logic simulation (Dreg.MSK)

When *clock* is high, the master latch is updated to a new value of the input *D*. The slave latch produces to the output *Q* the previous value of *D*. When *clock* goes down, the master latch turns to memory state. The slave circuit is updated. The change of the clock from 1 to 0 is the active edge of the clock. This type of latch is a negative edge flip flop.

Use the Verilog compiler to generate the edge-trigged latch description in Verilog format, or by creating a schematic diagram including the "D" register symbol, in the symbol palette of DSCH . As can be seen, the register is built up from one single call to the primitive `dreg`. For simulation:

- *Reset* is active on a level 1. *Reset* is activated twice, at the beginning and later, using a piece-wise linear description included in the pulse property.
- *Clk* is a clock with 10ns at 0 and 10ns at 1.
- *D* is the data chosen here not synchronized with *Clk*, in order to observe various behaviors of the register.

To compile the DREG file, use the command **Compile→Compile Verilog Text**. The corresponding layout is reported below. The piece-wise-linear data is transferred to the text label "Reset" appearing in the lower corner of the D flip flop layout of figure 6-6.

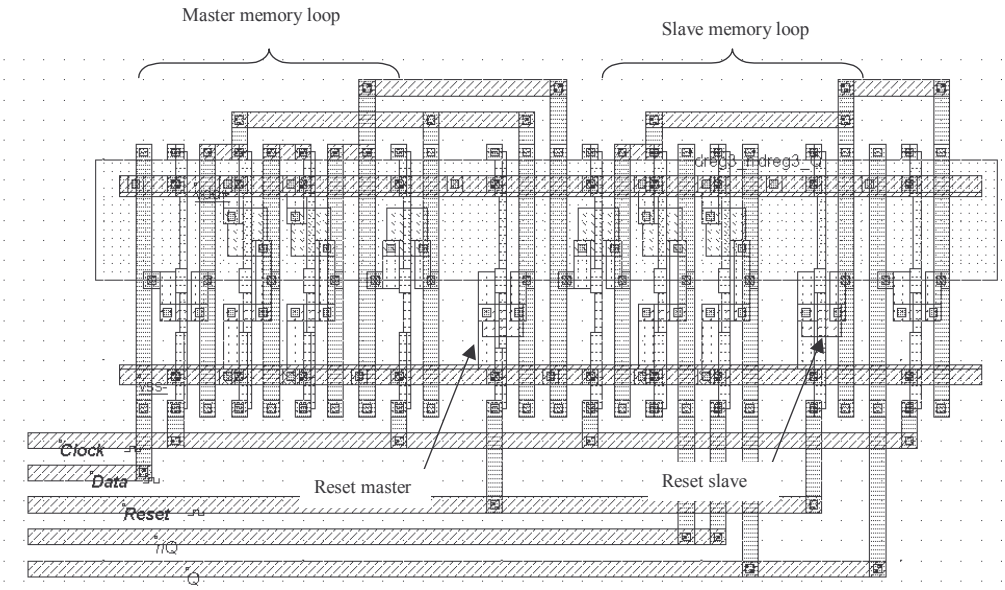


Figure 6-6 : Compiled version of the Edge-trigged D Flip Flop (DregCompile.MSK)

For testing the Dreg, the *Reset* signal is activated twice, at the beginning and later, using a piece-wise linear property (figure 6-6). The *Clock* signal has a 2 ns period. *D* is the data chosen here not synchronized with *Clock*, in order to observe various behaviors of the register.
The simulation of the edge-trigged D-register is reported in figure 6-6. The signals *Q* and *nQ* always act in opposite. When *Reset* is asserted, the output *Q* is 0, *nQ* is 1. When *Reset* is not active, *Q* takes the value of *D* at a fall edge of the clock. For all other cases, *Q* and *nQ* remain in memory state. The latch is thus sensitive to the fall edge of the clock.

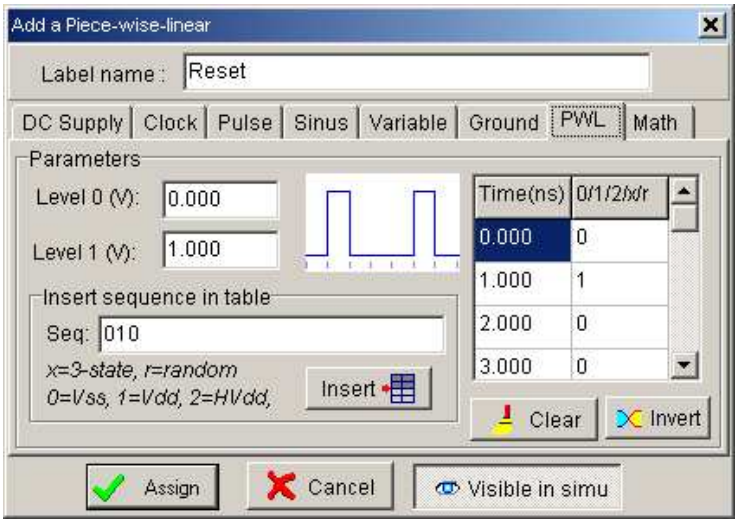


Figure 6-7 : Piece-wise-linear property used for sophisticated control of input signals (DregCompile.MSK)

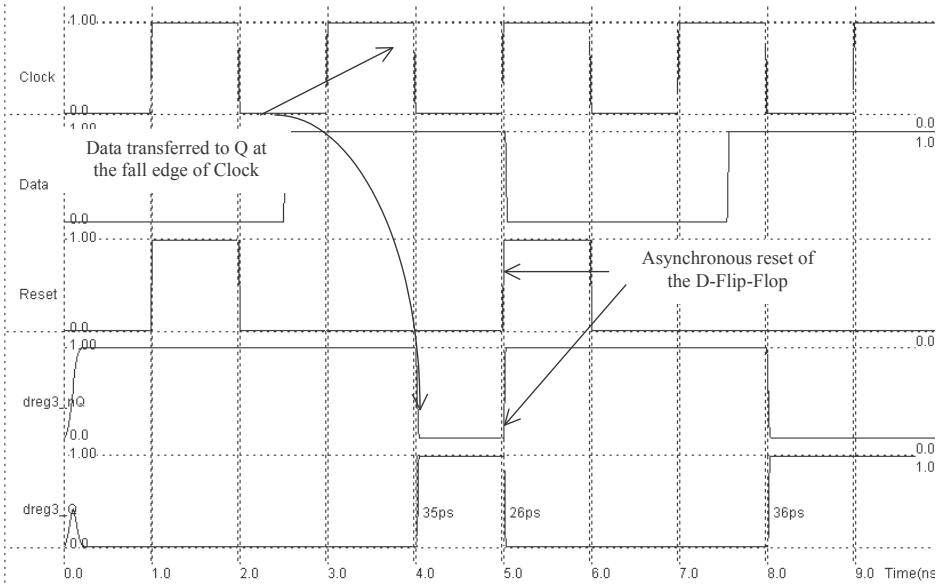


Figure 6-8 : Simulation of the DREG cell (DregCompile.MSK)

Added Features in the Full version

Latches	The truth table and schematic diagram of the static D latch, also called Static D-Flip-Flop are described. The main characteristics of the latch switching are presented.
Counters	The one-bit counter is able to produce a signal featuring half the frequency of a clock. The implementation is detailed. Up and down counters are also described.
Registers	Shift registers, serial registers are described.

7 Memory Circuits

Basic Memory Organization

Figure 7-1 shows a typical memory organization layout [Sharma]. It consists of a memory array, a row decoder, a column decoder and a read/write circuit. The row decoder selects one row from 2^N, thanks to a N-bit row selection address. The column decoder selects one row from 2^M, thanks to a M-bit column selection address. The memory array is based on 2^N rows and 2^M columns of a repeated pattern, the basic memory cell. A typical value for N and M is 10, leading to 1024 rows and 1024 columns, which corresponds to 1048576 elementary memory cells (1Mega-bit).

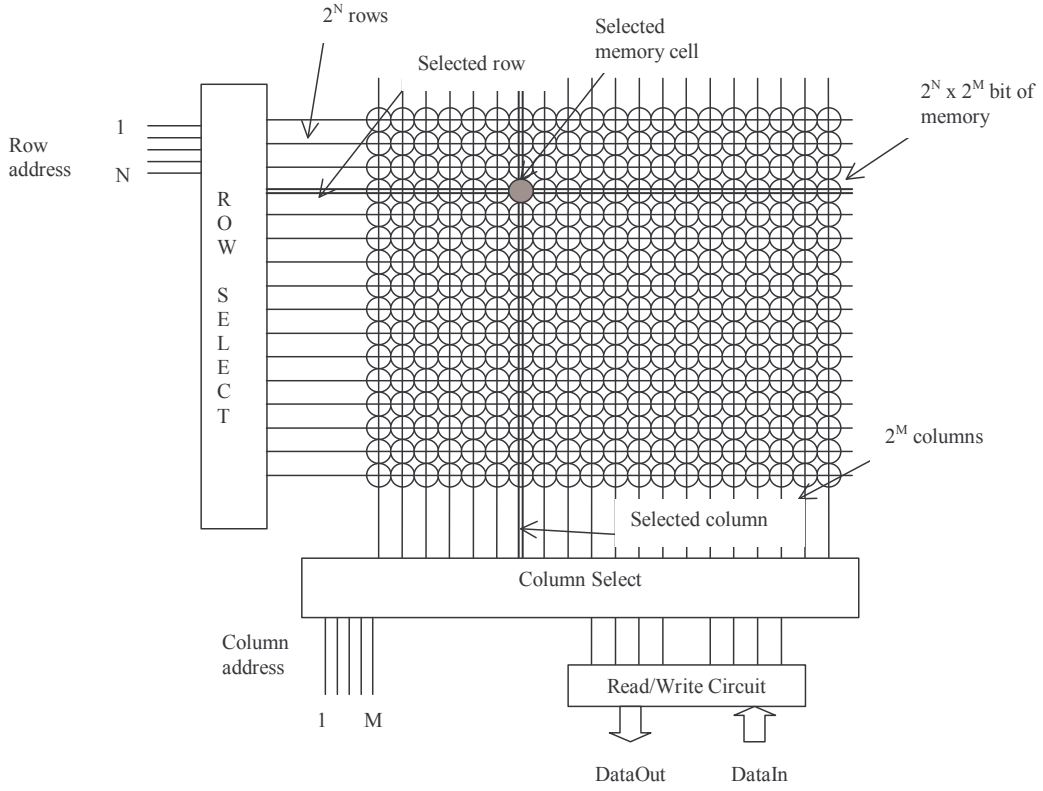


Figure 7-1 : Typical memory organization

RAM Memory

The basic cell for static memory design is based on 6 transistors, with two pass gates instead of one. The corresponding schematic diagram is given in Figure 7-2. The circuit consists of the 2 cross-coupled inverters, but uses two pass transistors instead of one.

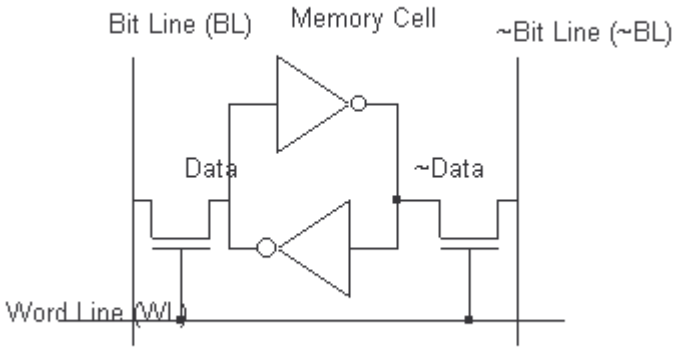


Figure 7-2 : The layout of the 6 transistor static memory cell (RAM6T.SCH)

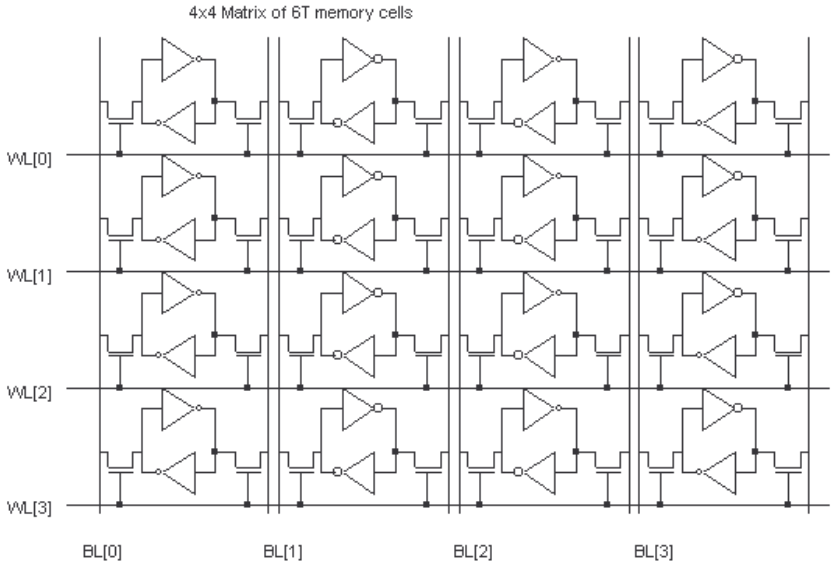


Figure 7-3 : An array of 6T memory cells, with 4 rows and 4 columns (RAM6T.SCH)

The cell has been designed to be duplicated in X and Y in order to create a large array of cells. Usual sizes for Megabit SRAM memories are 256 column x 256 rows or higher. A modest arrangement of 4x4 RAM cells is proposed in figure 7-3. The selection lines *WL* concern all the cells of one row. The bit lines *BL* and *~BL* concern all the cells of one column.

The RAM layout is given in Figure 7-4. The *BL* and *~BL* signals are made with metal2 and cross the cell from top to bottom. The supply lines are horizontal, made with metal3. This allows easy matrix-style duplication of the RAM cell.

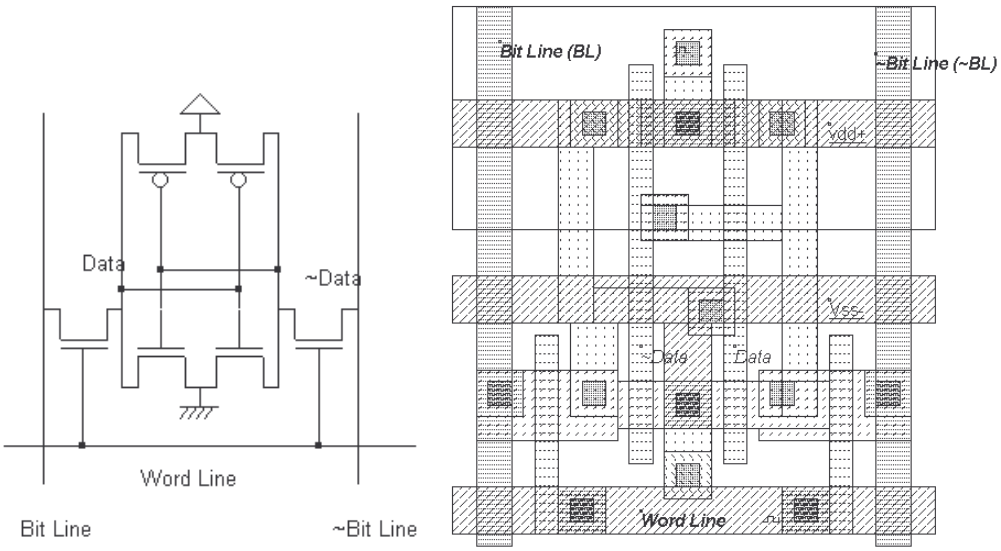


Figure 7-4 : The layout of the static RAM cell (RAM6T.MSK).

WRITE CYCLE. Values 1 or 0 must be placed on *Bit Line*, and the data inverted value on *~Bit Line*. Then the selection *Word Line* goes to 1. The two-inverter latch takes the *Bit Line* value. When the selection *Word Line* returns to 0, the RAM is in a memory state.

READ CYCLE. The selection signal *Word Line* must be asserted, but no information should be imposed on the bit lines. In that case, the stored data value propagates to *Bit Line*, and its inverted value *~Data* propagates to *~Bit Line*.

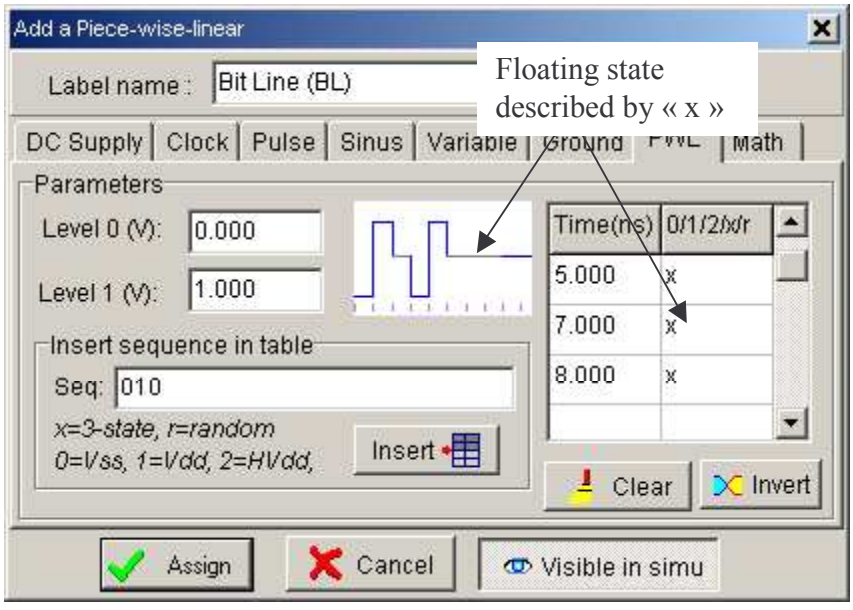


Figure 7-5 : The bit Line pulse used the "x" floating state to enable the reading of the memory cell (RamStatic6T.MSK)

SIMULATION. The simulation parameters correspond to the read and write cycle in the RAM. The proposed simulation steps consist in writing a “0”, a “1”, and then reading the “1”. In a second phase, we write a “1”, a “0”, and read the “0”. The *Bit Line* and *~Bit Line* signals are controlled by pulses (Figure 7-5). The floating state is obtained by inserting the letter "x" instead of 1 or 0 in the description of the signal.

The simulation of the RAM cell is proposed in figure 7-6. At time 0.0, *Data* reaches an unpredictable value of 1, after an unstable period. Meanwhile, *~Data* reaches 0. At time 0.5 ns, the memory cell is selected by a 1 on *Word Line*. As the *Bit Line* information is 0, the memory cell information *Data* goes down to 0. At time 1.5 ns, the memory cell is selected again. As the *Bit Line* information is now 1, the memory cell information *Data* goes to 1. During the read cycle, in which *Bit Line* and *~Bit Line* signals are floating, the memory sets these wires respectively to 1 and 0, corresponding to the stored values.

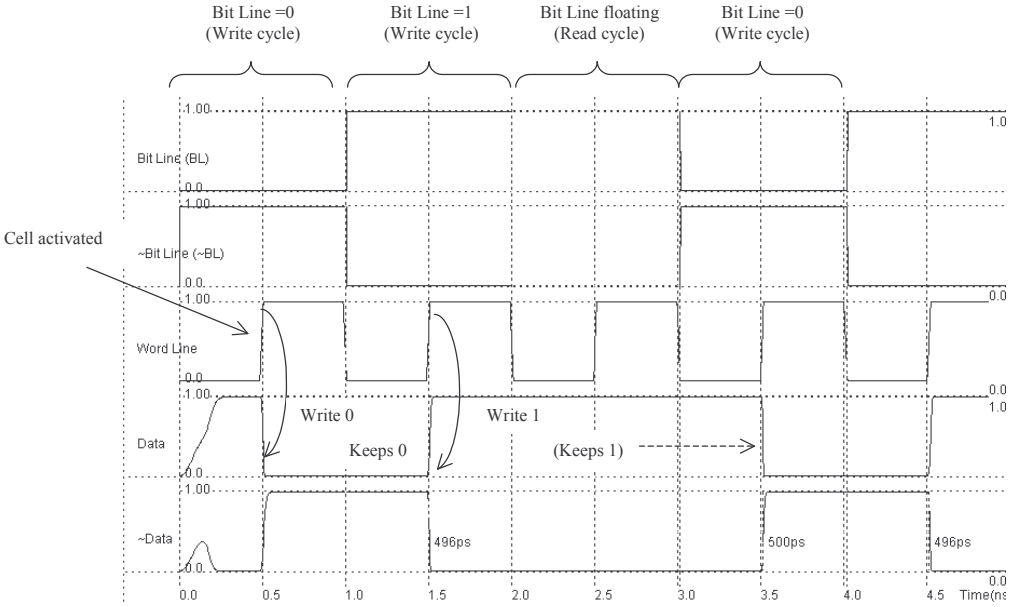


Figure 7-6 : Write cycle for the static RAM cell (RamStatic6T.MSK).

Selection Circuits

The row selection circuit decodes the row address and activates one single row. This row is shared by all word line signals of the row. The row selection circuit is based on a multiplexor circuit. One line is asserted while all the other lines are at zero.

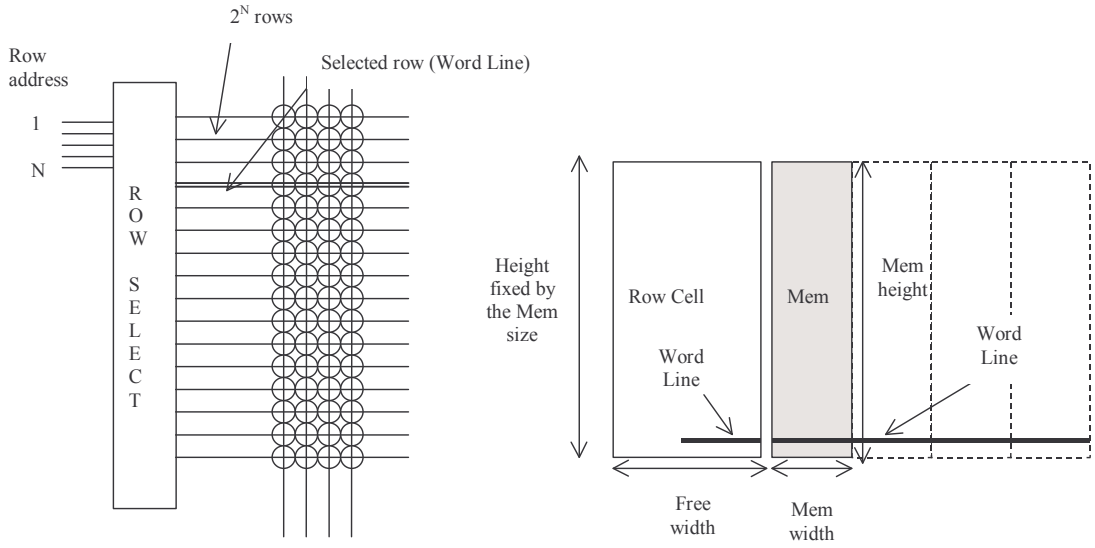


Figure 7-7 : The row selection circuit

In the row selection circuit for the 16x4 array, we simply need to decode a two-bit address. Using AND gates is one simple solution. In the case of a very large number of address lines, the decoder is split into sub-decoders, which handle a reduced number of address lines.

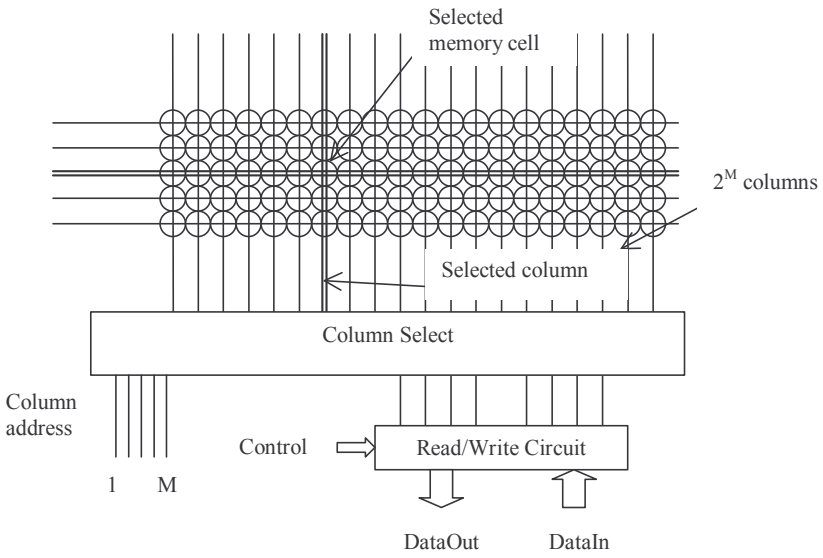


Figure 7-8 : The column selection circuit principles

The column decoder selects a particular column in the memory array to read the contents of the selected memory cell (Figure 7-8) or to modify its contents. The column selector is based on the same principles as those of the row decoder. The major modification is that the data flows both ways, that is either from the memory cell to the *DataOu*t signal (Read cycle), or from the *DataIn* signal to the cell (Write cycle).

A Complete 64 bit SRAM

The 64 bit SRAM memory interface is shown in figure 7-9. The 64 bits of memory are organized in words of 4 bits, meaning that *DataIn* and *DataOut* have a 4 bit width. Each data *D0..D15* occupies 4 contiguous memory cells in the array. Four address lines are necessary to decode one address among 16. The memory structure requires two address lines *A0* and *A1* for the word lines *WL[0]..WL[3]* and two address lines *A2* and *A3* for the bit line selection. The final layout of the 64 bit static RAM is proposed in Figure 7-10.

