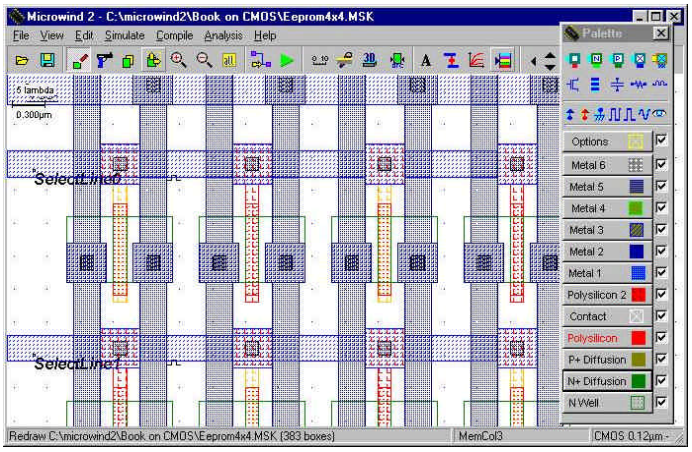
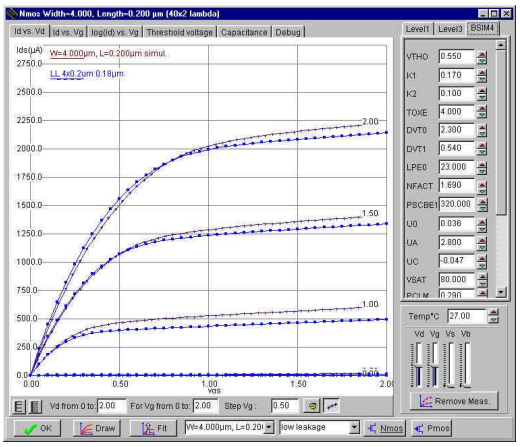
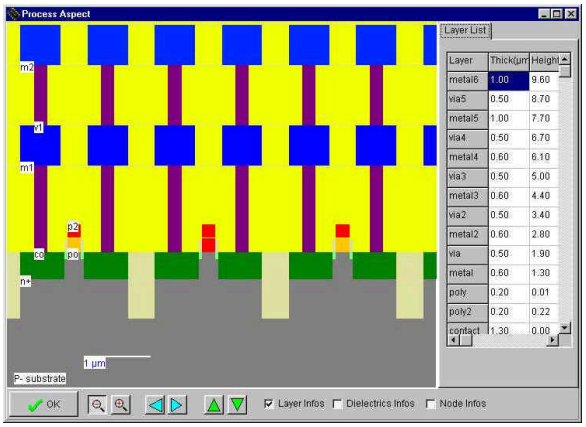