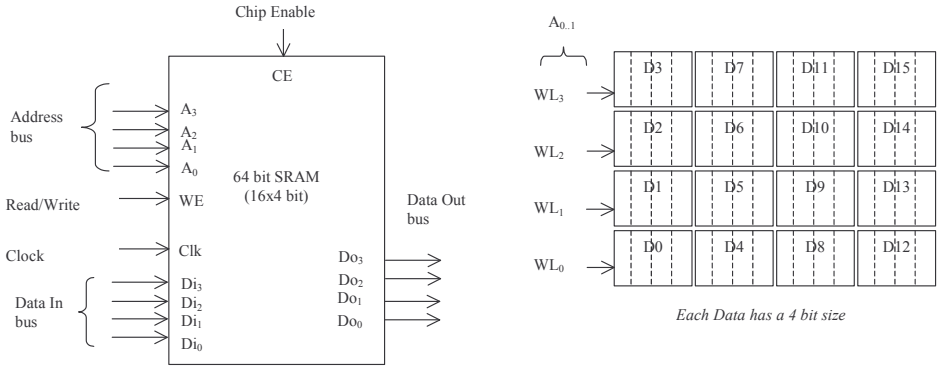


Figure 7-9 : The architecture of the 64 bit RAM (RAM64.MSK)

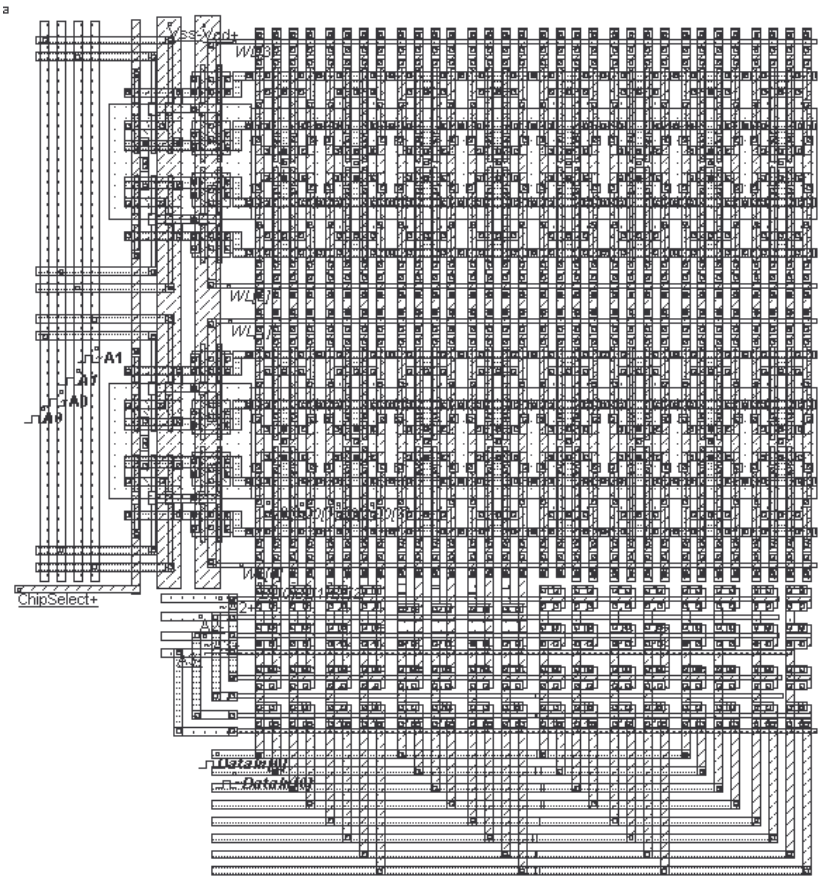


Figure 7-10 : The complete RAM layout (RAM64.MSK)

Dynamic RAM Memory

The dynamic RAM memory has only one transistor, in order to improve the memory matrix density by almost one order of magnitude. The storage element is no longer the stable inverter loop, as for the static RAM, but only a capacitor *Cs*, also called the storage capacitor. The write and hold operation for a "1" is shown in figure 7-11. The data is set on the bit line, the word line is then activated and *Cs* is charged. As the pass transistor is n-type, the analog value reaches *VDD-Vt*. When *WL* is inactive, the storage capacitor *Cs* holds the "1".

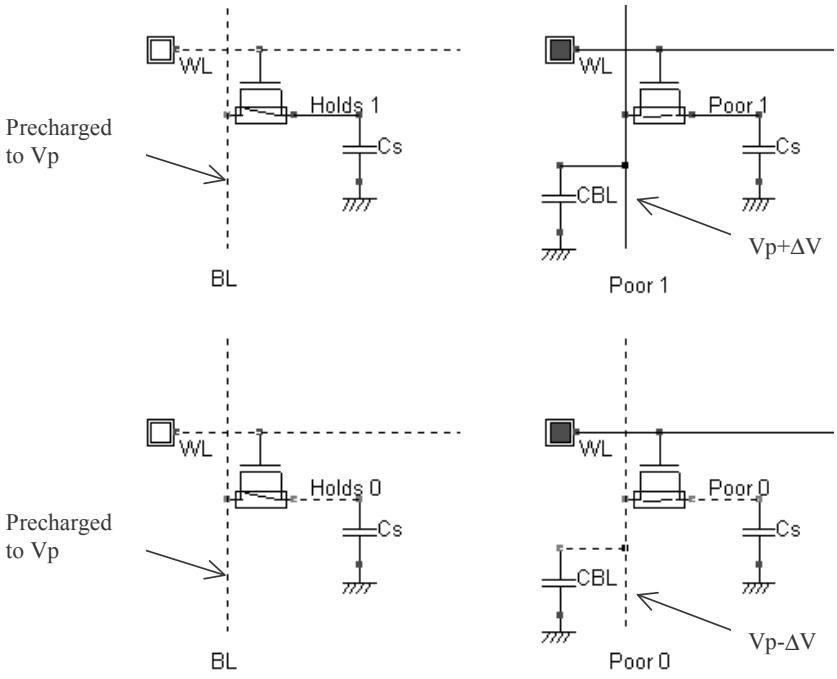


Figure 7-11 : Simulation of the Read cycle for the 1 transistor dynamic RAM cell (RAM1T.SCH)

The reading cycle is destructive for the stored information. Suppose that *Cs* holds a 1. The bit line is precharged to a voltage *Vp* (Usually around VDD/2). When the word line is active, a communication is established between the bit line, loaded by capacitor *CBL*, and the memory, loaded by capacitor *CS*. The charges are shared between these nodes, and the result is a small increase of the voltage *Vp* by *ΔV*, thanks to the injection of some charges from the memory.

The cross-section of the DRAM capacitor is given in figure 7-12. The bit line is routed in metal2, and is connected to the cell through a metal1 and diffusion contact. The word line is the polysilicon gate. On the right side, the storage capacitor is a sandwich of conductor material connected to the diffusion, a thin oxide (SiO2 in this case) and a second conductor that fills the capacitor and is connected to ground by a contact to the first level of metal. The capacitance is around 20fF in this design. Higher capacitance values may be obtained using larger option layer areas, at the price of a lower cell density.

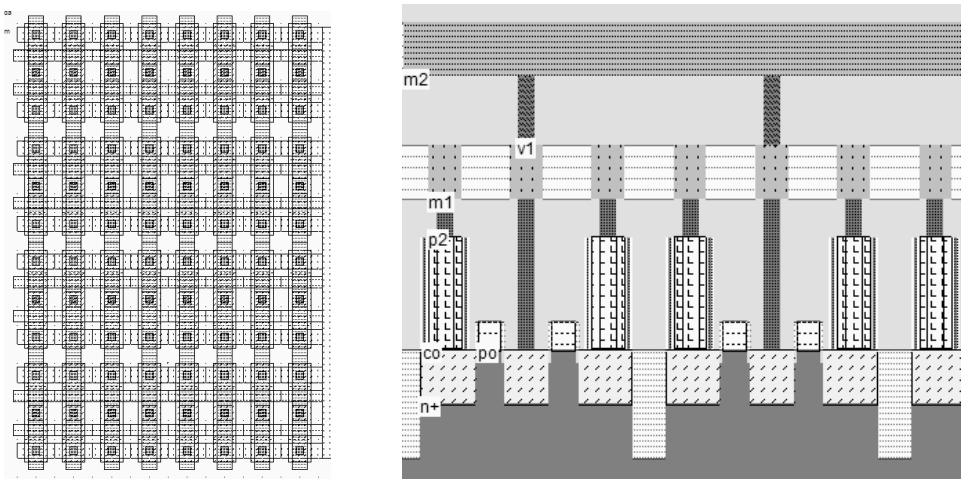


Figure 7-12 : The stacked capacitor cell and its cross-section (DramEdram.MSK)

EEPROM

The basic element of an EEPROM (Electrically Erasable PROM) memory is the floating-gate transistor. The concept was introduced several years ago for the EPROM (Erasable PROM). It is based on the possibility of trapping electrons in an isolated polysilicon layer placed between the channel and the controlled gate. The charges have a direct impact on the threshold voltage of a double-gate device. When there is no charge in the floating gate (Figure 7-13, upper part), the threshold voltage is low, meaning that a significant current may flow between the source and the drain, if a high voltage is applied on the gate. However, the channel is small as compared to a regular MOS, and the Ion current is 3 to 5 times lower, for the same channel size.

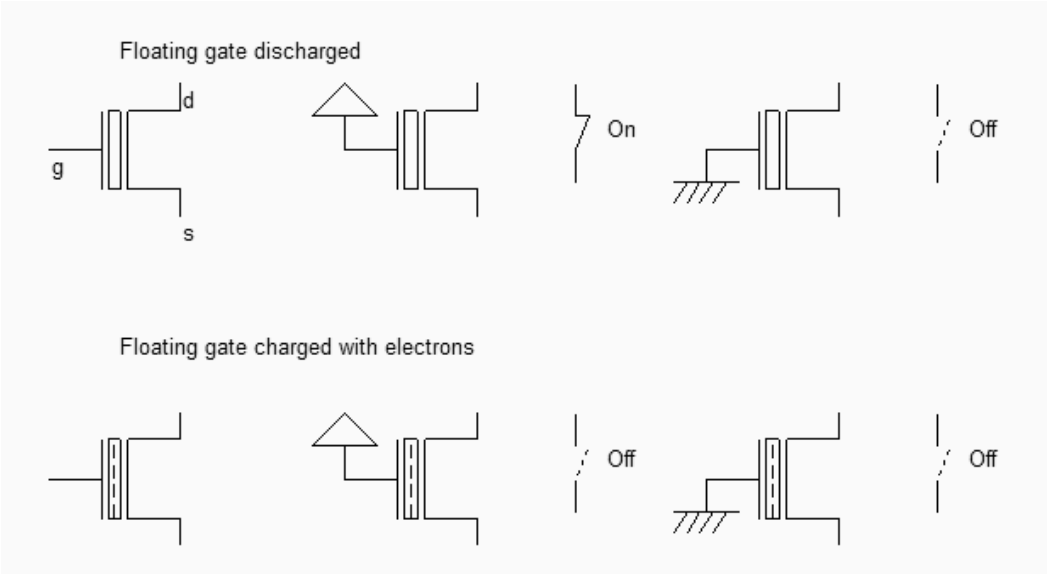


Figure 7-13 : The two states of the double gate MOS (EepromExplain.SCH)

When charges are trapped in the floating polysilicon layer (Figure 7-14, left), the threshold voltage is high, almost no current flows through the device, independently of the gate value. As a matter of fact, the electrons trapped in the floating gate prevent the creation of the channel by repealing channel electrons. Data retention is a key feature of EEPROM, as it must be guaranteed for a wide range of temperatures and operating conditions. Optimum electrical properties of the ultra thin gate oxide and inter-gate oxide are critical for data retention. The typical data retention of an EEPROM is 10 years.

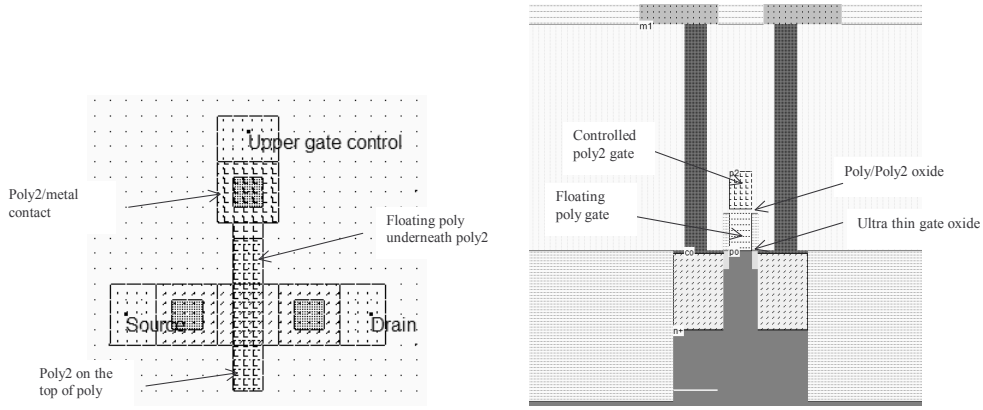


Figure 7-14 : The double gate MOS generated by Microwind (Eeprom.MSK)

The double gate MOS layout is shown in figure 7-14. The structure is very similar to the n-channel MOS device, except for the supplementary *poly2* layer on top of the polysilicon. The lower polysilicon is unconnected, resulting in a floating node. Only the *poly2* upper gate is connected to a metal layer through a *poly2/metal* contact situated at the top. The cross-section of figure 7-14 right reveals the stacked *poly/poly2* structure, with a thin oxide in between.

Flash Memories

Flash memories are a variation of EEPROM memories. Flash arrays can be programmed electrically bit-by-bit but can only be erased by blocks. Flash memories are based on a single double poly MOS device, without any selection transistor (Figure 7-15). The immediate consequence is a more simple design, which leads to a more compact memory array and more dense structures. Flash memories are commonly used in micro-controllers for the storage of application code, which gives the advantage of non volatile memories and the possibility of reconfiguring and updating the code many times.

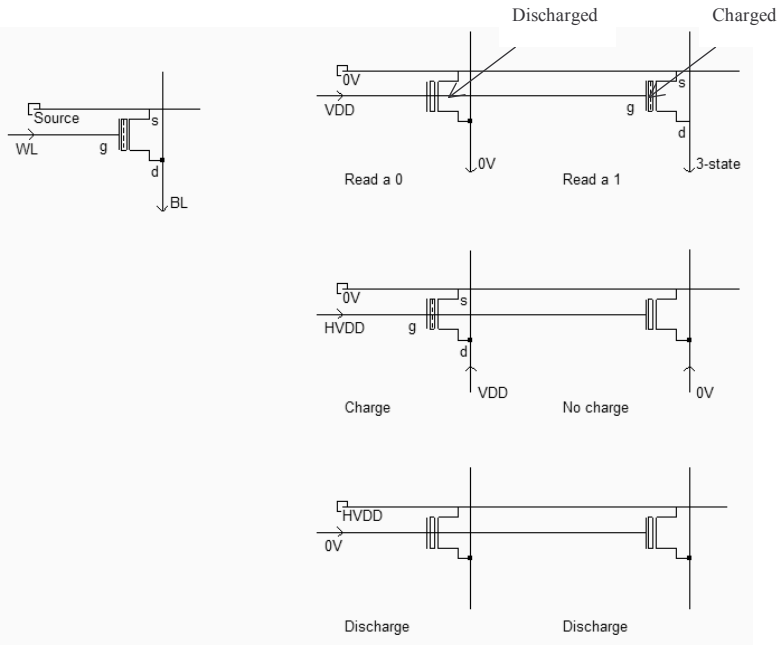


Figure 7-15 : The flash memory point and the principles for charge/discharge (FlashMemory.SCH)

The Flash memory point usually has a "T-shape", due to an increased size of the source for optimum tunneling effect. The horizontal polysilicon2 is the bit line, the vertical metal2 is the word line which links all drain regions together. The horizontal metal line links all sources together (7-16).

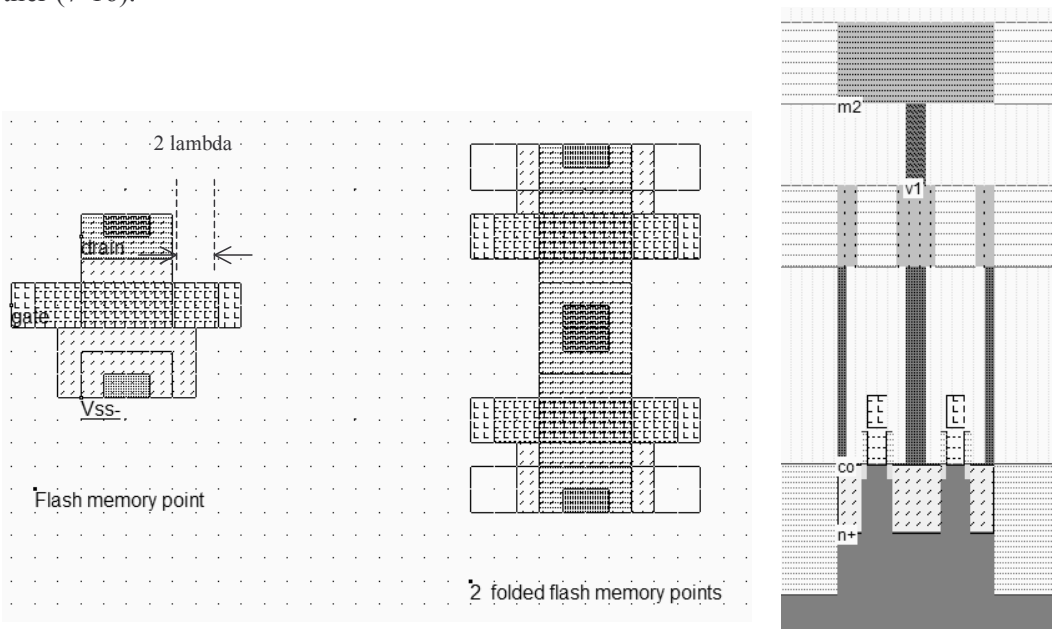


Figure 7-16 : The flash memory point and the associated cross-section (Flash8x8.MSK)

Memory Interface

All inputs and outputs of the RAM are synchronized to the rise edge of the clock, and more than one word can be read or written in sequence. The typical chronograms of a synchronous RAM are shown in figure 7-17. The active edge of the clock is usually the rise edge. One read cycle includes 3 active clock edges in the example shown in figure 7-17. The row address selection is active at the first rise edge, followed by the column address selection. The data is valid at the third fall edge of the system clock.

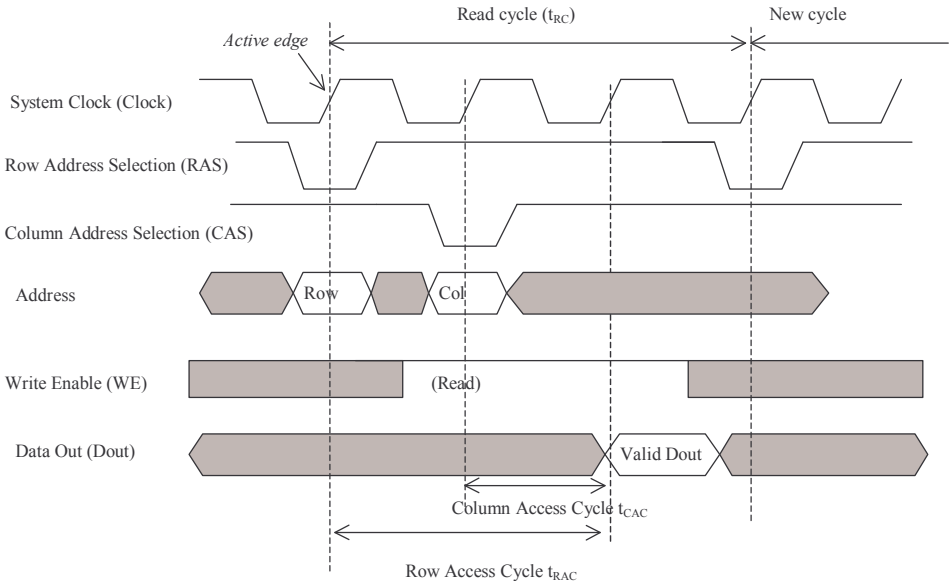


Figure 7-17 : Synchronous RAM timing diagram

Added Features in the Full version

World of memories	Semiconductor memories are vital components in modern integrated circuits. The introductory part details the main families of memories.
Memories	Compact memory cell obtained by sharing all possible contacts: the supply contact, the ground contact and the bit line contacts. Detailed information about ROM memories.
Double-gate MOS	The programming of a double-poly transistor involves the transfer of electrons from the source to the floating gate through the thin oxide. Details are provided on the programming and charge removal.
Ferroelectric RAM	FRAM memories are the most advanced of the Flash memory challengers. The FRAM memory point is based on a two-state ferroelectric insulator. A complete description and simulation of the FRAM is proposed.
Interfacing	Some information is provided about the Double data Rate memories, which involve both the rise and fall edge of the clock.

8 Analog Cells

This chapter deals with analog basic cells, from the simple resistor and capacitor to the operational amplifier. A very complete description of analog cells may be found in [Razavi], and details on analog layout techniques may be found in [Hastings].

Resistor

An area-efficient resistor available in CMOS process consists of a strip of polysilicon. The resistance between *s1* and *s2* is usually counted in a very convenient unit called "ohm per square", noted Ω/ . The default value polysilicon resistance per square is 10Ω, which is quite small, but rises to 200Ω if the salicide material is removed (Figure 8-1).

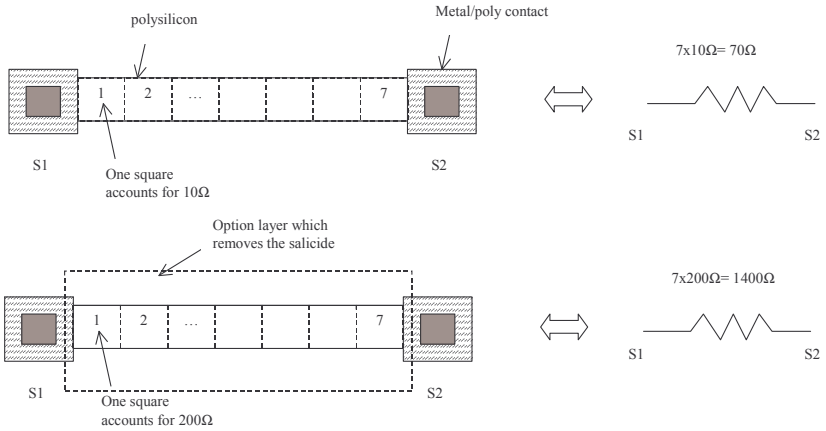


Figure 8-1 : The polysilicon resistance with unsalicide option

In the cross-section shown in figure 8-2, the salicide material deposited on the upper interface between the polysilicon layer and the oxide creates a metal path for current that reduces the resistance dramatically. Notice the shallow trench isolation and surrounding oxide that isolate the resistor from the substrate and other conductors, enabling very high voltage biasing (up to 100V). However, the oxide is a poor thermal conductor which limits the power dissipation of the polysilicon resistor.

The salicide is part of the default process, and is present at the surface of all polysilicon areas. However, it can be removed thank to an option layer programmed by a double click in the option layer box, and a tick at "Remove Salicide". In the example shown in figure 8-3, the default resistance is 76Ω, and the unsalicide resistance rises to 760Ω.

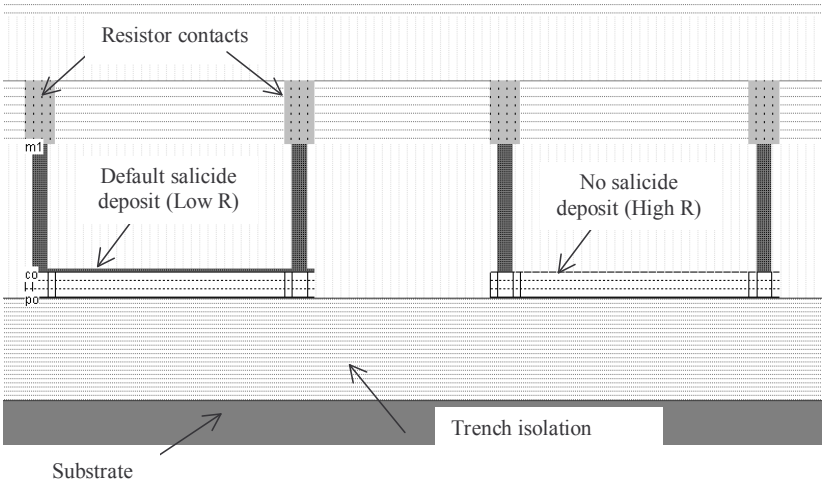


Figure 8-2 : Removing the salicide material to increase the sheet resistance (ResPoly.MSK)

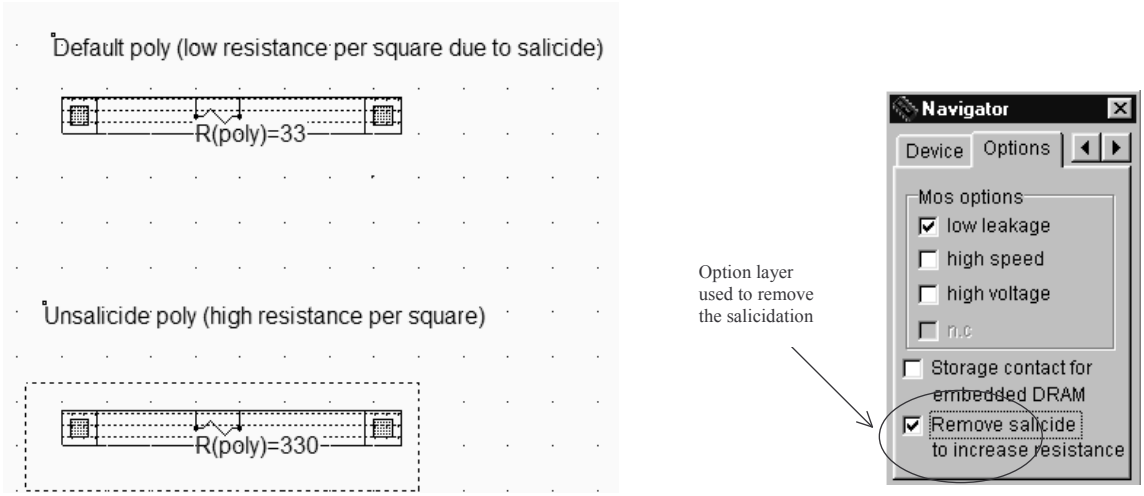


Figure 8-3 : Removing the salicide material thanks to an option layer

Other resistors consist of N+ or P+ diffusions. An interesting feature of diffusion resistor is the ability to combine a significant resistance value and a diode effect. The diffusion resistor is used in input/output protection devices.

The resistor value varies because of lithography and process variations. In the case of the poly resistance, the width, height and doping may vary (Figure 8-4 left). Polysilicon resistors are rarely designed with the minimum 2 lambda width, but rather 4 or 6 lambda, so that the impact of the width variations is smaller. But the equivalent resistance is smaller, meaning less silicon efficiency. A variation ΔW of 0.2λ on both edges results in a 20% variation of the resistance on a 2λ width resistor, but only a 10% variation for a larger resistor designed with a width of 4λ.

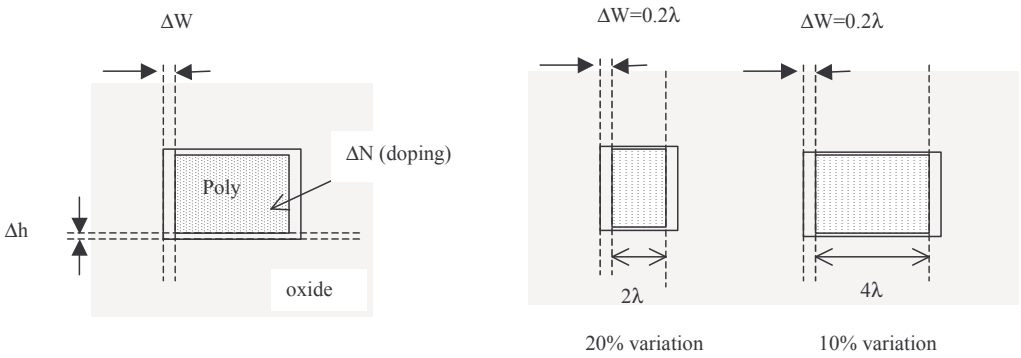


Figure 8-4 : Resistance variations with the process

Capacitor

Capacitors are used in analog design to build filters, compensation, decoupling, etc.. Ideally, the value of the capacitor should not depend on the bias conditions, so that the filtering effect would be situated at constant frequencies.

Diodes in reverse mode exhibit a capacitor behavior, however, the capacitance value is strongly dependent on the bias conditions. A simple N+ diffusion on a P-substrate is a NP diode, which may be considered as a capacitor as long as the N+ region is polarized at a voltage higher than the P-substrate voltage which is usually the case as the substrate is grounded (0V). In 0.12µm, the capacitance is around 300aF/µm2 (1 atto-Farad is equal to 10^{-18} Farad).

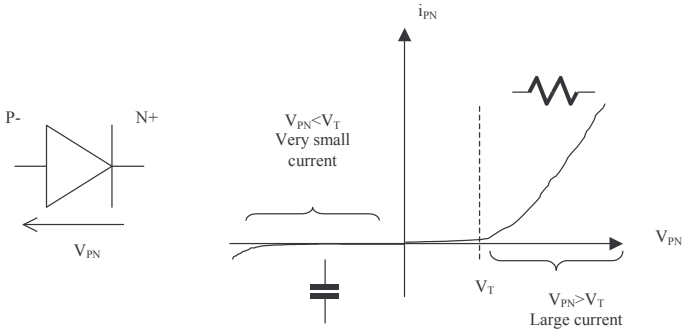


Figure 8-5 : The diffusion over substrate as a non-linear capacitor (Capa.MSK)

The typical variation of the capacitance with the diffusion voltage V_N is given in figure 8-6. The capacitance per µm2 provided in the electrical rules is a rude approximation of the capacitance variation. A large voltage difference between V_N and the substrate result in a thick zone with empty charges, which corresponds to a thick insulator, and consequently to a small capacitance. When V_N is lowered, the zone with empty charges is reduced, and the capacitance increases. If V_N goes lower than the substrate voltage, the diode starts to conduct.

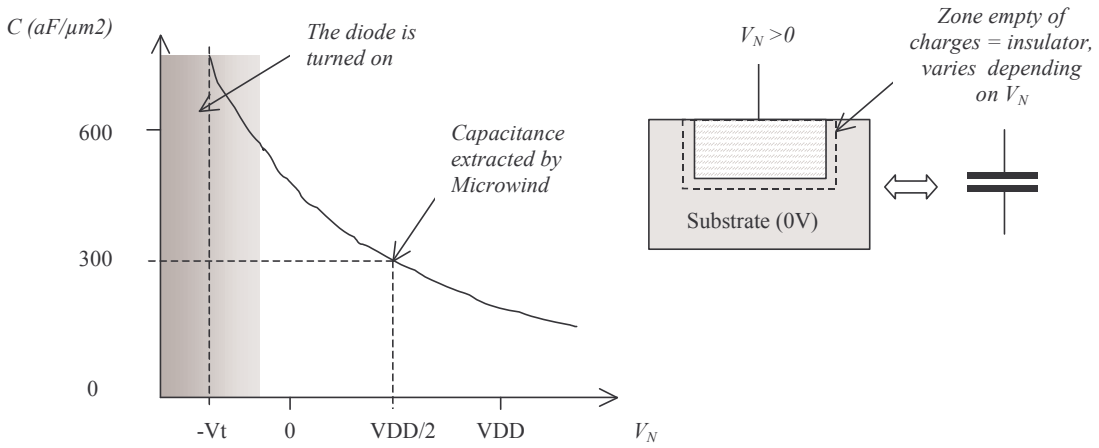


Figure 8-6 : The diffusion capacitance varies with the polarization voltage

Poly-Poly2 Capacitor

Most deep-submicron CMOS processes incorporate a second polysilicon layer (poly2) to build floating gate devices for EEPROM. An oxide thickness around 20 nm is placed between the poly and poly2 materials, which induces a plate capacitor around 1,7 fF/µm^2. In MICROWIND , the command "**Edit → Generate → Capacitor**" gives access to a specific menu for generating capacitor (Figure 8-7). The parameter in the design rule file (cmos65nm.RUL for the 65-nm technology) used to configure the poly-poly2 capacitor is `CP2PO`.

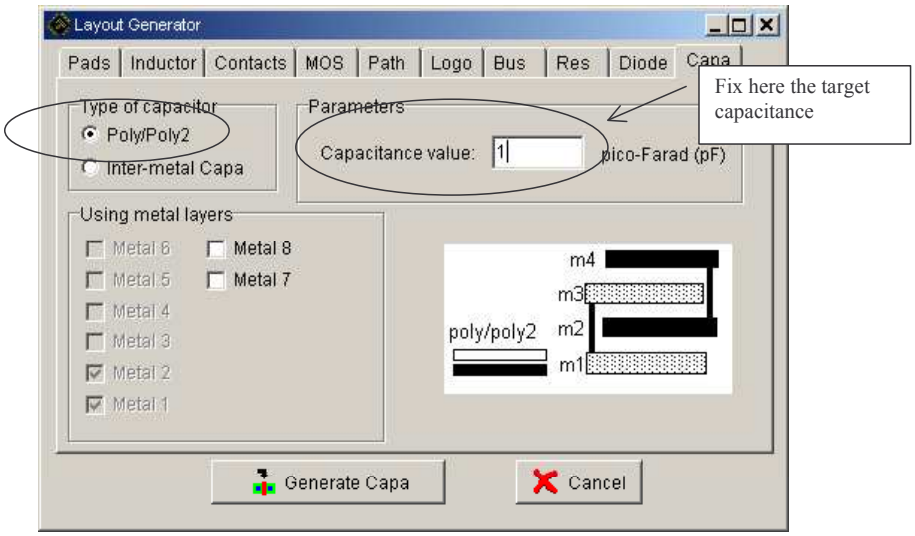


Figure 8-7 : The generator menu handles the design of poly/poly2 capacitor and inter-metal capacitors

The poly/poly2 capacitor simply consists of a sheet of polysilicon and a sheet of poly2, separated by a specific dielectric oxide which is 20-nm in the case of the default CMOS 65-nm process.

Diode-connected MOS

The schematic diagram of the diode-connected MOS is proposed in figure 8-8. This circuit features a high resistance within a small silicon area. The key idea is to build a permanent connection between the drain and the gate. Most of the time, the source is connected to ground in the case of n-channel MOS, and to VDD in the case of p-channel MOS.

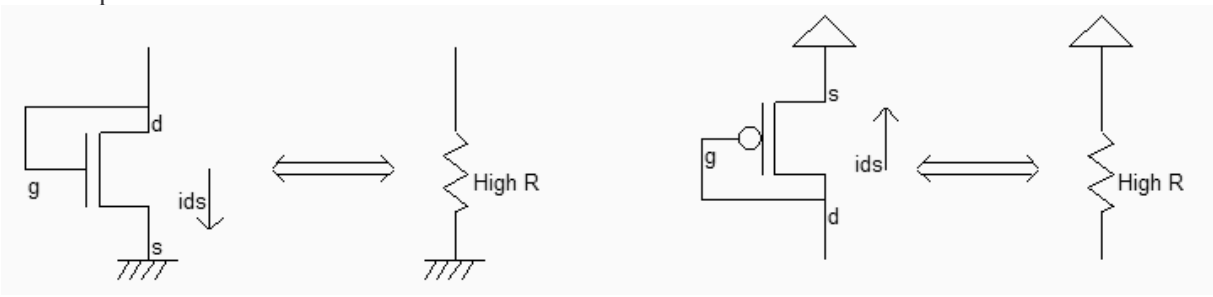


Figure 8-8 : Schematic diagram of the MOS connected as a diode (MosRes.SCH)

To create the diode-connected MOS, the easiest way is to use the MOS generator. Enter a large length and a small width, for example W=0.14µm and L=1.4µm. This sizing corresponds to a long channel, featuring a very high equivalent resistance. Add a poly/metal contact and connect the gate to one diffusion. Add a clock on that node. Add a VSS property to the other diffusion. The layout result is shown in figure 8-9.

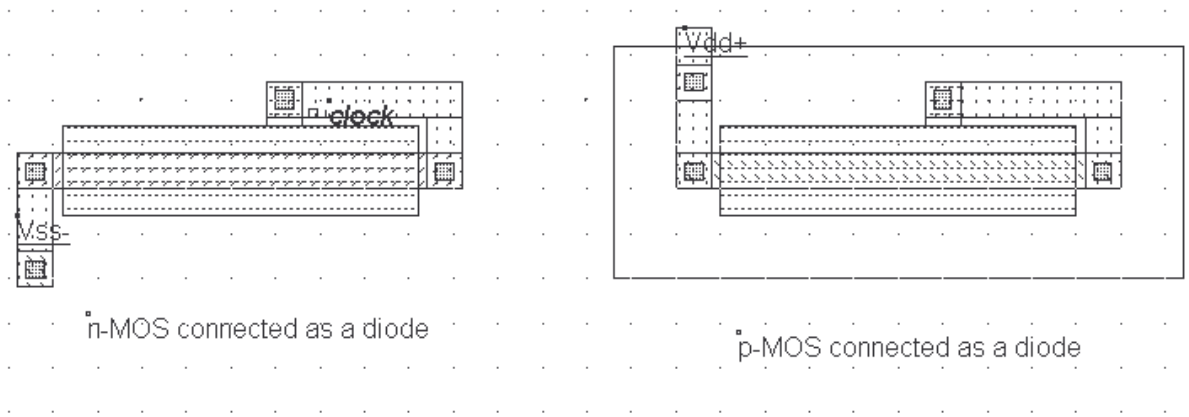


Figure 8-9 : Schematic diagram of the MOS connected as a diode (ResMos.MSK)

Now, click **Simulation on Layout.** In a small window, the MOS characteristics are drawn, with the functional point drawn as a color dot (Figure 8-10). It can be seen that the I/V characteristics correspond to a diode. The resistance is the invert value of the slope in the *Id/Vd* characteristics. For *Vds* larger than 0.6V, the resistance is almost constant. As the current *Ids* increases of 10µA in 0.4V, the resistance can be estimated around 40KΩ. A more precise evaluation is performed by MICROWIND if you draw the slope manually. At the bottom of the screen, the equivalent resistance appears, together with the voltage and current.

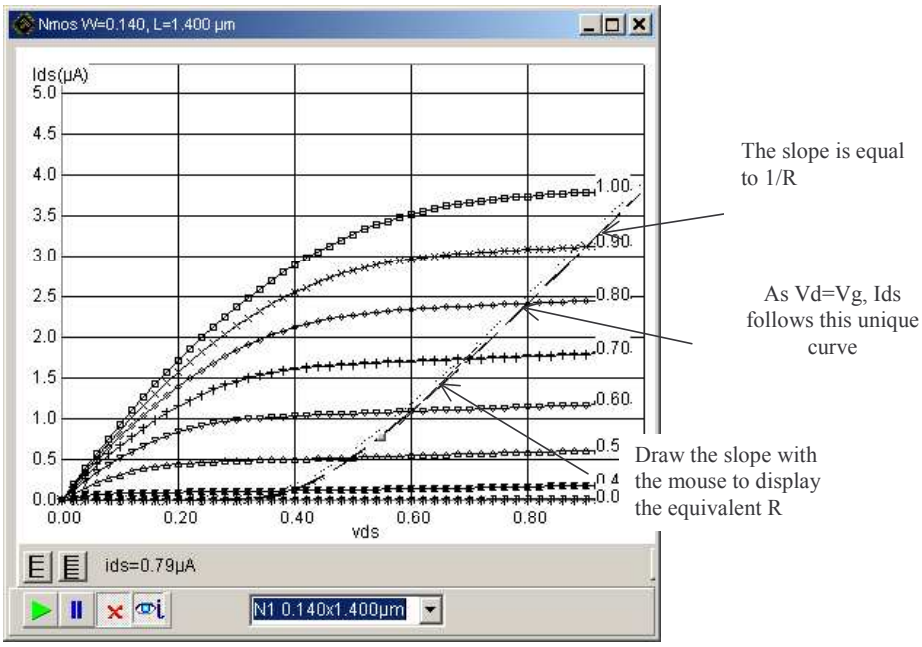


Figure 8-10 : Using the Simulation on Layout to follow the characteristics of the diode-connected MOS (ResMos.MSK)

In summary, the MOS connected as a diode is a capacitance for *Vgs<Vt*, a high resistance when *Vgs* is higher than the threshold voltage *Vt*. The resistance obtained using such a circuit can easily reach 100KΩ in a very small silicon area.

Voltage Reference

The voltage reference is usually derived from a voltage divider made from resistance. The output voltage *Vref* is defined by equation 8-1.

$$V_{ref} = \frac{R_N}{R_N + R_P} V_{DD} \qquad \text{(Eq. 8-1)}$$

with

V_{DD}=power supply voltage (1.0 V in 65-nm)

R_N=equivalent resistance of the n-channel MOS (Ω)

R_P=equivalent resistance of the p-channel MOS (Ω)

Notice that two n-MOS or two p-MOS properly connected feature the same function. P-MOS devices offer higher resistance due to lower mobility, compared to n-channel MOS. Four voltage reference designs are shown in figure 8-11. The most common design uses one p-channel MOS and one n-channel MOS connected as diodes.

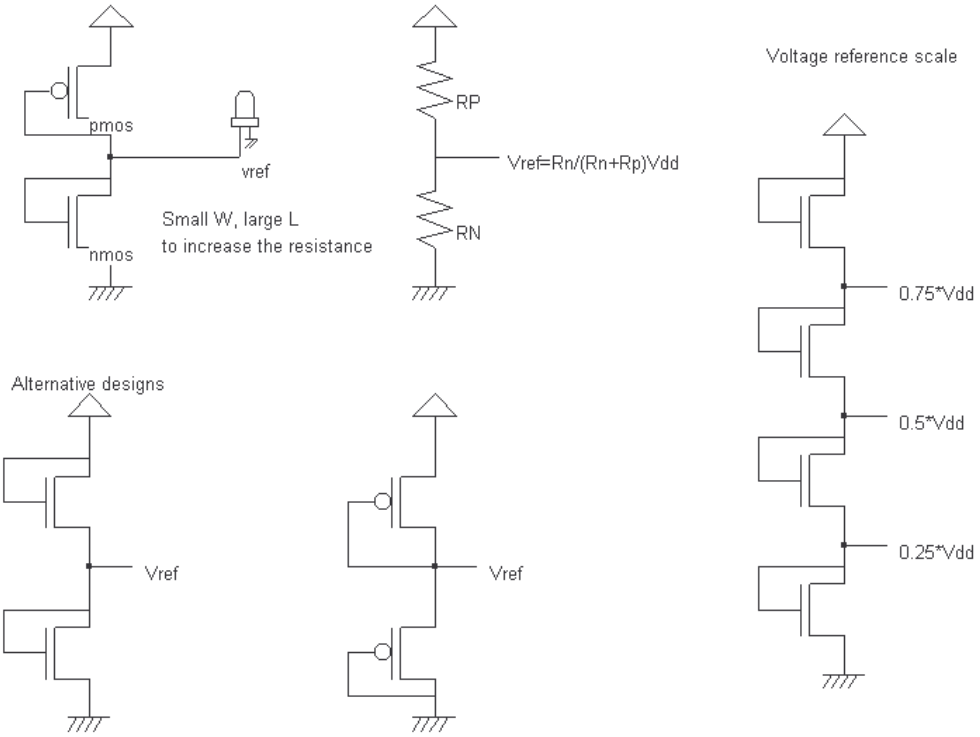


Figure 8-11 : Voltage reference using PMOS and NMOS devices as large resistance

The alternative solutions consist in using two n-channel MOS devices only (Left lower part of figure 8-12), or their opposite built from p-channel devices only. Not only one reference voltage may be created, but also three, as shown in the right part of the figure, which use four n-channel MOS devices connected as diodes.

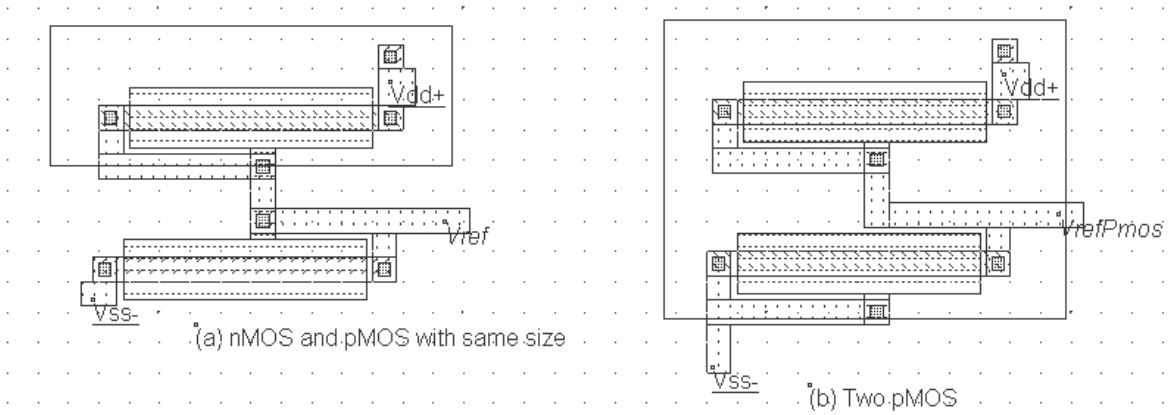


Figure 8-12 : Voltage reference circuits (a) with one nMOS and one pMOS (b) with two pMOS (Vref.MSK)

Amplifier

The goal of the amplifier is to multiply by a significant factor the amplitude of a sinusoidal voltage input *Vin*, and deliver the amplified sinusoidal output *Vout* on a load. The single stage amplifier may consist of a MOS device (we choose here a n-channel MOS) and a load. The load can be a resistance or an inductance. In the circuit, we use a resistance made with a p-channel MOS device with gate and drain connected (Figure 8-13). The pMOS which replaces the passive load is called an active resistance.

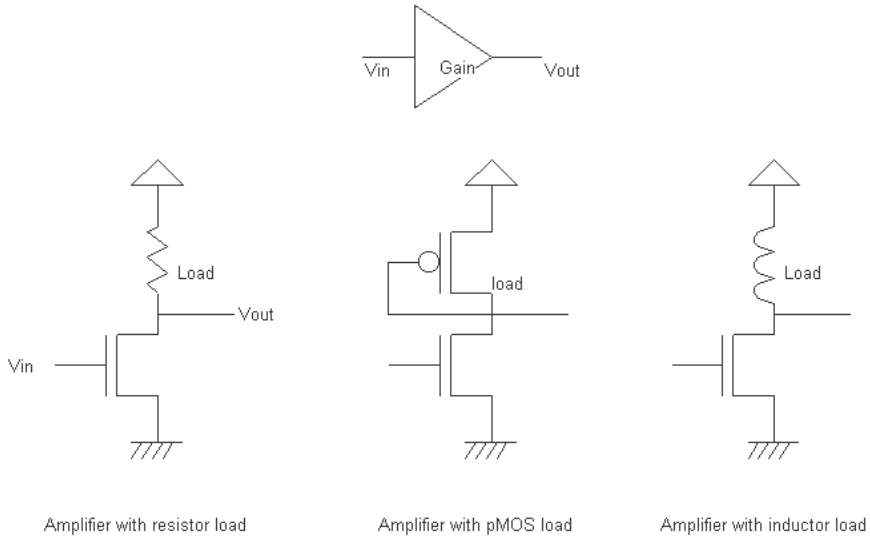


Figure 8-13 : Single stage amplifier design with MOS devices (AmpliSingle.SCH)

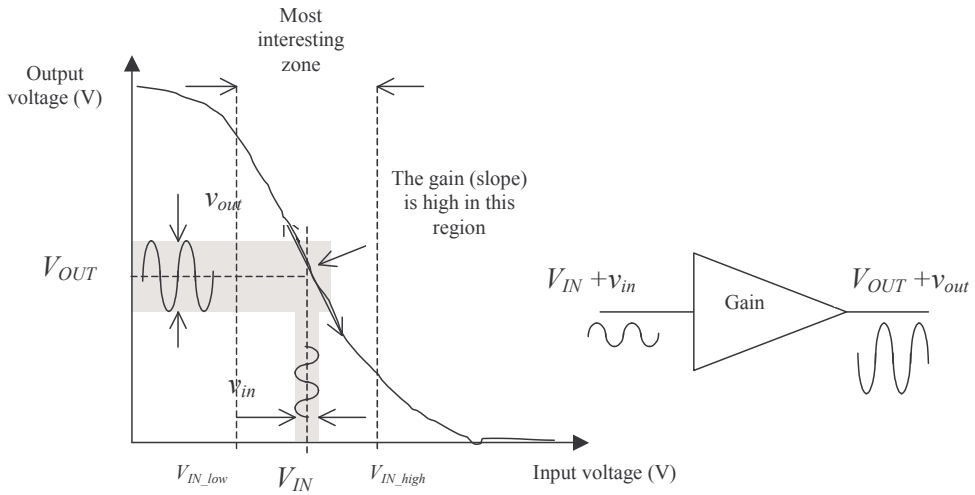


Figure 8-14 : The amplifier has a high gain in a certain input range, where a small input signal vin is amplified to a large signal vout.

The single stage amplifier characteristics between *Vin* and *Vout* have a general shape shown in figure 8-14. The most interesting zone corresponds to the input voltage range where the transfer function has a linear shape, that is between *VIN_low* and *VIN_high*. Outside this voltage range, the behavior of the circuit does not correspond anymore to an amplifier. If we add a small sinusoidal input v_{in} to V_{IN}, a small variation of current i_{ds} is added to the static current I_{DS}, which induces a variation v_{out} of the output voltage V_{OUT}. The link between the variation of current i_{ds} and the variation of voltage v_{in} can be approximated by equation 8-2.

$$i_{ds} = g_m v_{gs} \qquad \text{(Equ. 8-2)}$$

In figure 8-15, a nMOS device with large width and minimum length is connected to a high resistance pMOS load. A 50 mV sinusoidal input (*vin*) is superimposed to the static offset 0.5 V (V_{IN}). What we expect is a 500 mV sinusoidal wave (*vout*) with a certain DC offset (V_{OUT}).

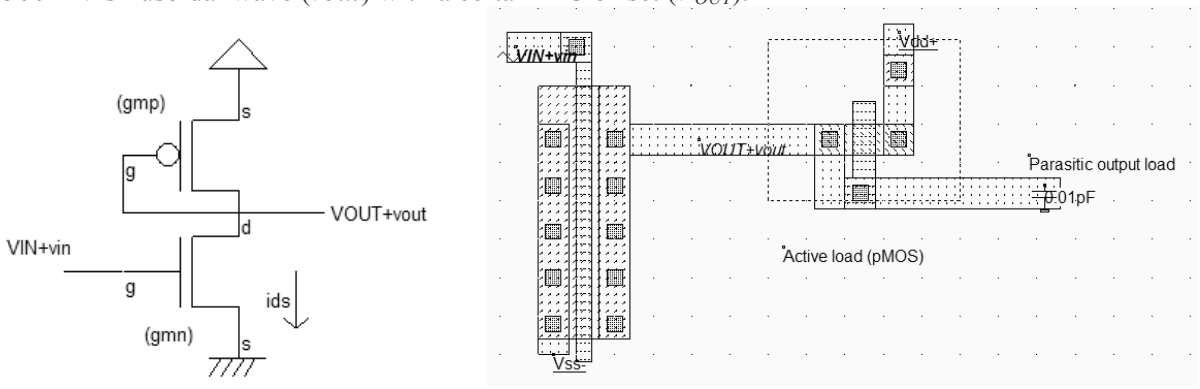


Figure 8-15 : Single stage amplifier layout with a pMOS as a load resistor (AmpliSingle.MSK)

What we need now is to find the characteristics *Vout/Vin* in order to tune the offset voltage V_{IN}. In the simulation window, click **Voltage vs voltage**" and **More**, to compute the static response of the amplifier (Figure 8-16). The range of voltage input that exhibits a correct gain appears clearly. For V_{DS} higher than 0.2 V and lower than 0.4 V, the output gain is around 5. Therefore, an optimum offset value could be 0.30 V. Change the parameter **Offset** of the input sinusoidal wave to place the input voltage in the correct polarization. A gain of 5.0 is observed when the offset V_{IN} is 0.30 V (figure 8-16).

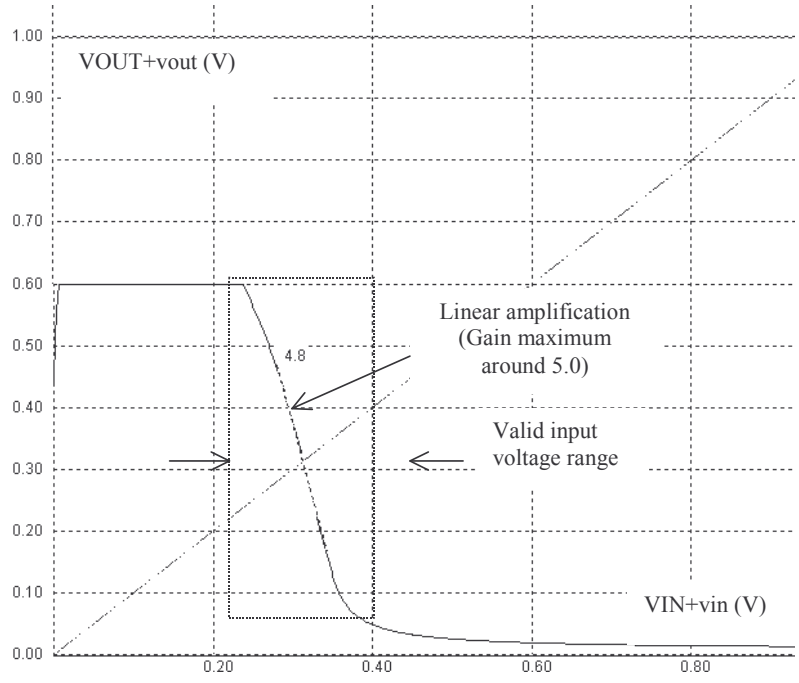


Figure 8-16 : Single stage amplifier static response showing the valid input voltage range (AmpliSingle.MSK)

Simple Differential Amplifier

The goal of the differential amplifier is to compare two analog signals, and to amplify their difference. The differential amplifier formulation is reported below (Equation 8-3). Usually, the gain *K* is high, ranging from 10 to 1000. The consequence is that the differential amplifier output saturates very rapidly, because of the supply voltage limits.

$$Vout = K(Vp - Vm) \qquad \text{(Equ. 8-3)}$$

The schematic diagram of a basic differential amplifier is proposed in figure 8-17. An nMOS device has been inserted between the differential pair and the ground to improve the gain. The gate voltage *Vbias* controls the amount of current that can flow on the two branches. This pass transistor permits the differential pair to operate at lower *Vds*, which means better analog performances and less saturation effects.

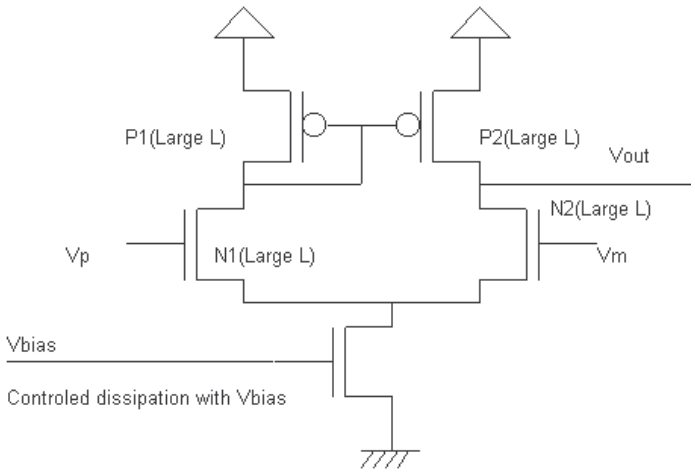


Figure 8-17 : An improved differential amplifier (AmpliDiff.SCH)

The best way to measure the input range is to connect the differential amplifier as a follower, that is *Vout* connect to *Vm*. The *Vm* property is simply removed, and a contact poly/metal is added at the appropriate place to build the bridge between *Vout* and *Vm*. A slow ramp is applied on the input *Vin* and the result is observed on the output. We use again the « Voltage vs. Voltage » to draw the static characteristics of the follower. The BSIM4 model is forced for simulation by a label "BSIM4" on the layout.

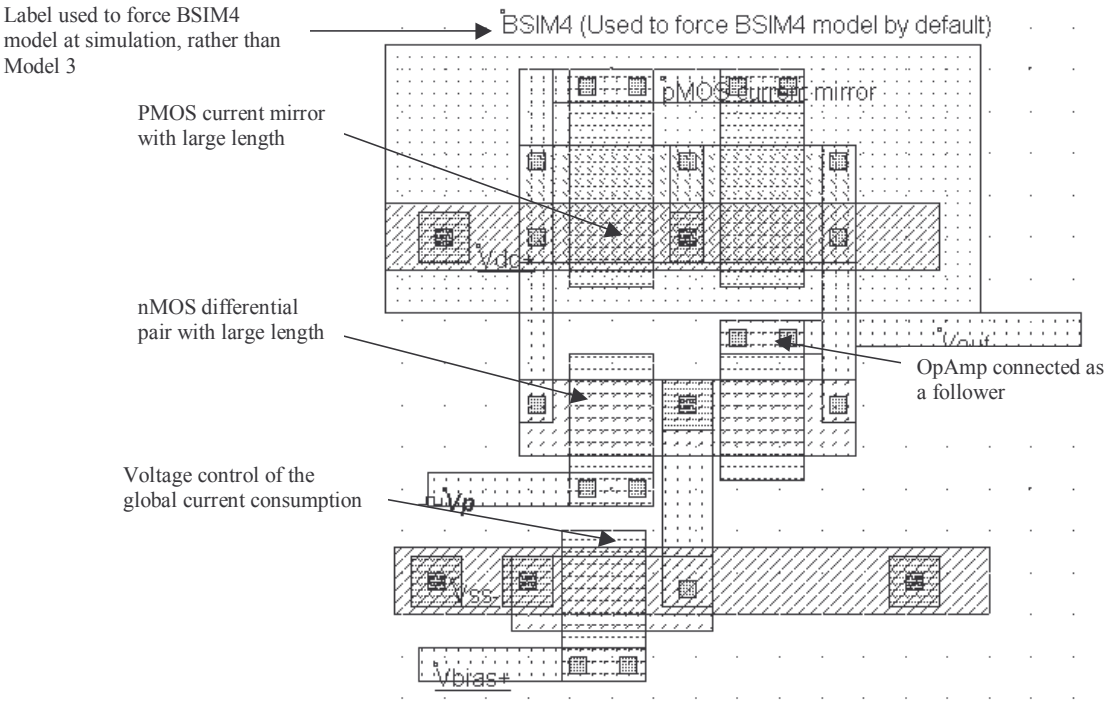


Figure 8-18 : The layout corresponding to the improved differential amplifier (AmpliDiffLargeLength.MSK)

The simulation of the circuit is performed here using the CMOS 0.12µm technology. Use **File → Select Foundry** and select "cmos012.RUL". As can be seen from the resulting simulation reported in figure 8-19, a low *Vbias* features a larger voltage range, specifically at high voltage values. The follower works properly starting 0.4V, independently of the *Vbias* value. A high *Vbias* leads to a slightly faster response, but reduces the input range and consumes more power as the associated nMOS transistor drives an important current. The voltage *Vbias* is often fixed to a value a little higher than the threshold voltage *Vtn*. This corresponds to a good compromise between switching speed and input range.

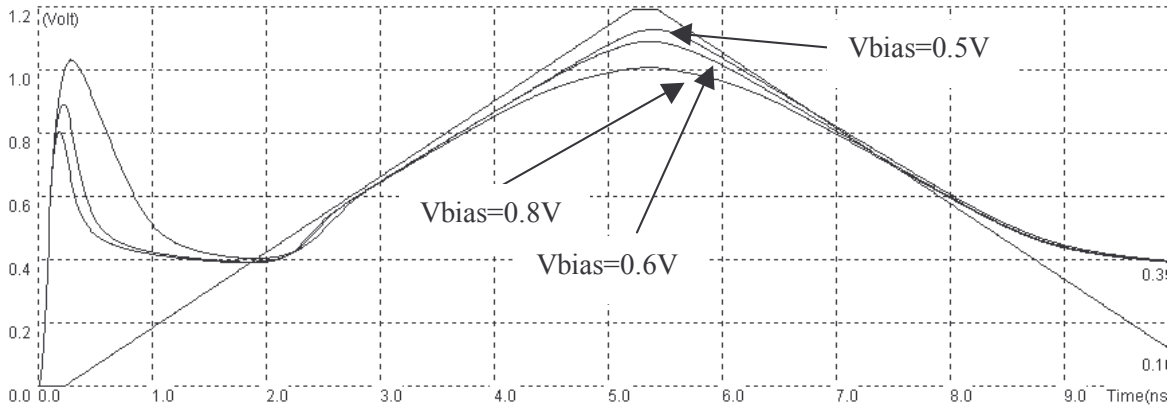


Figure 8-19 : Effect of Vbias on the differential amplifier performance (AmpliDiffVbias.MSK)

Added Features in the Full version

Amplifiers	The push-pull amplifier is built using a voltage comparator and a power output stage. Its schematic diagram and performances are detailed.
Improved layout techniques	A set of design techniques can improve the current mirror behavior: MOS orientation, channel length modulation effects, dummy devices, MOS matching.
Resistor	There exist efficient techniques to reduce the resistance variations within the same chip. Layout techniques which minimize the effects of process variations are presented.
Capacitor	The multiplication of metal layers create lateral and vertical capacitance effects of rising importance. The spared silicon area in upper metal layers may be used for small size capacitance. The implementation of these capacitor is described.
Current Mirror	The current mirror is one of the most useful basic blocs in analog design. It is primarily used to copy currents. The principles and behavior of current mirrors are given in the full version. The cascode current mirror is also presented, which has several advantages over the simple current mirror.

9 Radio Frequency Circuits

On-Chip Inductors

Inductors are commonly used for filtering, amplifying, or for creating resonant circuits used in radio-frequency applications. The inductance symbol in DSCH and MICROWIND is as follows (Figure 9-1).

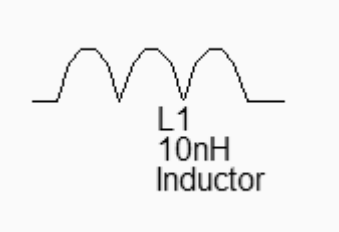


Figure 9-1 : The inductance symbol

The quality factor Q is a very important metric to quantify the resonance effect. A high quality factor Q means low parasitic effects compared to the desired inductance effect. The formulation of the quality factor is not as easy as it could appear. An extensive discussion about the formulation of Q depending on the coil model is given in [Lee]. We consider the coil as a serial inductor *L1*, a parasitic serial resistor *R1*, and two parasitic capacitors *C1* and *C2* to the ground, as shown in figure 9-2. Consequently, the Q factor is approximately given by equation 9-1.

$$Q = \frac{\sqrt{\frac{L1}{(C1+C2)}}}{R1} \qquad \text{(Equ. 9-1)}$$

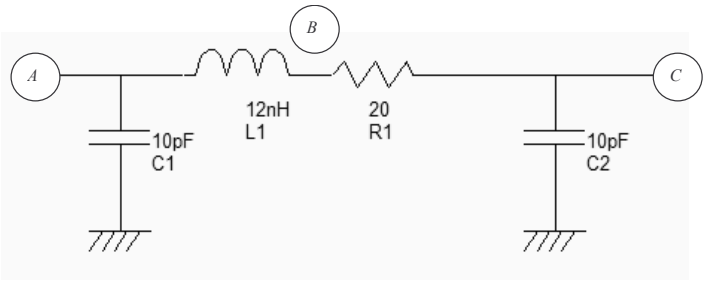


Figure 9-2 : The equivalent model of the 12nH default coil and the approximation of the quality factor Q

The inductor can be generated automatically by MICROWIND using the command **Edit → Generate → Inductor** . The inductance value appears at the bottom of the window, as well as the parasitic resistance and the resulting quality factor Q.

Using the default parameters, the coil inductance approaches 12 nH, with a quality factor of 1,15. The corresponding layout is shown in figure 9-3. Notice the virtual inductance (*L1*) and resistance (*R1*) symbols placed in the layout. The serial inductor is placed between *A* and *B* and a serial resistance between *B* and *C*. If these symbols were omitted, the whole inductor would be considered as a single electrical node.

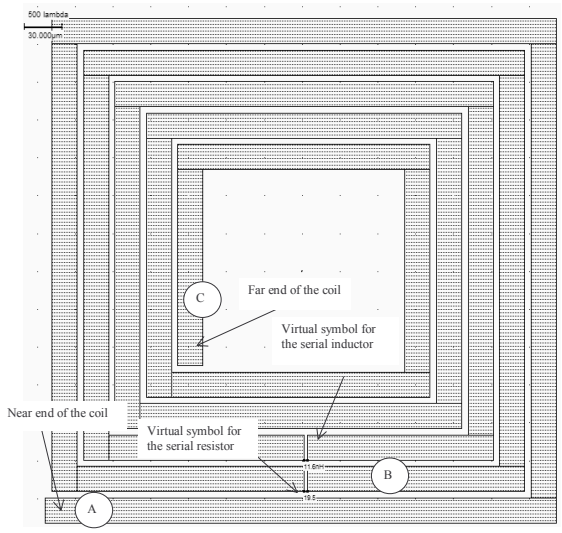


Figure 9-3 : The inductor generated by default (inductor12nH.MSK)

The coil can be considered as a RLC resonant circuit. At very low frequencies, the inductor is a short circuit, and the capacitor is an open circuit (Figure 9-4 left). This means that the voltage at node *C* is almost equal to *A*, if no load is connected to node *C*, as almost no current flows through *R1*. At very high frequencies, the inductor is an open circuit, the capacitor a short circuit (Figure 9-4 right). Consequently, the link between *C* and *A* tends towards an open circuit.

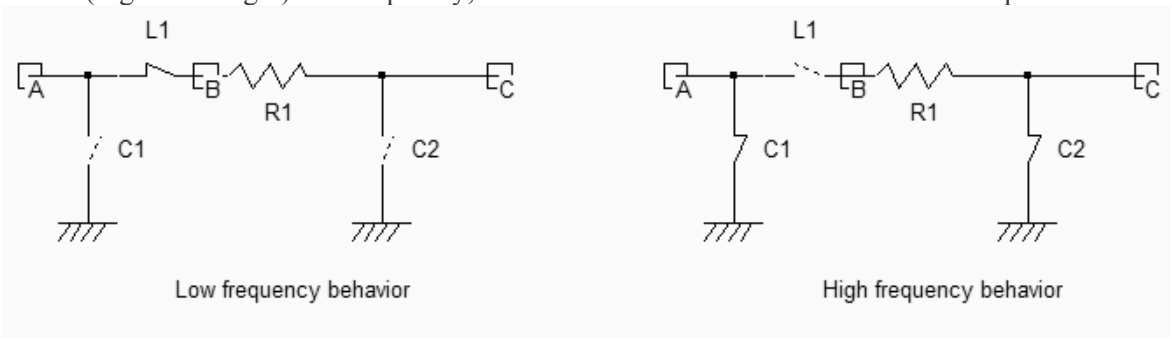


Figure 9-4 : The behavior of a RLC circuit at low and high frequencies (Inductor.SCH)

At a very specific frequency the LC circuit features a resonance effect. The theoretical formulation of this frequency is given by equation 9-2.

$$f_r = \frac{1}{2\pi\sqrt{L1(C1+C2)}} \qquad \text{(Equ. 9-2)}$$

We may see the resonance effect of the coil and an illustration of the quality factor using the following procedure. The node *A* is controlled by a sinusoidal waveform with increased frequency (Also called "chirp" signal). We specify a very small amplitude (0.1 V), and a zero offset.

The resonance can be observed when the voltage at nodes *B* and *C* is higher than the input voltage *A*. The ratio between *B* and *A* is equal to the quality factor Q.

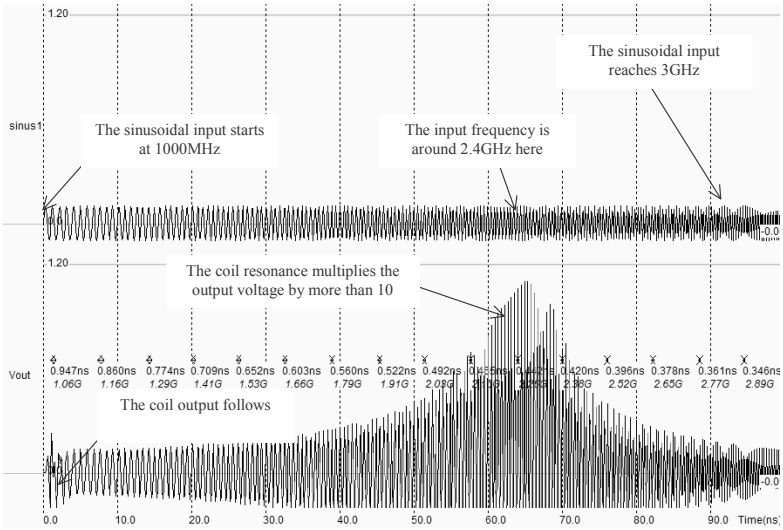


Figure 9-5 : The behavior of a RLC circuit near resonance (Inductor3nHighQ.MSK)

Power Amplifier

The power amplifier is part of the radio-frequency transmitter, and is used to amplify the signal being transmitted to an antenna so that it can be received at the desired distance. Most CMOS power amplifiers are based on a single MOS device, loaded with a "Radio-Frequency Choke" inductor L_{RFC}, as shown in figure 9-6.

The inductor serves as a load for the MOS device (At a given frequency *f*, the inductor is equivalent to a resistance $L.2\pi f$), with two significant advantages as compared to the resistor: the inductor do not consume DC power, and the combination of the inductor and the load capacitor *CL* creates a resonance. The power is delivered to the load *RL*, which is often fixed to 50 Ω. This load is for example the antenna monopole, which can be assimilated to a radiation resistance, as described in the previous section. The resonance effect is obtained between L_{RFC} and C_L.

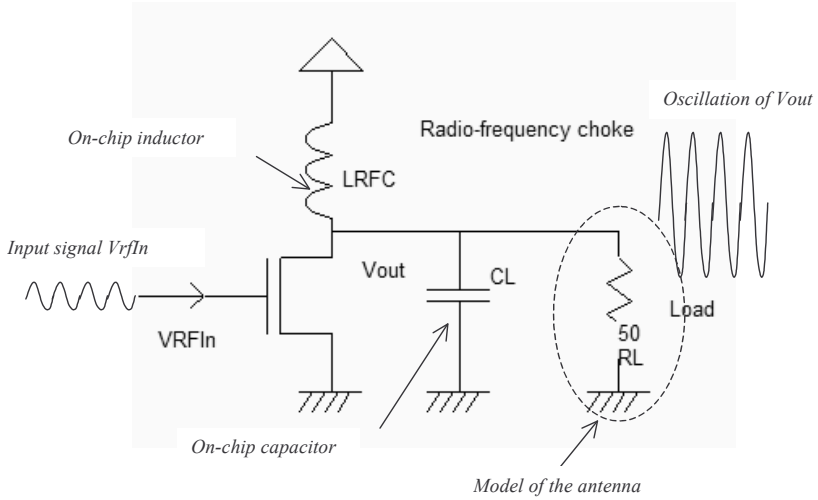


Figure 9-6 : The basic diagram of a power amplifier (PowerAmp.SCH)

An example of powerful MOS device is shown in figure 9-7. The maximum current is close to 40 mA. A convenient way to generate the polarization ring consists in using the Path generator command, and selection the option **Metal and p-diffusion**. Then draw the location for the polarization contacts in order to complete the ring.

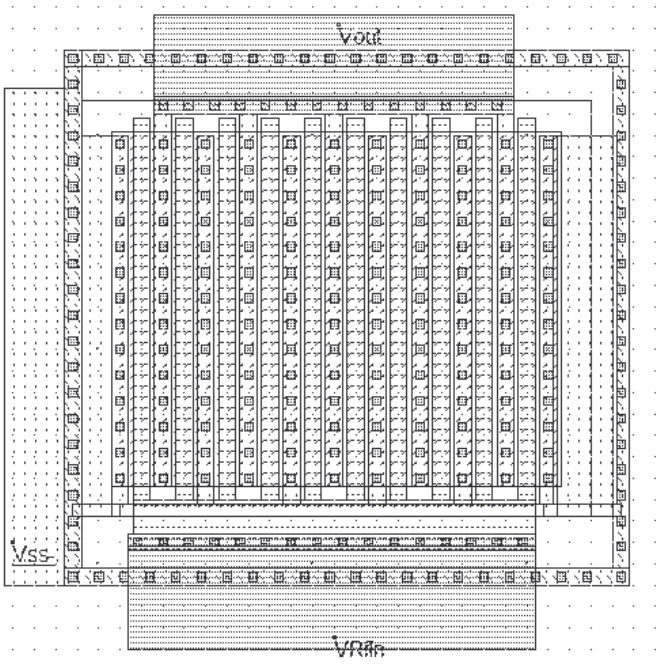


Figure 9-7 : The layout of the power MOS also includes a polarization ring, and the contacts to metal2 connections to VRF_in and VOut (PowerAmplifier.MSK)

The distinction between class A,B,AB, etc.. amplifiers is mainly given with the polarization of the input signal. A Class A amplifier is polarized in such a way that the transistor is always conducting. The MOS device operates almost linearly.

The sinusoidal input offset is 0.7 V, the amplitude is 0.4V. The power MOS functional point trajectory is plotted in figure 9-8, and is obtained using the command **Simulate on Layout**. We see the evolution of the functional point with the voltage parameters: as *Vgs* varies from 0.3 V to 1.1 V, *Ids* fluctuates between 20 mA and 50 mA. The MOS device is always conducting, which corresponds to class A amplifiers.

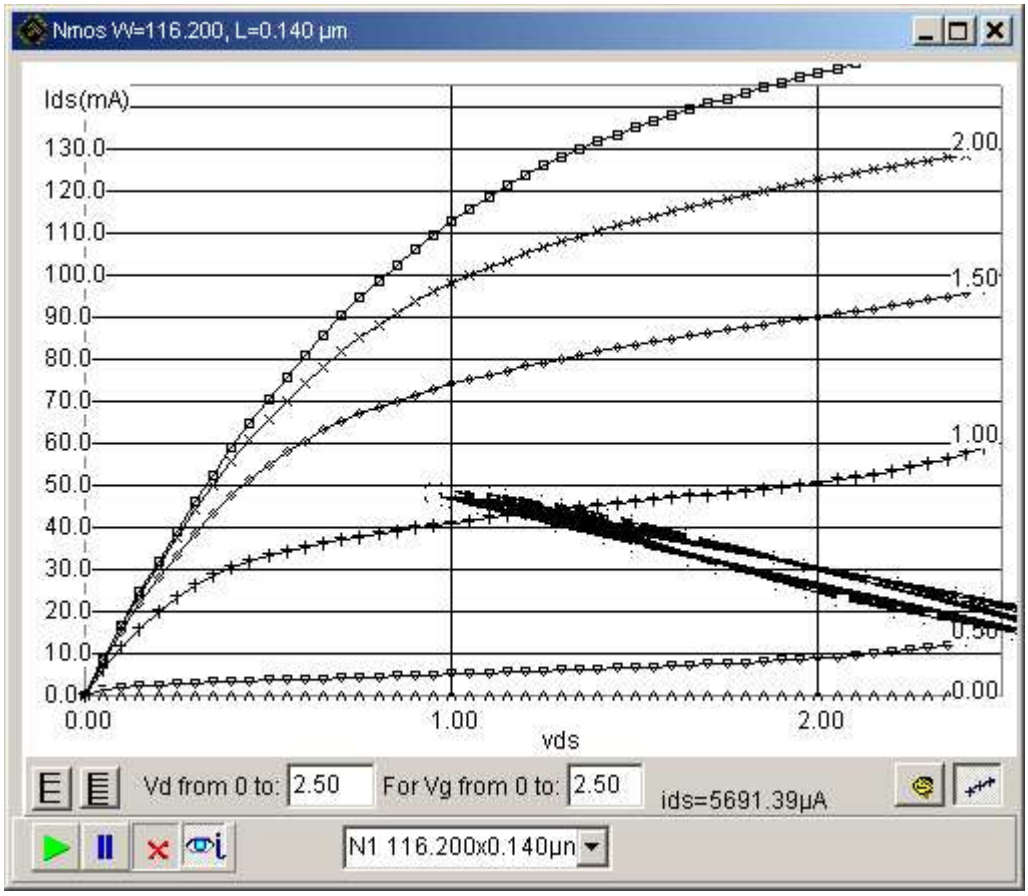


Figure 9-8 : The class A amplifier has a sinusoidal input (PowerAmplifierClassA.MSK)

Oscillator

The role of oscillators is to create a periodic logic or analog signal with a stable and predictable frequency. Oscillators are required to generate the carrying signals for radio frequency transmission, but also for the main clocks of processors. The ring oscillator is a very simple oscillator circuit, based on the switching delay existing between the input and output of an inverter. If we connect an odd chain of inverters, we obtain a natural oscillation, with a period which corresponds roughly to the number of elementary delays per gate. The fastest oscillation is obtained with 3 inverters (One single inverter connected to itself does not oscillate). The usual implementation consists in a series of five up to one hundred chained inverters (Figure 9-9).

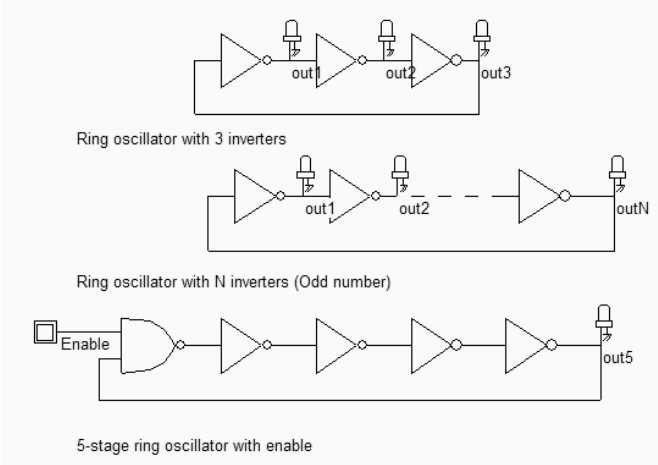


Figure 9-9 : A ring oscillator is based on an odd number of inverters (Inv3.SCH)

The main problem of this type of oscillators is the very strong dependence of the output frequency on virtually all process parameters and operating conditions. This means that any supply fluctuation has a significant impact on the oscillator frequency.

The LC oscillator proposed below is not based on the logic delay, as with the ring oscillator, but on the resonant effect of a passive inductor and capacitor circuit. In the schematic diagram of figure 9-10, the inductor *L1* resonates with the capacitor *C1* connected to S1 combined with *C2* connected to *S2*.

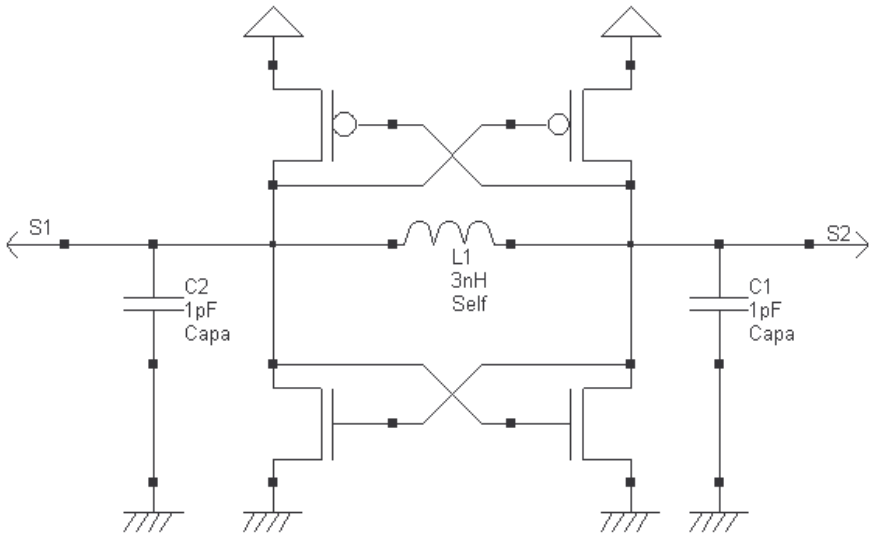


Figure 9-10 : A differential oscillator using an inductor and companion capacitor (OscillatorDiff.SCH)

The layout implementation is performed using a 3nH virtual inductor and two 1pF capacitor. The large width of active devices to ensure a sufficient current to charge and discharge the huge capacitance of the output node at the desired frequency.

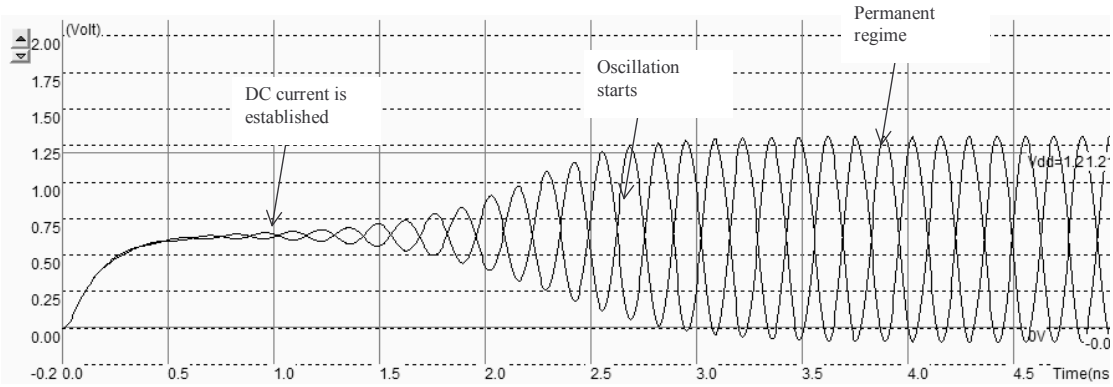


Figure 9-11 : Simulation of the differential oscillator (OscillatorDiff.MSK)

Using virtual capacitors instead of on-chip physical coils is recommended during the development phase. It allows an easy tuning of the inductor and capacitor elements in order to achieve the correct behavior. Once the circuit has been validated, the L and C symbols can be replaced by physical components. The time-domain simulation (Figure 9-11) shows a warm-up period around 1ns where the DC supply rises to its nominal value, and where the oscillator effect reaches a permanent state after some nano-seconds.

The Fourier transform of the output *s1* reveals a main sinusoidal contribution at f0 =3.725 GHz as expected, and some harmonics at *2 x f0* and *3 x f0* (Figure 9-12). The remarkable property of this circuit is its ability to remain in a stable frequency even if we change the supply voltage or the temperature, which features a significant improvement as compared to the ring oscillator. Furthermore, the variations of the MOS model parameters have almost no effect on the frequency.

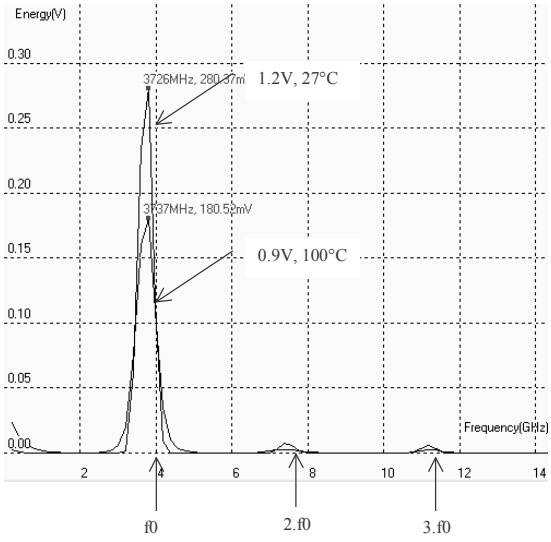


Figure 9-12 : The frequency spectrum of the oscillator (OscillatorDiff.MSK)

Phase-lock-loop

The phase-lock-loop (PLL) is commonly used in microprocessors to generate a clock at high frequency (F_{out}=2 GHz for example) from an external clock at low frequency (F_{ref} = 100 MHz for example). The PLL is also used as a clock recovery circuit to generate a clock signal from a series of bits transmitted in serial without synchronization clock (Figure 9-13).

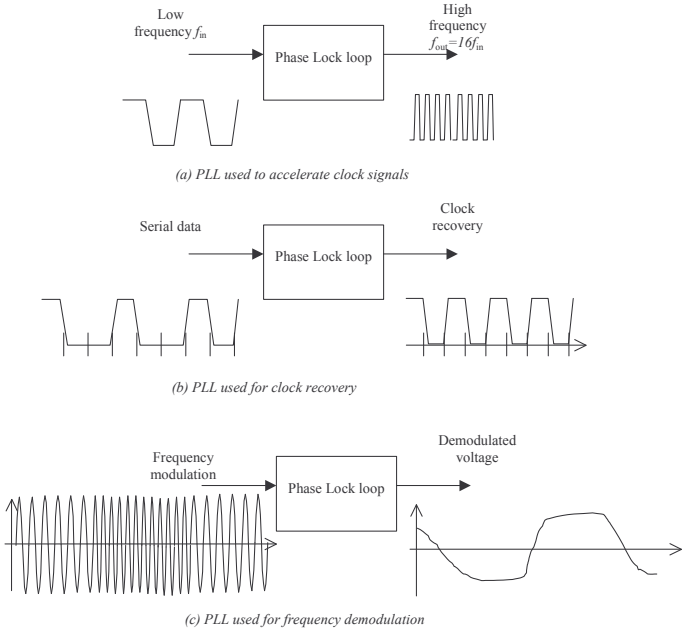


Figure 9-13 : Principles of phase lock loops

The PLL uses a high frequency oscillator with varying speed, a counter, a phase detector and a filter (figure 9-14). The PLL includes a feedback loop which lines up the output clock *ClkOut* with the input clock *ClkIn* through a phase locking stabilization process. When locked, the high input frequency f_{out} is exactly $Nx\,f_{in}$.

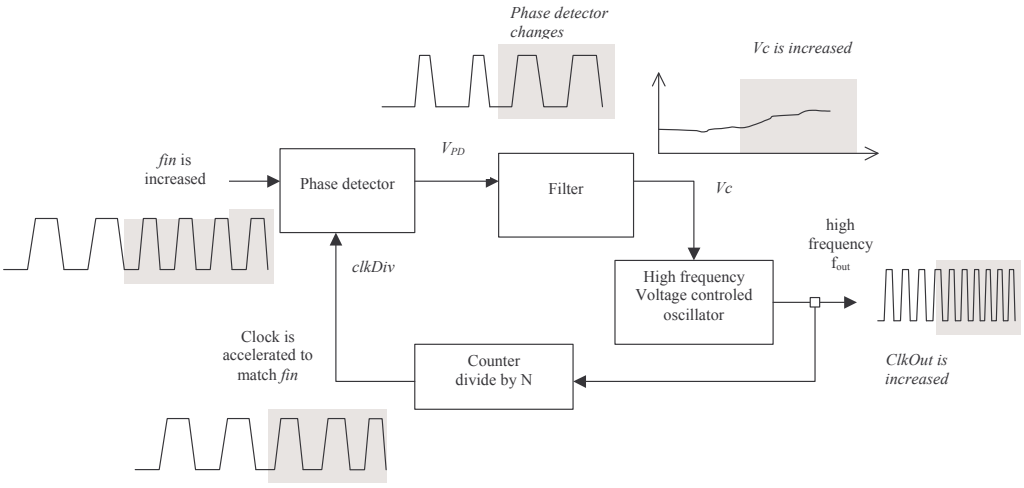


Figure 9-14 : Principles of phase lock loops

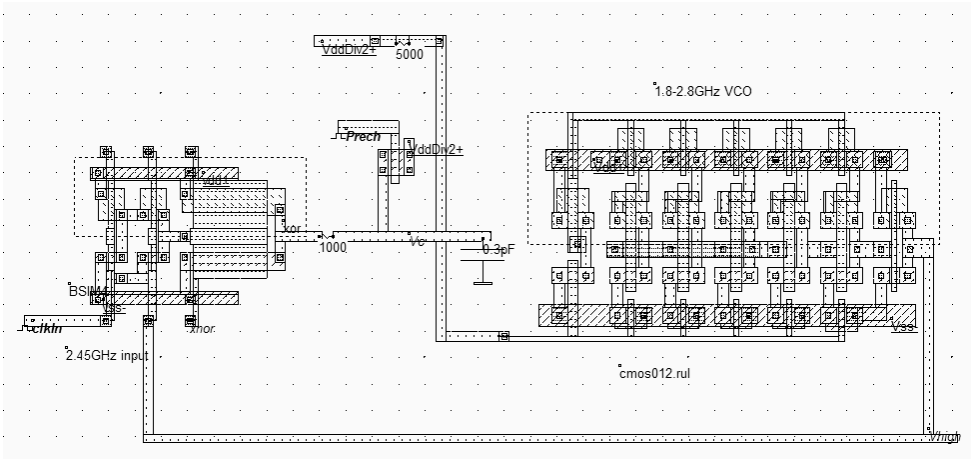


Figure 9-15 : Connecting the current-starved VCO to the phase detector (VCOPll.MSK)

A variation of the input frequency f_{in} is transformed by the phase detector into a pulse signal which is converted in turn into variation of the analog signal *Vc*. This signal changes the VCO frequency which is divided by the counter and changes *clkDiv* according to f_{in}. The implementation of the PLL shown in figure 9-15 includes a resistor *Rfilter* (1000 Ω) and *Rvdd2* (5000 Ω). The capacitor *Cfilter* is a virtual component fixed to 0.3 pF. These resistance and capacitance are easy to integrate on-chip.

The input frequency is fixed to 2.44 GHz. During the initialization phase (Simulation of figure 9-16), the precharge is active, which pushes rapidly the voltage of *Vc* around VDD/2. The VCO oscillation is started and the phase detector starts operating erratically. The output *Xnor* is an interesting indication of what happens inside the phase detector.

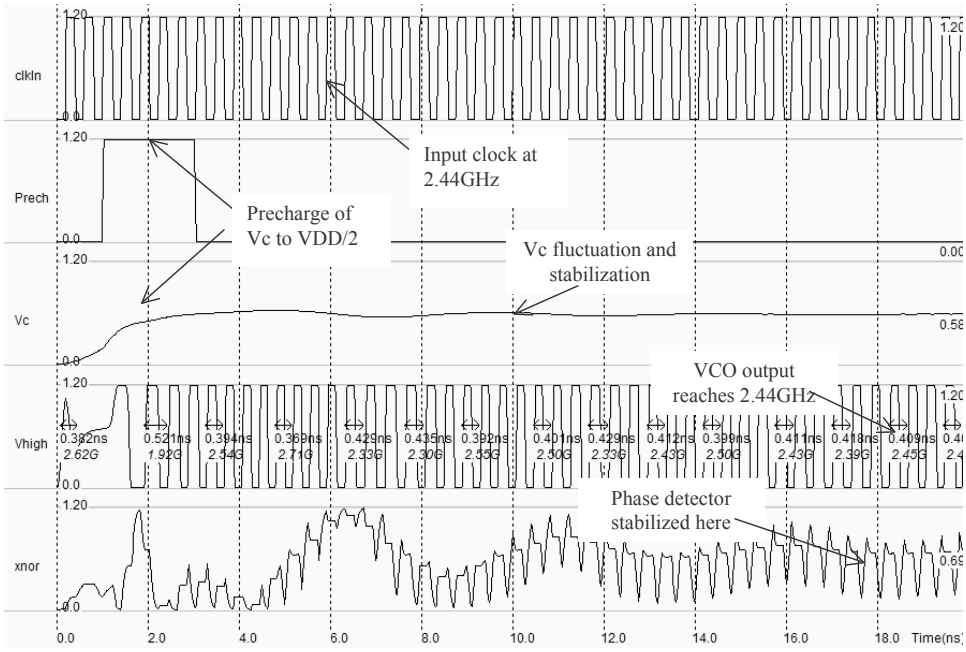


Figure 9-16: Simulation of the PLL showing the locking time (VCOPll.MSK)

We see that the phase difference is very important during the first 10 nanoseconds. Then, the VCO output starts to converge to the reference clock. In terms of voltage control, *Vc* tends to oscillate and then converge to a stable state where the PLL is locked and stable. The output is equal to the input, and the phase difference is equal to one fourth of the period ($\pi/2$) according to the phase detector principles.

Analog to digital and digital to analog converters

The analog to digital converters (ADC) and digital to analog converters (DAC) are the main links between the analog signals and the digital world of signal processing. The ADC and DAC viewed as black boxes are shown in figure 9-16. On the right side, the ADC takes an analog input signal *Vin* and converts it to a digital output signal *A*. The digital signal *A* is a binary coded representation of the analog signal using N bits: A_{N-1} … A_0. The maximum number of codes for N bits is 2^N. The digital signal is usually treated by a microprocessor unit (MPU) or by a specific digital signal processor (DSP) before being restituted as an output *B*. Then, the DAC, which has the opposite function compared to the ADC, converts the digital signal to the final analog output signal *Vout*.

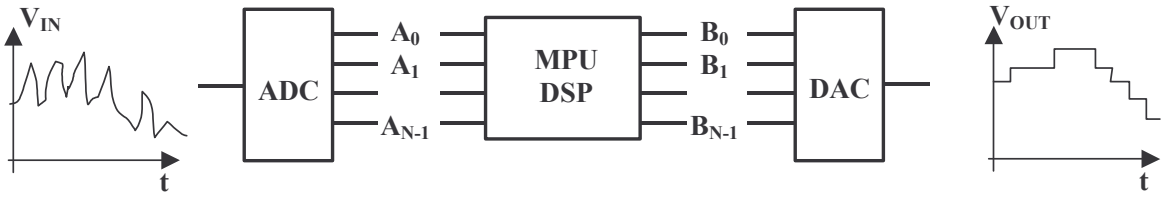


Figure 9-16 : Basic principle of N bits analog to digital and digital to analog converters.

The most basic DAC is based on a resistance ladder. This type of DAC consists of a simple resistor string of 2^N identical resistors, and a binary switch array whose inputs are a binary word. The analog output is the voltage division of the resistors flowing via pass switches (figure 9-17).

In the implementation shown in figure 9-18, the resistance ladder includes 8 identical resistors, which generate 8 reference voltage equally distributed between the ground voltage and *Vdac*. The digital-analog converter uses the three-bit input *B* (*B[2],B[1],B[0]*) to control the transmission gate network which selects one of the voltage references (A portion of *Vdac*) which is then transferred to the output *Vout*. A long path of polysilicon between VDD and VSS may give intermediate voltage references required for the DAC circuit.

B[2]	B[1]	B[0]	Vout*	Analog output Vout* (V) with Vdac=1.2V
0	0	0	0/8 Vdac	0.0
0	0	1	1/8 Vdac	0.15
0	1	0	2/8 Vdac	0.3
0	1	1	3/8 Vdac	0.45
1	0	0	4/8 Vdac	0.6
1	0	1	5/8 Vdac	0.75
1	1	0	6/8 Vdac	0.9
1	1	1	7/8 Vdac	1.05

Figure 9-17 : The specifications of a 3-bit digital-to-analog converter

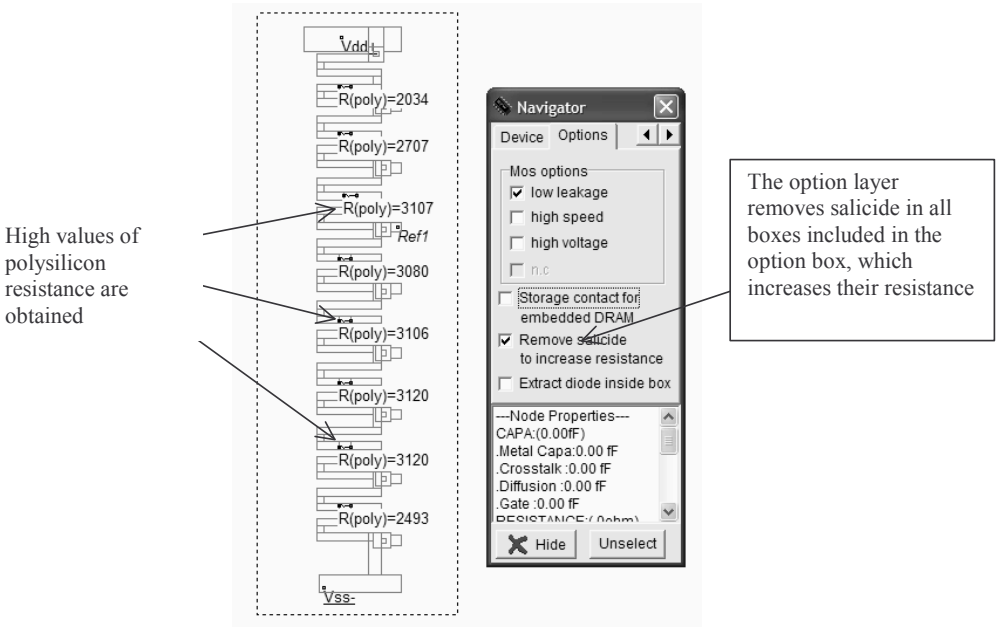


Figure 9-18 : The sheet resistance is increased by removing the salicide deposit, thanks to an option layer (DAC.MSK)

We activate the property "**Remove salicide to increase resistance**" of the option layer (Figure 9-18). Consequently, the resistor value is multiplied by 10 and can be used to design an area-efficient resistor network.

The simulation of the R ladder DAC (Figure 9-19) shows a regular increase of the output voltage *Vout* with the input B[0]..B[2] from “000” (0 V) to “111” (1.0 V). Each input change provokes a capacitance network charge and discharge. Notice the fluctuation of the reference voltage *Vref5* (one of the 8 reference voltages) too. This is due to the weak link to VDD and VSS through a highly resistive path. The analog level *Vout* increases regularly with increasing digit input *B*. The converter is monotonic.

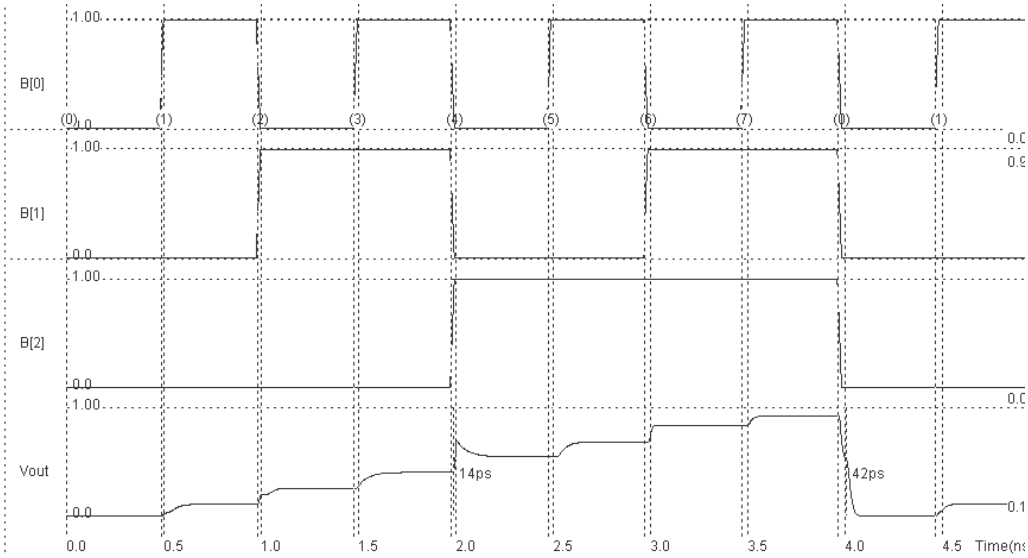


Figure 9-19 : Simulation of the digital-analog converter (DAC.MSK).

The analog to digital converter is considered as an encoding device, where an analog sample is converted into a digital quantity with a number N of bits. ADCs can be implemented by employing a variety of architectures. The 2-bit analog-digital converter converts an analog value *Vin* into a two-bit digital value A coded on 2-bit A_1,A_0. The flash converter uses three amplifiers which produce results C_0,C_1 and C_2, connected to a coding logic to produce A_1 and A_0 in a very short delay (Figure 9-20). The flash converters are widely used for very high sampling rates, a the cost of very important power dissipation.

Analog Input *Vin*	C2	C1	C0	A1	A0
Vin<Vref0	0	0	0	0	0
Vref0<Vin<Vref1	0	0	1	0	1
Vref1<Vin<Vref2	0	1	1	1	0
Vin>Vref2	1	1	1	1	1

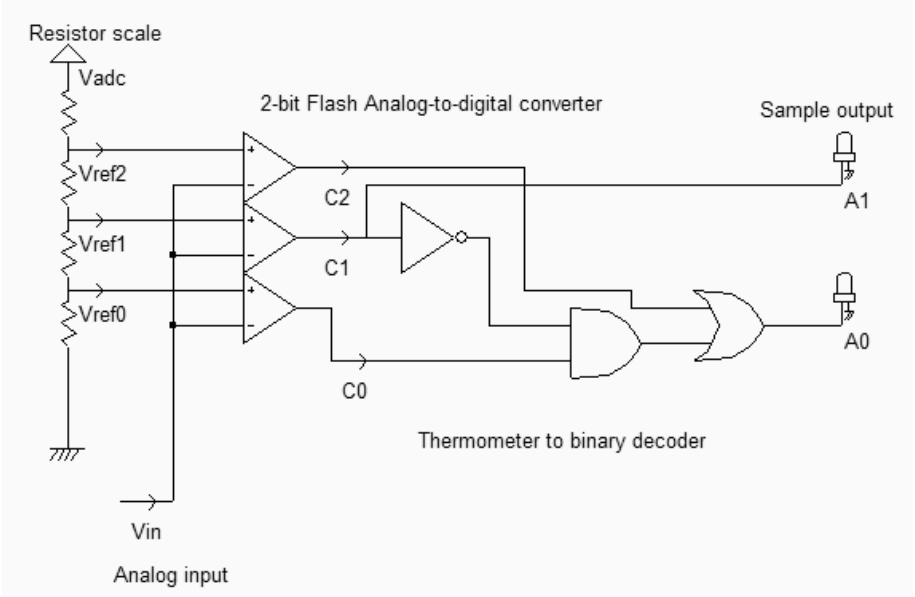


Figure 9-20 : The schematic diagram of the 2-bit flash ADC converter (AdcFlash2bits.SCH)

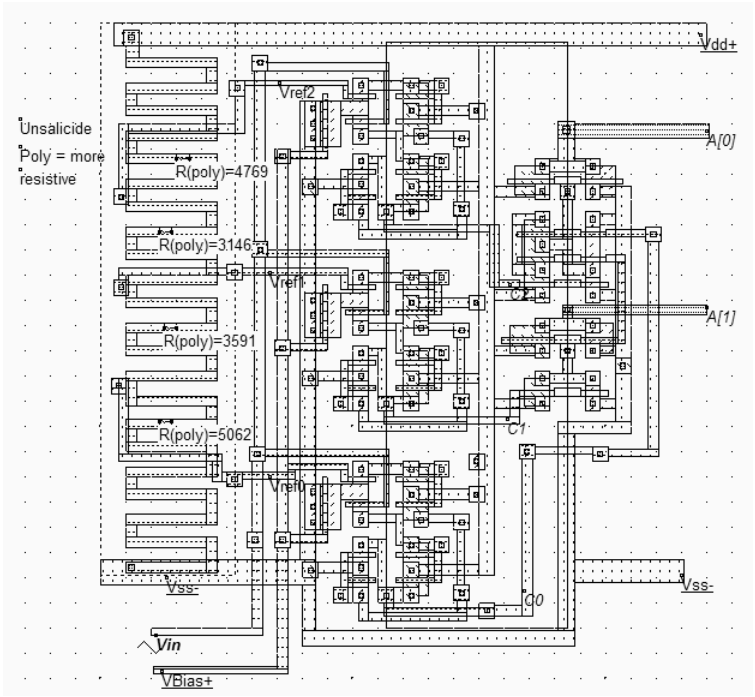


Figure 9-21 : Design of the analog-digital converter (ADC.MSK).

The resistor ladder generates intermediate voltage references used by the voltage comparators located in the middle of the layout. An unsalicide option layer multiplies the sheet resistance of the polysilicon ladder for an area-efficient implementation. The resistance symbol *R(poly)* is inserted in the layout to indicate to the simulator that an equivalent resistance must be taken into account for the analog simulation.

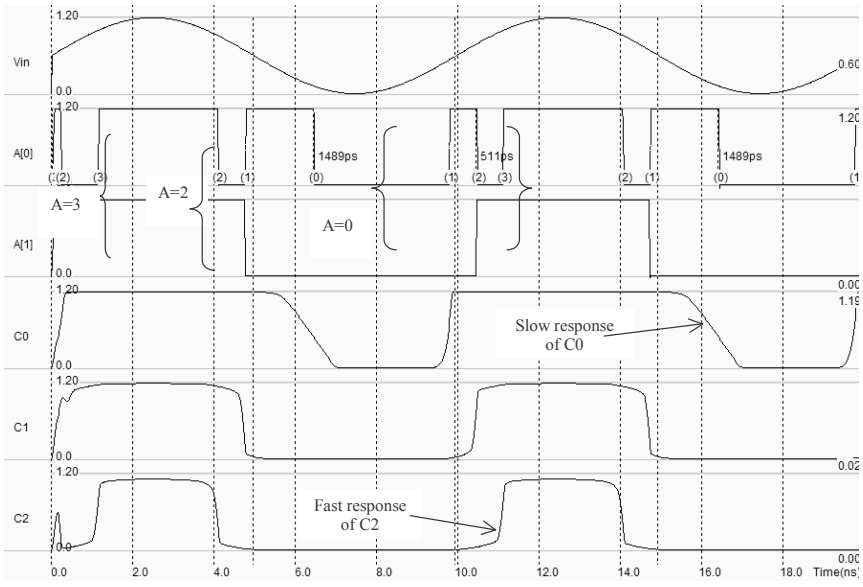


Figure 9-22 : Simulation of the analog-digital converter (ADC.MSK).

Open-loop amplifiers are used as voltage comparators. The comparators address the decoding logic situated to the right and that provides correct A_0 and A_1 coding.
In the simulation shown in figure 9-22, the comparators C_0 and C_1 work well but the comparator C_0 is used in the lower limit of the voltage input range. The generation of combinations "01", "10" and "11" is produced rapidly but the generation of "00" is slow. The comparator C_0 may be modified to provide a faster response in comparison with low voltage, by changing the biasing conditions. An alternative is to reduce the input voltage range, which means that the resistance scale would be supplied by *Vdac*- larger than VSS and *Vdac*+ smaller than VDD.

Added Features in the Full version

Voltage Controlled oscillator	The voltage controlled oscillator (VCO) generates a clock with a controllable frequency. The VCO is commonly used for clock generation in phase lock loop circuits, as described later in this chapter. The clock may vary typically by +/- 50% of its central frequency. A current-starved voltage controlled oscillator is detailed.
Gilbert mixer	The Gilbert mixer is used to shift the frequency of an input signal *Vin* to a high frequency. The Gilbert cell consists of only six transistors, and performs a high quality multiplication of the sinusoidal waves. The schematic diagram and the physical implementation are described in the full version.
Phase-Lock-Loop	Each basic component of the PLL (Phase comparator, filter, VCO) and the design issues are described, supported by a large set of simulations.
Digital to analog converter	The R-2R ladder consists of a network of resistors alternating between *R* and *2R*. For a N bits DAC, only N cells based on 2 resistors *R* and *2R* in series are required. The 4-bit and 8-bit implementation of this circuit are described.
Sample and Hold	The sample-and-hold main function is to capture the signal value at a given instant and hold it until the ADC has processed the information The principles and parasitic effects of the circuit are described.
Analog to digital converter	Successive approach analog to digital converter.

10 Input/Output Interfacing

This chapter is dedicated to the interfacing between the integrated circuit and the external word. After a brief justification of the power supply decrease, the input/output pads used to import and export signals are dealt with. Then, the input pad protections against electrostatic discharge and voltage overstress are described. The design of output buffers is also presented, with focus on current drive.

The Bonding Pad

The bonding pad is the interface between the integrated circuit die and the package. The pad has a very large surface (Almost giant compared to the size of logic cells) because it is the place where the connection wire is attached to build the electrical link to the outside word. The pad is approximately 80µm x 80µm. The basic design rules for the pad are shown in figure 10-1.

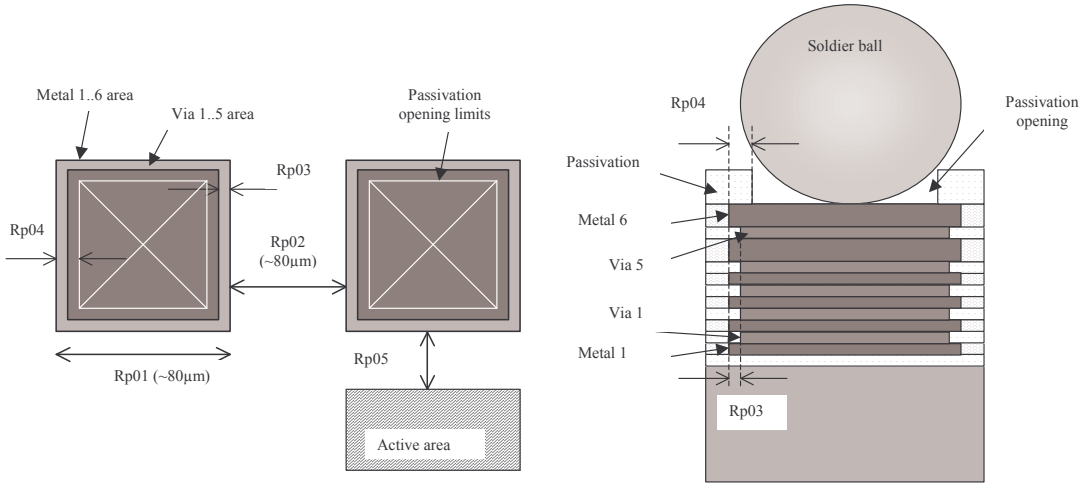


Figure 10-1 : The bonding pad design rules

The cross-section shown in figure 10-2 gives an illustration of the passivation opening and associated design rule *Rp04* on top of the metal and via stack. The thick oxide used for passivation is removed so that a bonding wire or a bonding ball can be connected by melting to the package. The pad can be generated by MICROWIND using the command **Edit → Generate → I/O pads**. The menu gives access to a single pad, with a default size given by the technology (around 80µm in this case), or to a complete pad rind, as detailed later.

The Pad ring

The pad ring consists of several pads on each of the four sides of the integrated circuit, to interface with the outside world. The default menu for an automatic generation of a pad ring is shown in figure 10-2. The proposed architecture is based on 5 pads on each side, meaning a total of 20 pads.

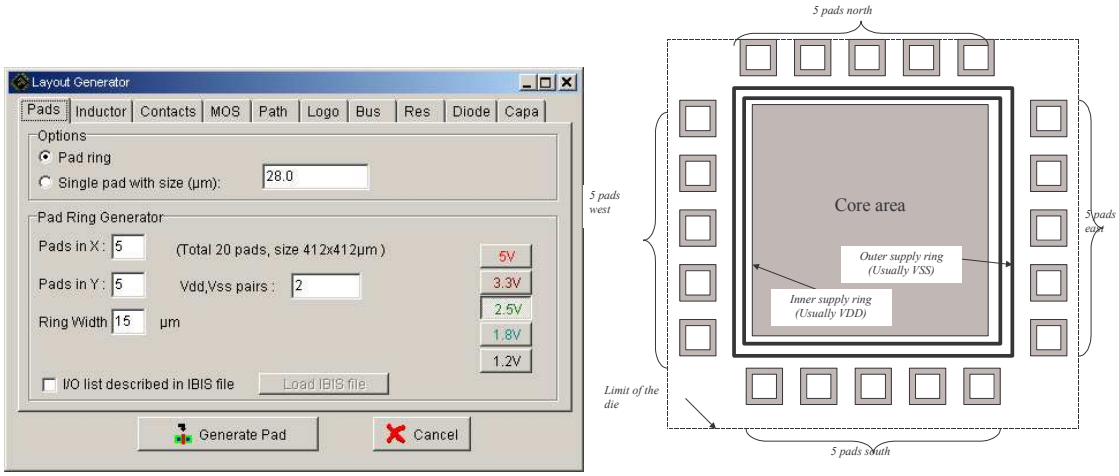


Figure 10-2 : The menu for generating the pad ring and the corresponding architecture

The supply rails

The supply voltage may be 5 V, 3.3 V, 2.5 V, 1.8 V or 1.2 V as listed in the menu shown in figure 10-2. Most designs in 65-nm use 1.0 V for the internal core supply and 2.5V for the interfacing. This is because the logic circuits of the core operate at low voltage to reduce power consumption, and the I/O structures operate at high voltage for external compatibility and higher immunity to external perturbations. Usually, an on-chip voltage regulator converts the high voltage into an internal low voltage.

A metal wire cannot drive an unlimited amount of current. When the average current density is higher than 2.10^9 A/m^2 [Hastings], the grains of the polycrystalline aluminum interconnect start to migrate (The phenomenon is called electro migration) and the conductor ultimately melts. To handle very high current density, the supply metal lines must be enlarged. A typical rule of thumb is 2 mA/µm width for aluminum supply lines and 5 mA/µm for copper, which means that a copper interconnect is superior to aluminum in sustaining large currents.

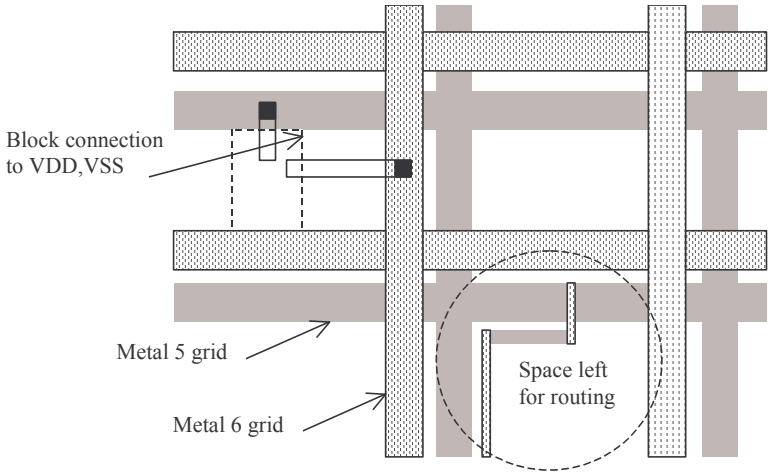


Figure 10-3 : The supply rails are routed in metal5 and metal6 with a regular grid to provide power supply in all regions of the integrated circuit

A complex logic core may consume amperes of current. In that case, the supply lines must be enlarged in order to handle very large currents properly. The usually design approach consists in creating a regular grid structure, as illustrated in figure 10-3, which provides the supply current at all points of the integrated circuit. In that test circuit, the VDD supply is assigned to *metal5*, VSS to *metal6*.

Input Structures

The input pad includes some over-voltage and under-voltage protections due to external voltage stress, electrostatic discharge (ESD) coupling with external electromagnetic sources, etc.. Such protections are required as the oxide of the gate connected to the input can easily be destroyed by over voltage. The electrostatic discharges may attain 1000 to 5000 V.

One of the most simple ESD protections is made up of one resistance and two diodes (Figure 10-4). The resistor helps to dissipate the parasitic energy and reduces the amplitude of the voltage overstress. One diode handles the negative voltage flowing inside the circuit (N+/P substrate diode), the other diode (P+/N well) handles the positive voltage. The combination of the serial resistor and the diode bridge represents an acceptable protection circuit against transient voltage overstress around +/-50 V.

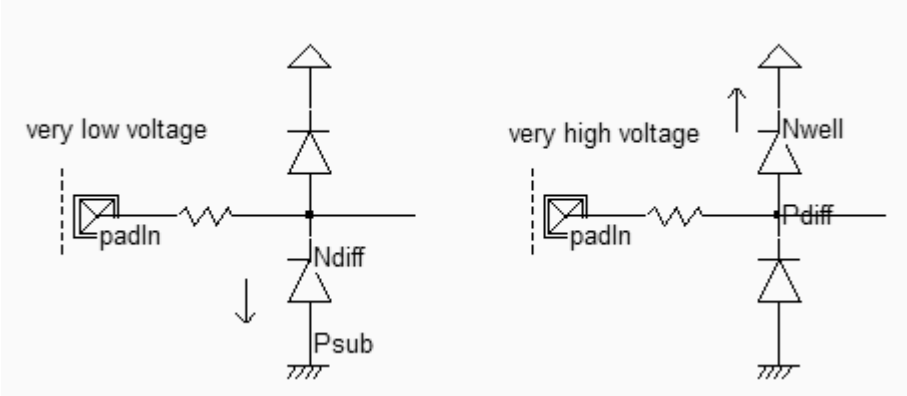


Figure 10-4 : Input protection circuit (IOPadIn.SCH)

Diodes are essential parts of the ESD protection. Used since the infancy stage of microelectronics, the diodes are still widely used because of their efficiency and simplicity [Dabral]. The native diodes in CMOS technology consist of an N+ diffusion in the p-substrate and a P+ diffusion in the n-well.

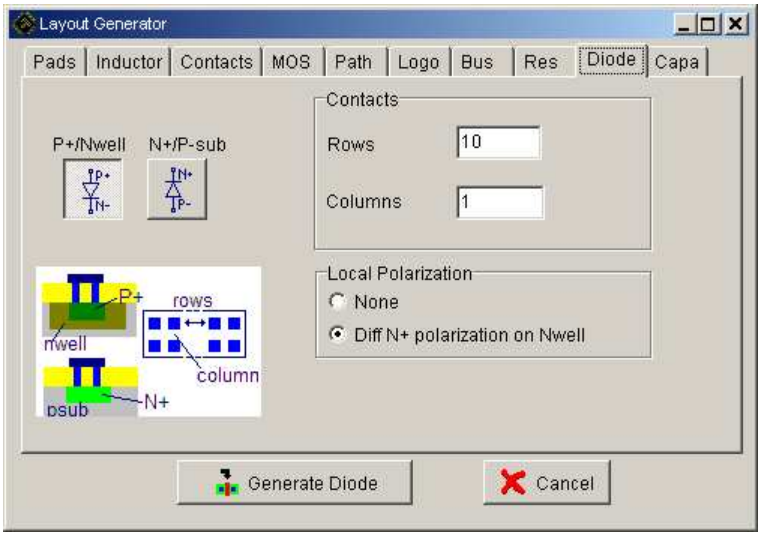


Figure 10-5: The diode generating menu in Microwind (By default a P+/well diode)

The command used to generate a protection diode in MICROWIND is **Edit → Generate → Diode**. Click either the P+/nwell diode or the N+/P substrate diode. By default, the diode is quite large, and connected to the upper metal by a row of 10 contacts. The N+ diode region is surrounded by a polarization ring made of P+ diffusion. The large number of rows ensures a large current capability, which is very important in the case of ESD protection devices.

A protection circuit example is simulated in figure 10-6. It consists of a pad 50 x 50 µm, a serial resistor around 200 Ω and two diodes. When a very high sinusoidal waveform (+/- 10 V) is injected, the diodes exhibit a clamping effect both for the positive and negative overstress.

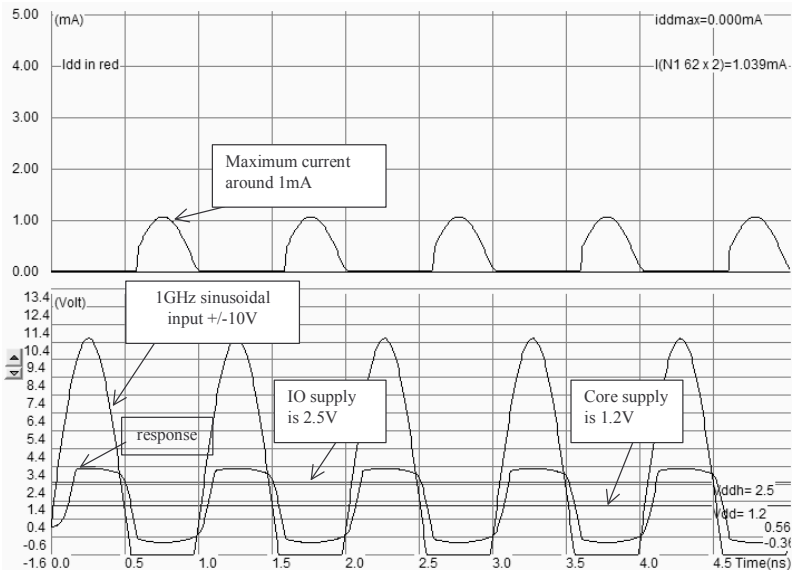


Figure 10-5 : The diodes clamp the positive and negative overstress so that the internal voltage keeps close to the voltage range [0..VDDH] (IoPadIN.MSK)

The best simulation mode is **Voltage and Currents**. The internal voltage remains within the voltage range [0..VDDH] while the voltage near the pad is –10 to +10 V wide. Notice that the current flowing in the diodes is around 1mA (Figure 10-5).

High voltage MOS

The general diagram of an input structure is given in figure 10-6. A high voltage buffer is used to handle voltage overstress issued from electrostatic discharges. The logic signal is then converted into a low voltage signal to be used in the core logic. For interfacing with input/output, specific high voltage MOS are introduced. These MOS devices are called high voltage MOS. They use a double gate oxide to handle the high voltage of the I/Os. The high voltage device symbols are drawn with a double line. The symbol *Vdd_HV* represents the I/O voltage, which is usually 2.5 V in CMOS 65-nm.

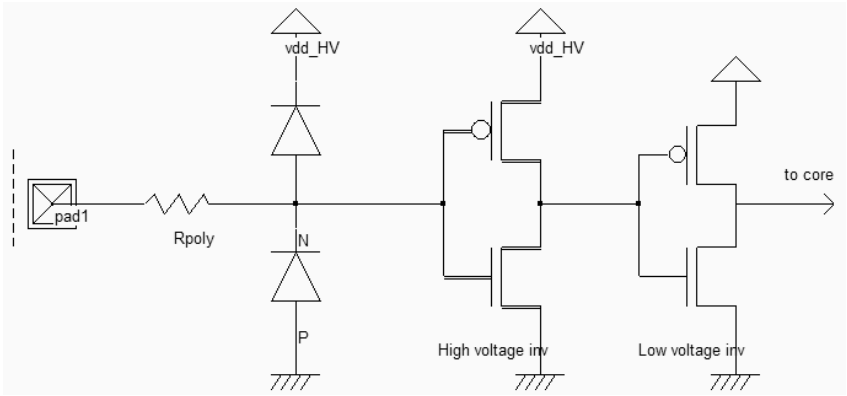


Figure 10-6 : The basic principles for an input circuit, including the ESD protection and the voltage translator (IOPadIn.SCH)

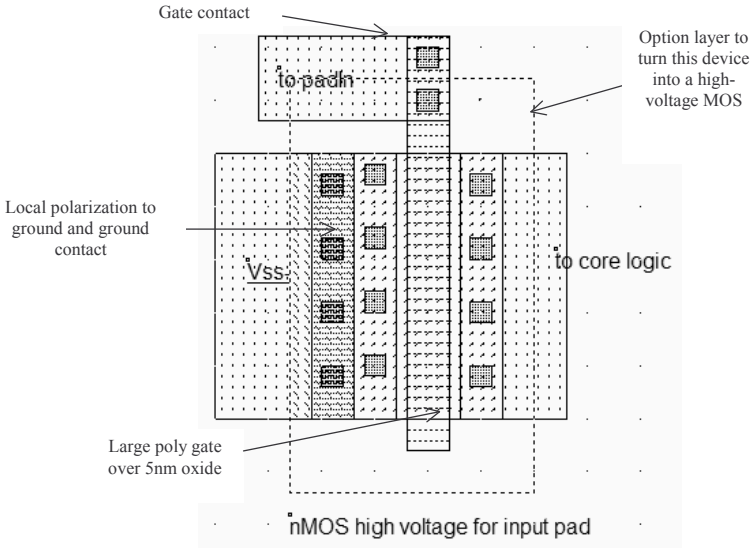


Figure 10-7 : Layout of the input MOS device (IOPadMos.MSK)

The high voltage MOS layout differs slightly from the normal MOS. The high voltage MOS uses a gate width which is much larger than that of the regular MOS. Usually, the lateral drain diffusion, which aims at limiting the hot-carrier effect at boosting the device lifetime, is removed in high voltage MOS devices. In 65-nm, the gate oxide of the high voltage MOS is around 3 nm, while the core MOS is 1.1 nm.

The gate oxide is twice thicker than the low voltage MOS. The high voltage device performance corresponds approximately to a 0.25µm MOS device. To turn a normal MOS into a high voltage MOS, the designer must add an option layer (The dot rectangle in figure 10-7). The tick in front of **High voltage MOS** assigns high voltage properties to the device : double oxide, removed LDD, different rules for minimum length, and different MOS model parameters.

Level shifter

The role of the level shifter is to translate the low voltage logic signal *Data_Out* into a high voltage logic signal which controls the buffer devices. Figure 10-8 gives the schematic diagram of a level shifter circuit which has no problem of parasitic DC power dissipation. The circuit consists of a low voltage inverter, the level shifter itself and the buffer. The circuit has two power supplies: a low voltage *VDD* for the left-most inverter, and a high voltage V*ddHV* for the rest of the circuit.

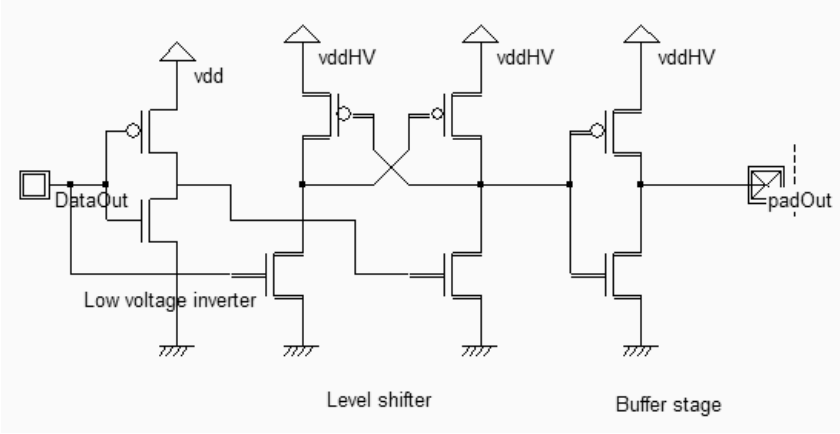


Figure 10-8 : Schematic diagram of a level shifter (IOPadOut.SCH)

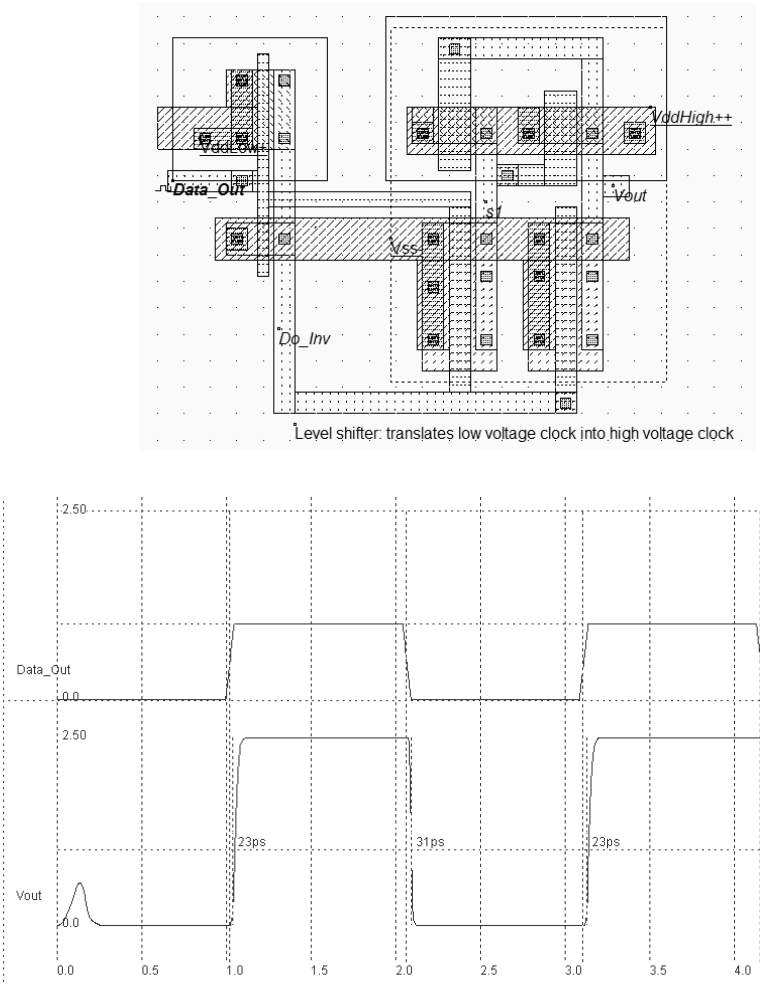


Figure 10-9 : Layout and simulation of the level shifter (LevelShift.MSK)

The layout of the level shifter is shown in figure 10-10. The left part works at low voltage 1.0 V, the right part works with high-voltage MOS devices, at a supply of 2.5V (*VddHigh*). The data signal *Data_Out* has a 0-1.0 V voltage swing.

The output *Vout* has a 0-2.5V voltage swing. This time, no DC consumption appears except during transitions of the logic signals, as shown in the simulation of figure 10-9.

Added Features in the Full version

Pad/Core limitation	When the active area of the chip is the main limiting factor, the pad structure may be designed in such a way that the width is large but the height is as small as possible. This situation, called "Core Limited", as well as its opposite "Pad limited" are detailed.
Schmitt trigger	Using a Schmitt trigger instead of an inverter helps to transform a very noisy input signal into a clean logic signal. The Schmitt trigger circuit switching is illustrated and compared to the normal inverter.
Ibis	IBIS is a standard for electronic behavioral specifications of integrated circuit input/output analog characteristics. MICROWIND uses IBIS to pilot the generation of pads.

11 Design Rules

Select a Design Rule File

The software can handle various technologies. The process parameters are stored in files with the appendix '.RUL'. The default technology corresponds to a generic 8-metal 65-nm CMOS process. The default file is CMOS65n.RUL.

- To select a new foundry, click on **File → Select Foundry** and choose the appropriate technology in the list.
- To set a specific foundry as the default foundry, click **File → Properties** , **'Set as Default Technology'**.
- Click Help → Design Rules to display the design rules (figure 11-1).

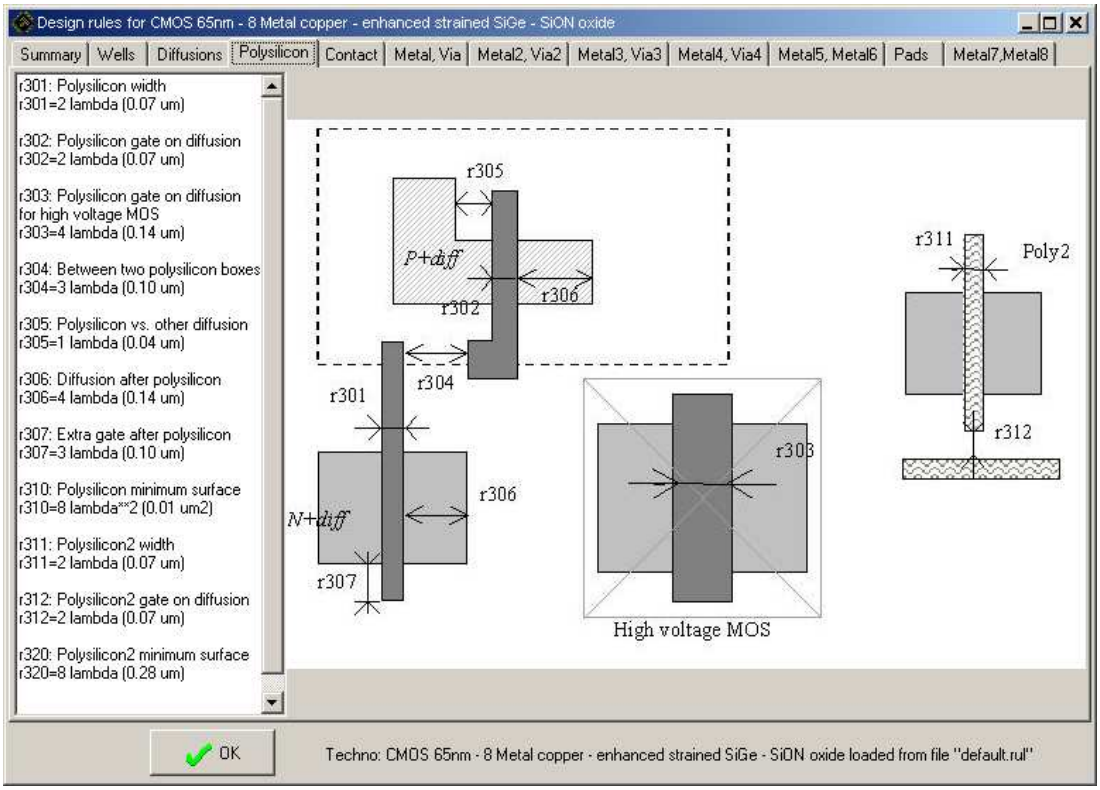


Figure 11-1 : illustration of design rules using the command Help → Design Rules

Lambda Units

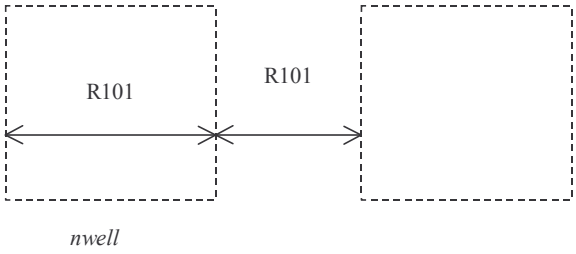
The MICROWIND software works is based on a lambda grid, not on a micro grid. Consequently, the same layout may be simulated in any CMOS technology. The value of lambda is half the minimum polysilicon gate length. Table 11-1 gives the correspondence between lambda and micron for all CMOS technologies available in version 3.1.

Technology file available in version 3.1	Minimum gate length	Value of lambda
Cmos12.rul	1.2µm	0.6µm
Cmos08.rul	0.7µm	0.35µm
Cmos06.rul	0.5µm	0.25µm
Cmos035.rul	0.4µm	0.2µm
Cmos025.rul	0.25µm	0.125µm
Cmos018.rul	0.2µm	0.1µm
Cmos012.rul	0.12µm	0.06µm
soi012.rul (SOI version)	0.12µm	0.06µm
Cmos90n.rul	0.1µm	0.05µm
Cmos65n.rul	0.07µm	0.035µm
Cmos45n.rul	0.05µm	0.025µm

Table 11-1: correspondence between technology and the value of lambda in µm

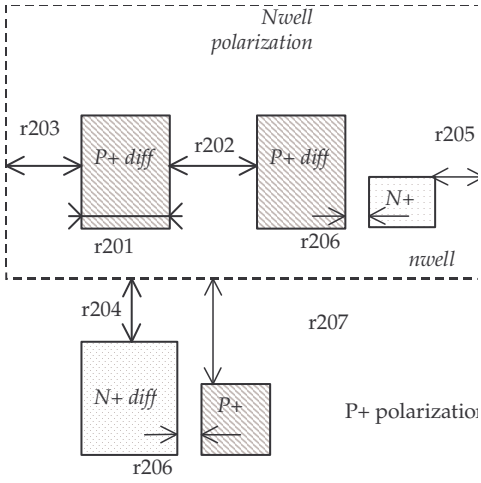
N-Well

r101	Minimum well size	12 λ
r102	Between wells	12 λ
r110	Minimum well area	144 λ^2



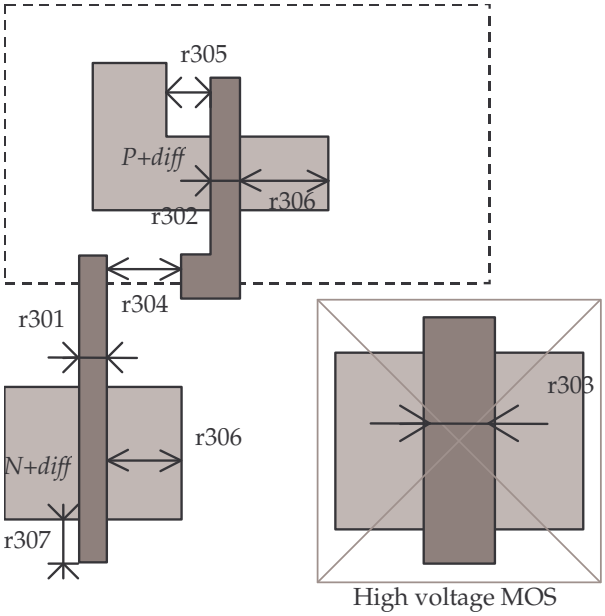
Diffusion

r201	Minimum N+ and P+ diffusion width	4 λ
r202	Between two P+ and N+ diffusions	4 λ
r203	Extra nwell after P+ diffusion :	6 λ
r204:	Between N+ diffusion and nwell	6 λ
r205	Border of well after N+ polarization	2 λ
r206	Between N+ and P+ polarization	0 λ
r207	Border of Nwell for P+ polarization	6 λ
r210	Minimum diffusion area	24 λ^2



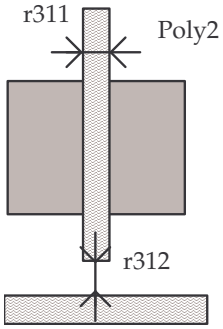
Polysilicon

r301	Polysilicon width	2λ
r302	Polysilicon gate on diffusion	2 λ
r303	Polysilicon gate on diffusion for high voltage MOS	4 λ
r304	Between two polysilicon boxes	3 λ
r305	Polysilicon vs. other diffusion	2 λ
r306	Diffusion after polysilicon	4 λ
r307	Extra gate after polysilicon	3 λ
r310	Minimum surface	8 λ^2



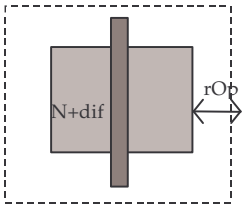
2nd Polysilicon Design Rules

r311	Polysilicon2 width	2 λ
r312	Polysilicon2 gate on diffusion	2 λ
r320	Polysilicon2 minimum surface	8 λ^2



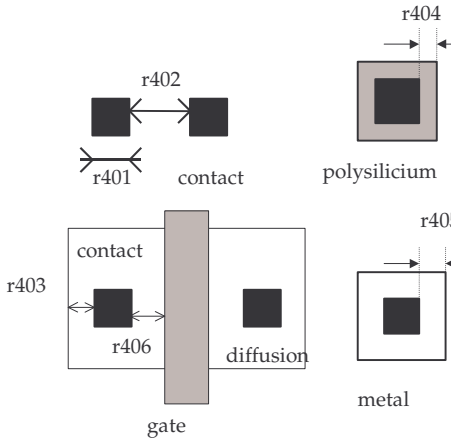
MOS option

rOpt	Border of "option" layer over diff N+ and diff P+	7λ



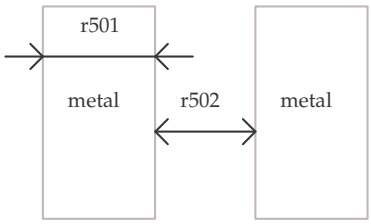
Contact

r401	Contact width	2 λ
r402	Between two contacts	5 λ
r403	Extra diffusion over contact	2 λ
r404	Extra poly over contact	2 λ
r405	Extra metal over contact	2 λ
r406	Distance between contact and poly gate	3 λ
r407	Extra poly2 over contact	2 λ



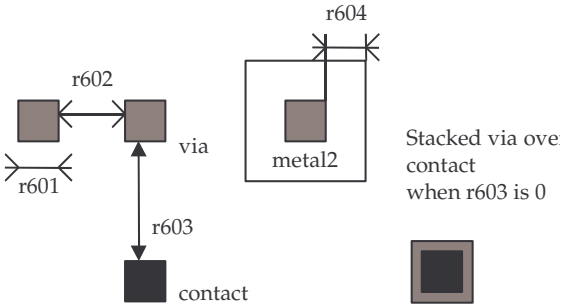
Metal 1

r501	Metal width	4 λ
r502	Between two metals	4 λ
r510	Minimum surface	16 λ^2



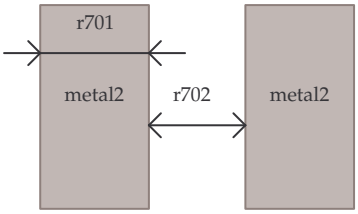
Via

r601	Via width	2 λ
r602	Between two Via	5 λ
r603	Between Via and contact	0 λ
r604	Extra metal over via	2 λ
r605	Extra metal2 over via:	2 λ



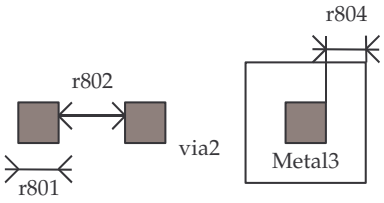
Metal 2

r701	Metal width::	4 λ
r702	Between two metal2	4 λ
r710	Minimum surface	16 λ^2



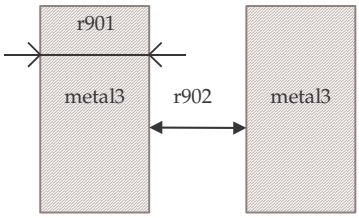
Via 2

r801	Via2 width : 2 λ
r802	Between two Via2: 5 λ
r804	Extra metal2 over via2: 2 λ
r805	Extra metal3 over via2: 2 λ



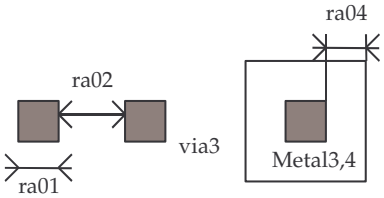
Metal 3

r901	Metal3 width: 4 λ
r902	Between two metal3 : 4 λ
r910	Minimum surface : 32 λ^2



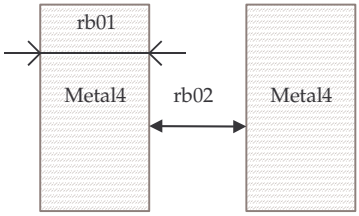
Via 3

ra01	Via3 width : 2 λ
ra02	Between two Via3: 5 λ
ra04	Extra metal3 over via3: 2 λ
ra05	Extra metal4 over via3: 2 λ



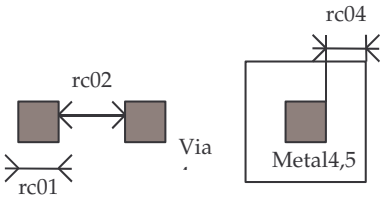
Metal 4

rb01	Metal4 width: 4 λ
rb02	Between two metal4 : 4 λ
rb10	Minimum surface : 32 λ^2



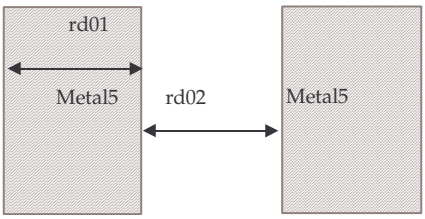
Via 4

rc01	Via4 width : 2 λ
rc02	Between two Via4: 5 λ
rc04	Extra metal4 over via2: 3 λ
rc05	Extra metal5 over via2: 3 λ



Metal 5

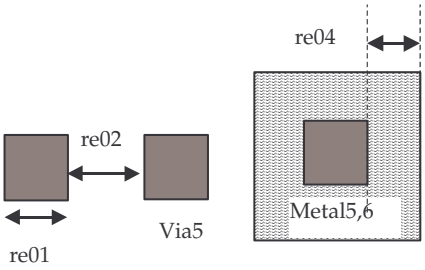
rd01	Metal5 width: 8 λ
rd02	Between two metal5 : 8 λ
rd10	Minimum surface : 100 λ^2



Via 5

re01
re02
re04
re05

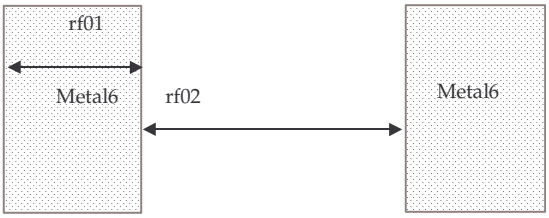
Via5 width : 4 λ
Between two Via5: 6 λ
Extra metal5 over via5: 3 λ
Extra metal6 over via5: 3 λ



Metal 6

rf01
rf02
rf10

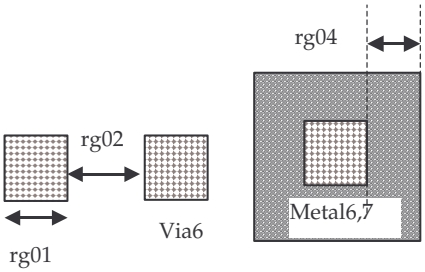
Metal6 width: 8 λ
Between two metal6 : 15 λ
Minimum surface : 300 λ^2



Via 6

rg01
rg02
rg04
rg05

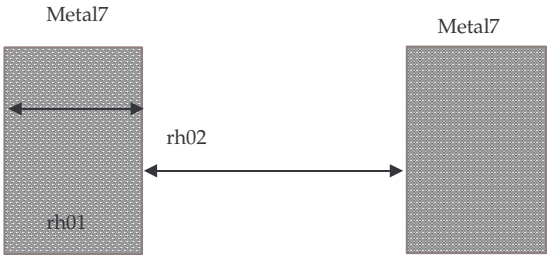
Via6 width : 4 λ
Between two Via6: 6 λ
Extra metal6 over via6: 3 λ
Extra metal7 over via6: 3 λ



Metal 7

rh01
rh02
rh10

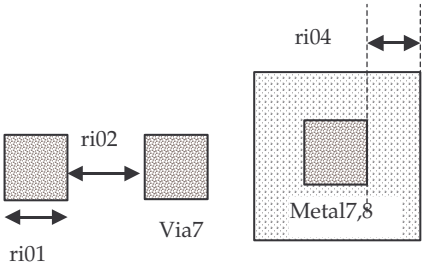
Metal7 width: 8 λ
Between two metal7 : 15 λ
Minimum surface : 300 λ^2



Via 7

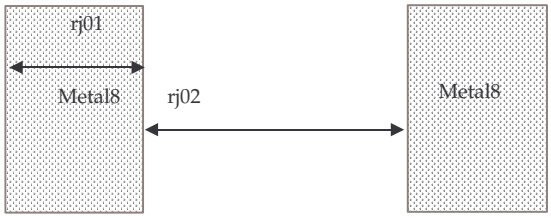
ri01
ri02
ri04
ri05

Via7 width : 4 λ
Between two Via7: 6 λ
Extra metal7 over via7: 3 λ
Extra metal8 over via7: 3 λ



Metal 8

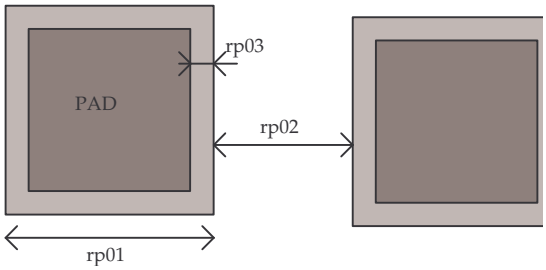
rj01	Metal8 width: 8 λ
rj02	Between two metal8 : 15 λ
rj10	Minimum surface : 300 λ^2



Pads

The rules are presented below in µm. In .RUL files, the rules are given in lambda. As the pad size has an almost constant value in µm, each technology gives its own value in λ.

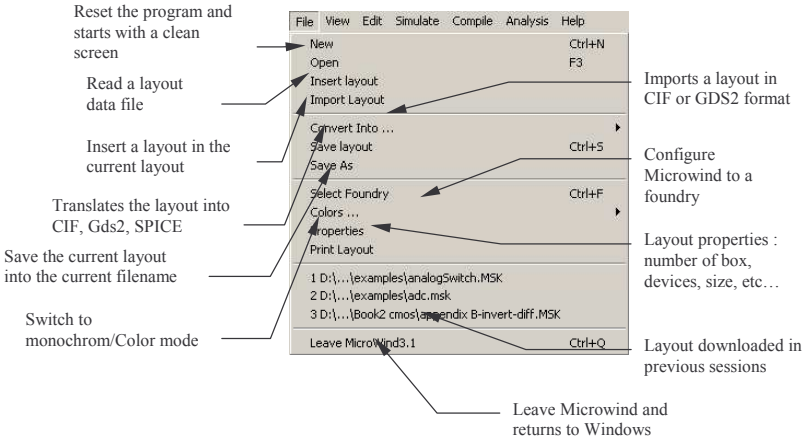
rp01	Pad width:	80 µm
rp02	Between two pads	80 µm
rp03	Opening in passivation v.s via :	5µm
rp04	Opening in passivation v.s metals:	5µm
rp05	Between pad and unrelated active area :	20 µm



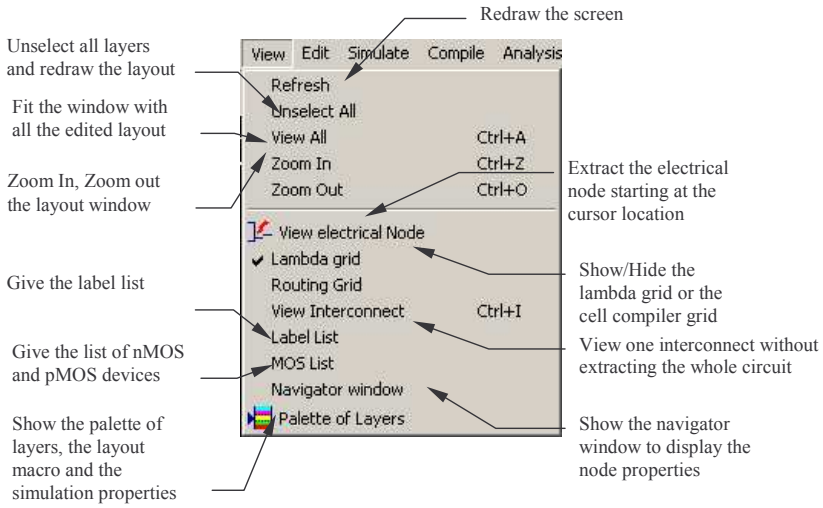
12 MICROWIND and DSCH Menus

Microwind 3.1

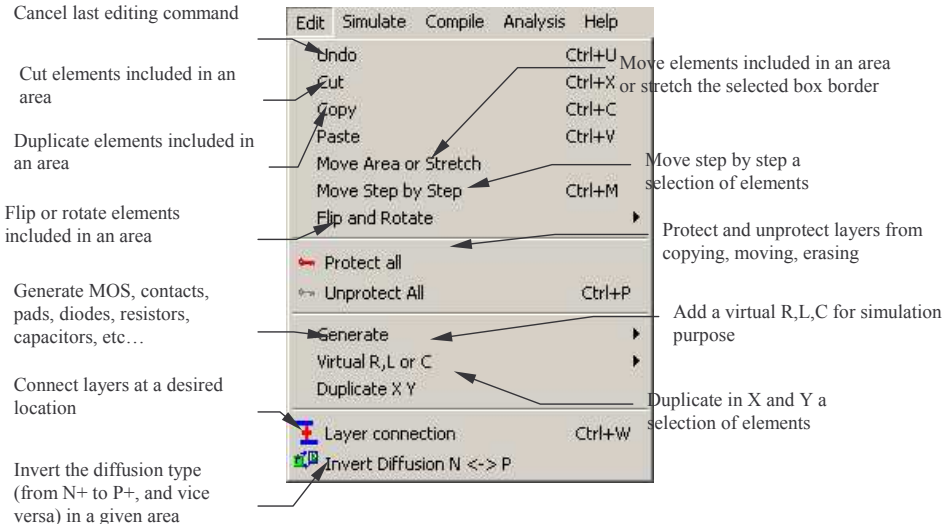
FILE MENU



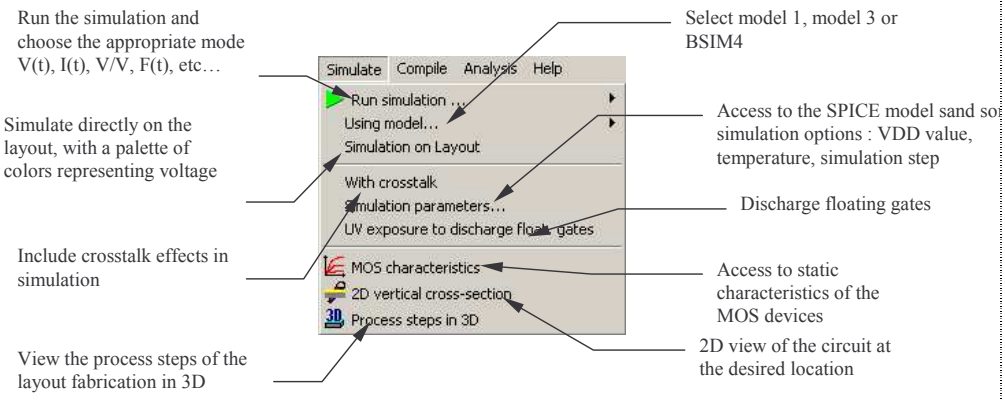
VIEW MENU



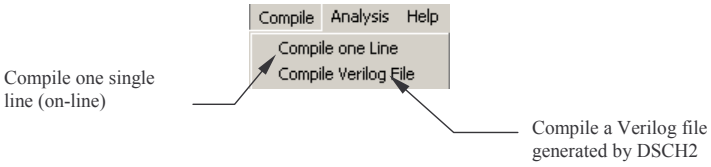
EDIT MENU



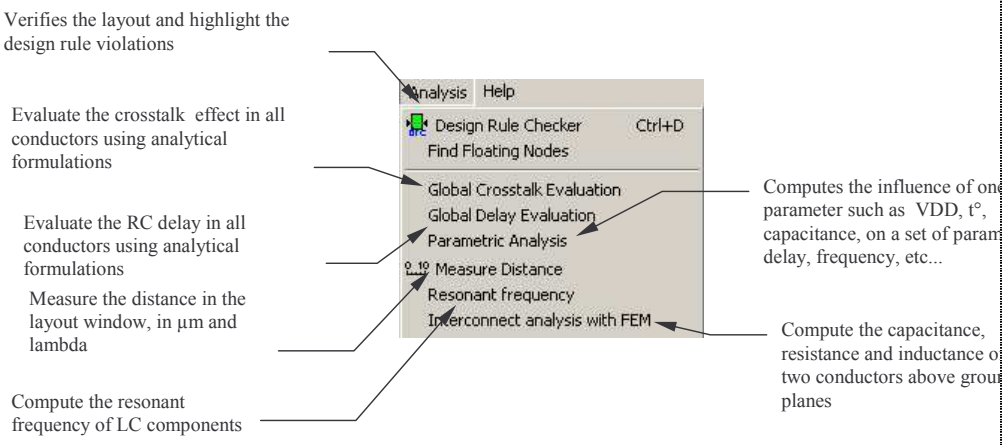
SIMULATE MENU



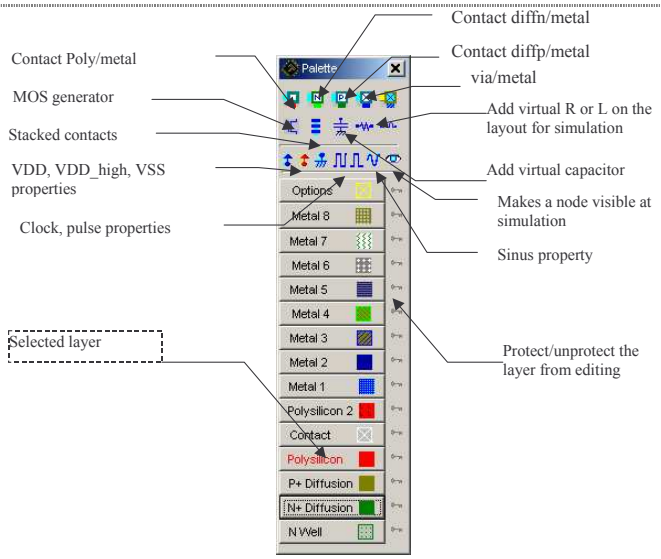
COMPILE MENU



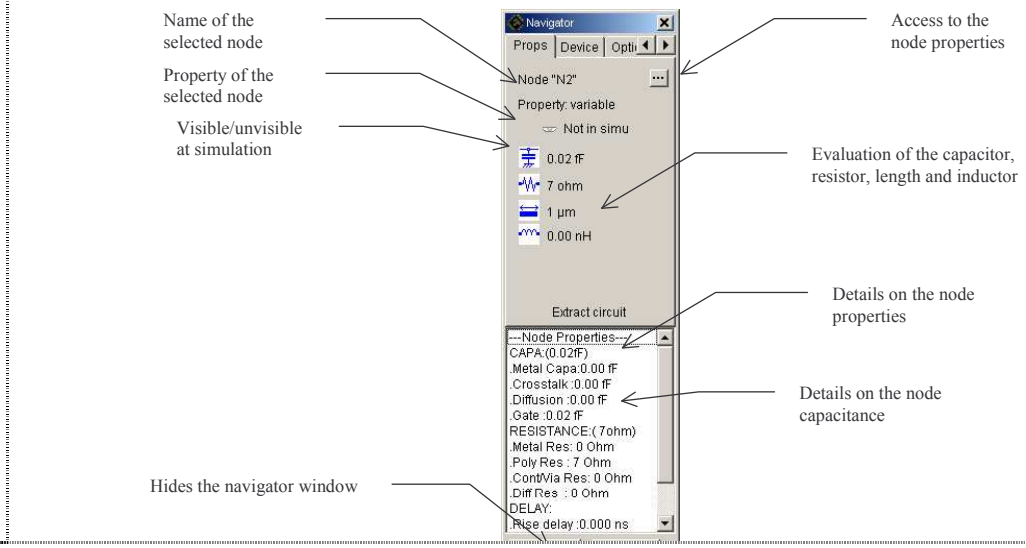
ANALYSIS MENU



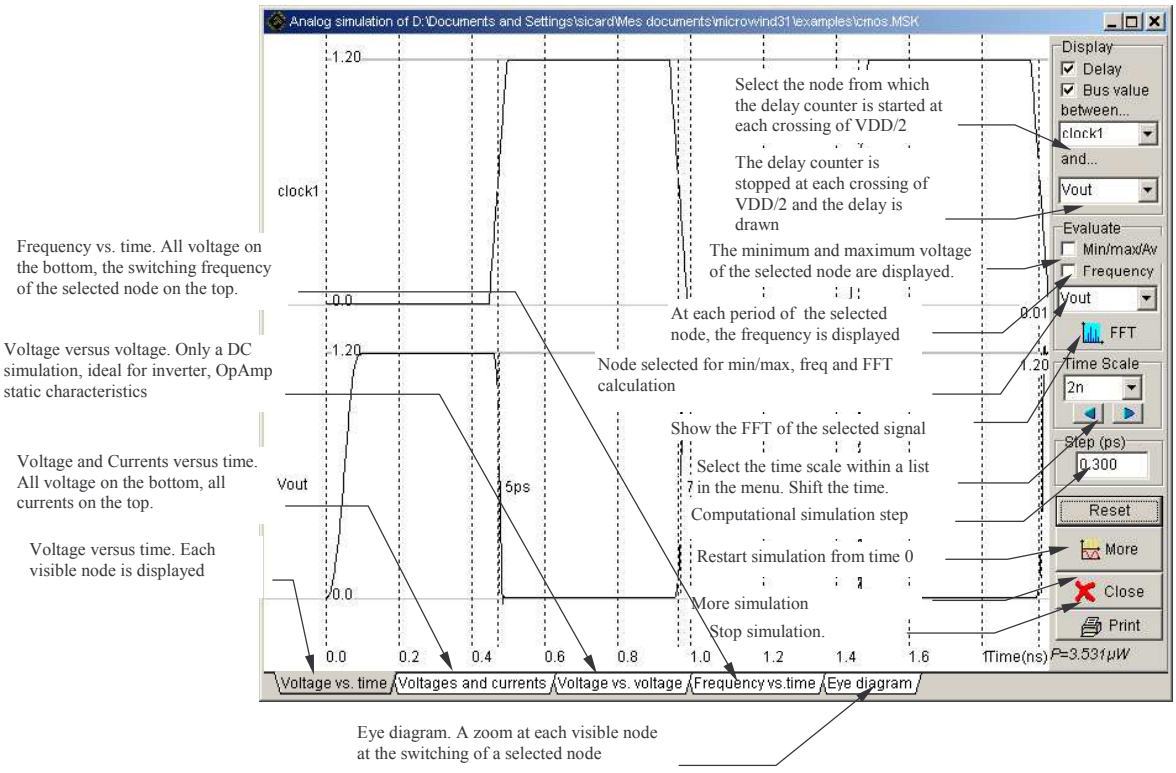
PALETTE



NAVIGATOR WINDOW



Microwind Simulation menu



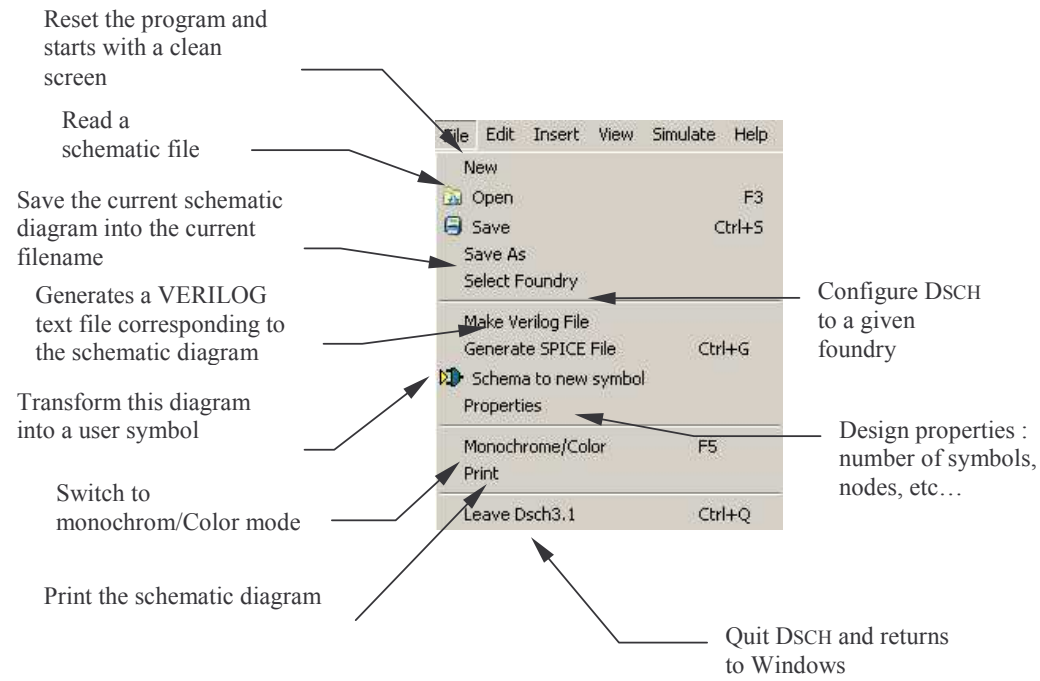
LIST OF ICONS

	Open a layout file (MSK format)		Extract and simulate the circuit
	Save the layout file in MSK format		Measure the distance in lambda and micron between two points
	Draw a box using the selected layer of the palette		2D vertical aspect of the device

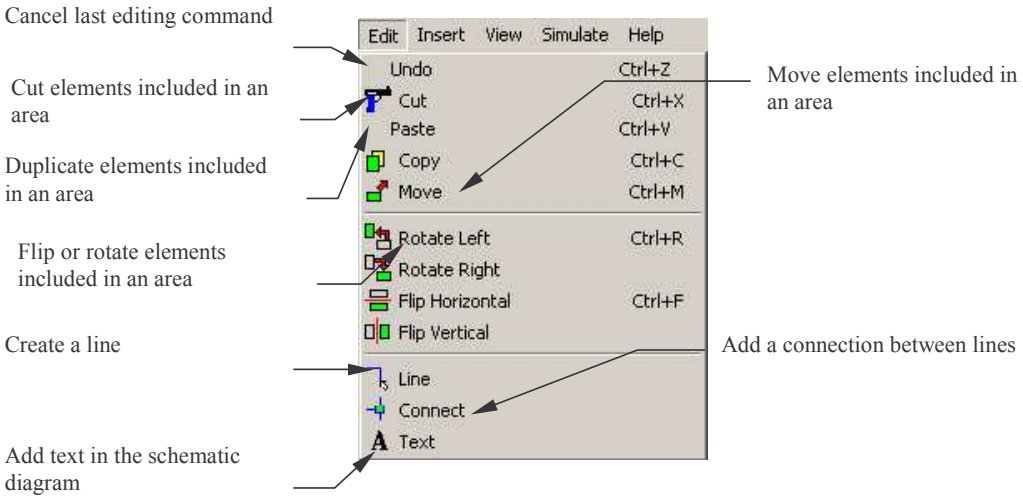
	Delete boxes or text.		Step by step fabrication of the layout in 3D
	Copy boxes or text		Design rule checking of the circuit. Errors are notified in the layout
	Stretch or move elements		Add a text to the layout. The text may include simulation properties.
	Zoom In		Connect the lower to the upper layers at the desired location using appropriate contacts.
	Zoom Out		Static MOS characteristics
	View all the drawing		View the palette
	Extract and view the electrical node pointed by the cursor		Move the layout up, left, right, down

DSCH MENUS

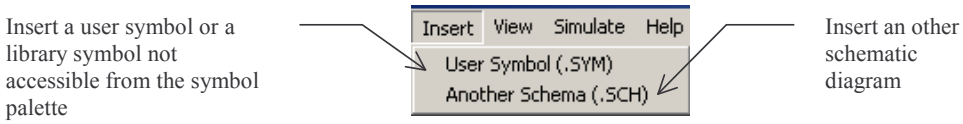
File Menu



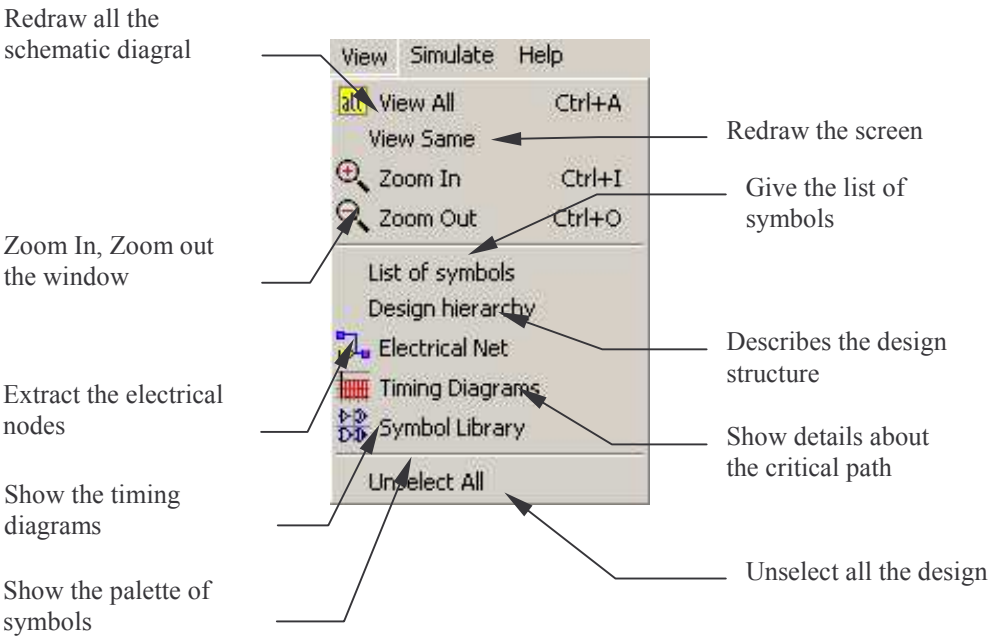
Edit Menu



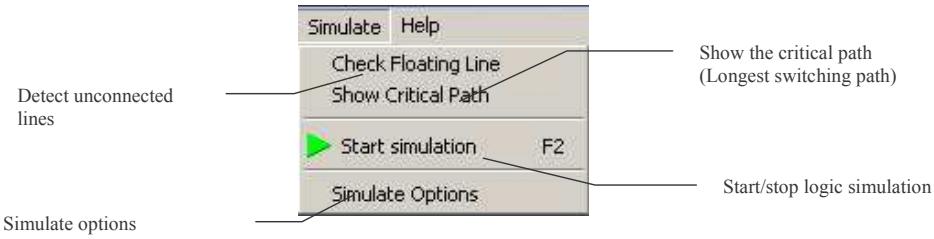
Insert Menu



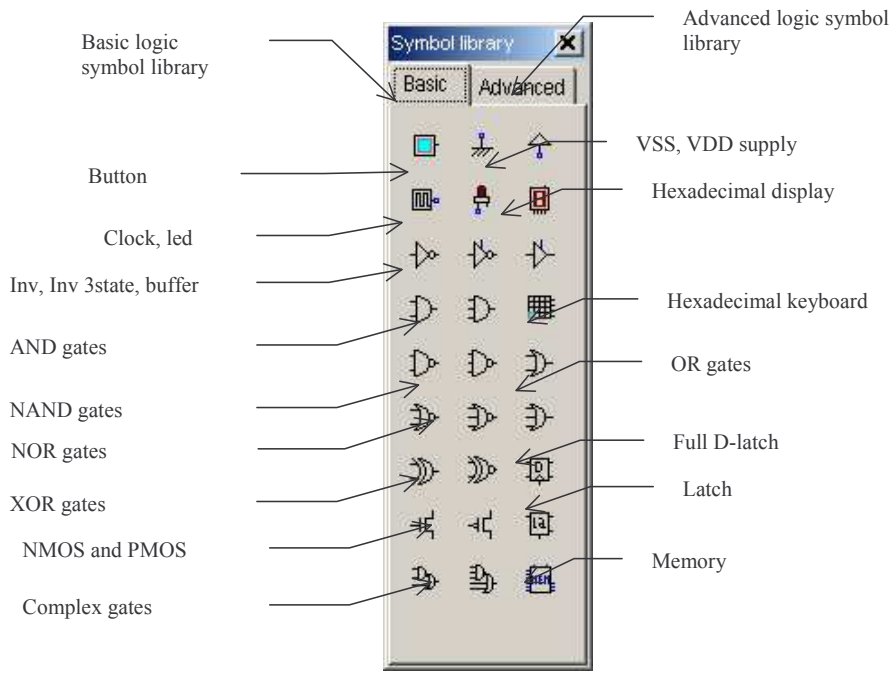
View Menu



Simulate Menu



Symbol Palette



Silicon Menu

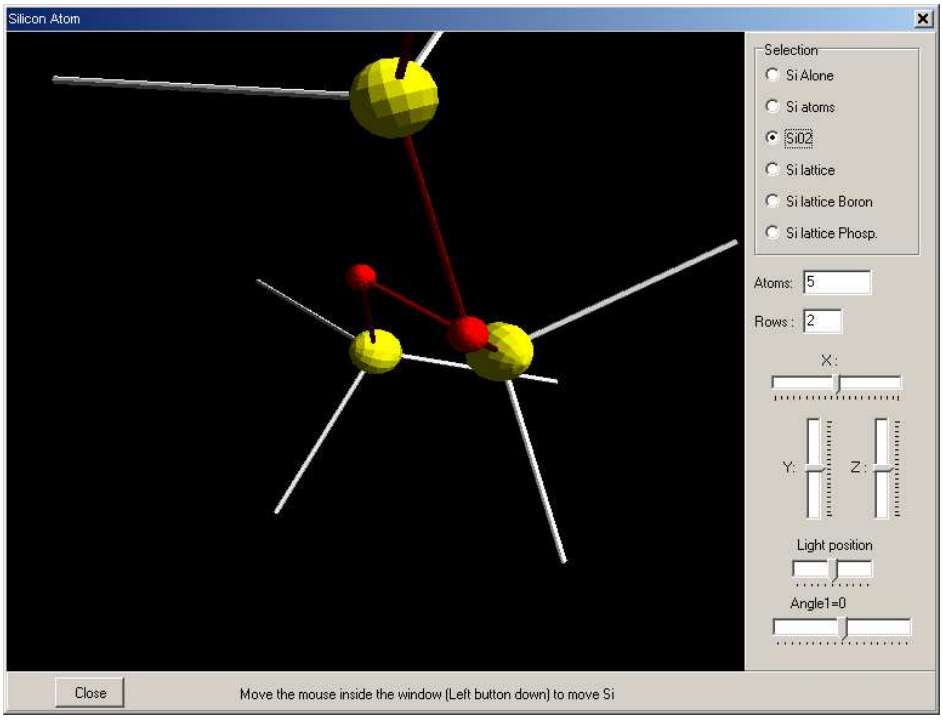


Figure 12-1 : the « silicon » main menu

The software “silicon” is able to give a user’s controlled 3D view of silicon atoms such as SiO2 (figure 12-1). The 3D view of the lattice shown in figure 12-2 shows the regular aspect of Si atoms and the very specific properties of the material. One boron atom acts as a dopant in the structure.

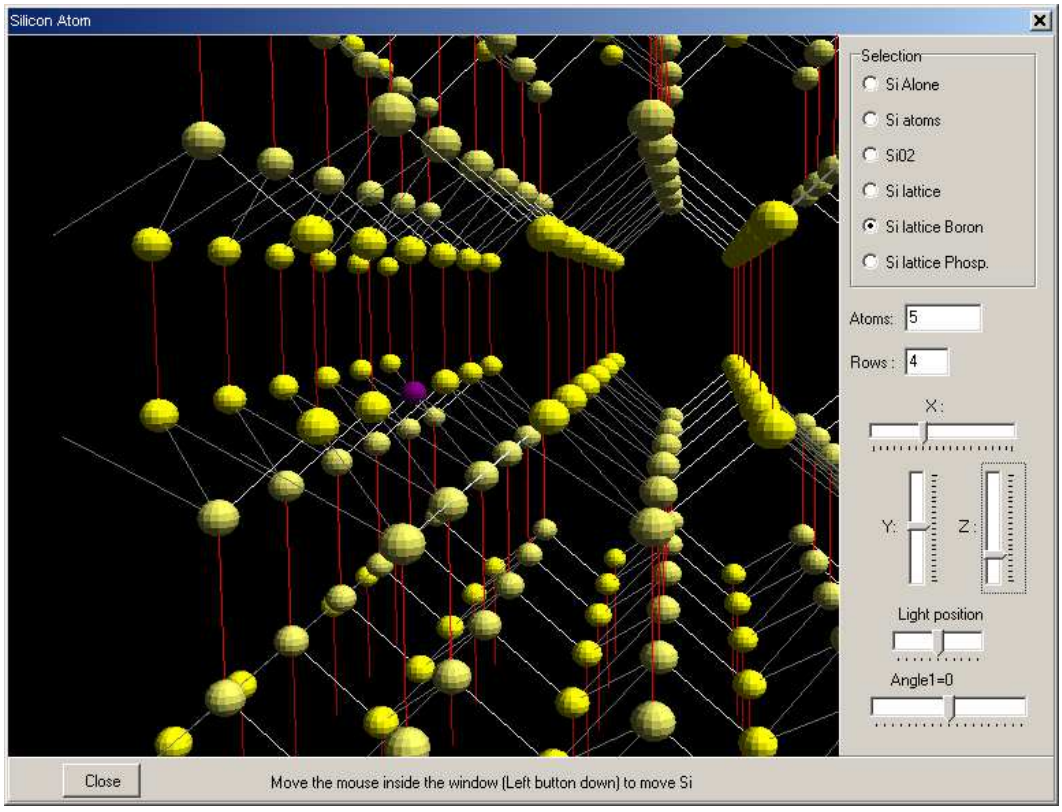


Figure 12-2 : the silicon lattice and a boron dopant

13 References

[Backer] R.J Backer, H. W. Li, D. E. Boyce "CMOS design, layout and simulation", IEEE Press, 1998, www.ieee.org

[Bai] P. Bai et.al., “A 65nm logic technology featuring 35nm gate lengths, enhanced channel strain, 8 Cu interconnect layers, low-k ILD and 0.57 μm^2 SRAM cell”, IEDM 2004, 13-15 Dec. 2004, pp. 657 – 660

[Ghani] T. Ghani et. al. “A 90nm high volume manufacturing logic technology featuring novel 45nm gate length strained silicon CMOS transistors”, Digest of International Electron Devices Meeting 2003, IEDM ’03.

[Hastings] A. Hastings "The Art of Analog Layout", Prentice-Hall, 2001, ISBN 0-13-087061-7

[Lee] K. Lee, M. Shur, T.A Fjeldly, T. Ytterdal "Semiconductor Device Modeling for VLSI", Prentice Hall, 1993, ISBN 0-13-805656-0

[Liu] W. Liu "Mosfet Models for SPICE simulation including Bsim3v3 and BSIM4", Wiley & Sons, 2001, ISBN 0-471-39697-4

[Moore] Moore, G. E., 1965, Cramming more components onto integrated circuits, Electronics, Volume 38, N°8.

[Razavi] B. Razavi "Design of Analog CMOS integrated circuits", McGraw Hill, ISBN 0-07-238032-2, 2001, www.mhhe.com

[Sicard 2005] E. Sicard, S. Ben Dhia “Basic CMOS cell design”, Tata McGraw Hill, 2005, ISBN 0-07-059933-5

[Sicard 2006] E. Sicard, M. Aziz “Introducing 65 nm technology in Microwind3”, application note available on www.microwind.org, August 2006.

[Weste] N. Weste, K. Eshraghian "Principles of CMOS VLSI design", Addison Wesley, ISBN 0-201-53376-6, 1993

[Sharma] A.K. Sharma "Semiconductor Memories", IEEE Press, 1996, ISBN 0-7803-1000-4

ISBN 2-87649-xxx-y
Published by INSA Toulouse
135 Av de Rangueil
31077 Toulouse France
August 2006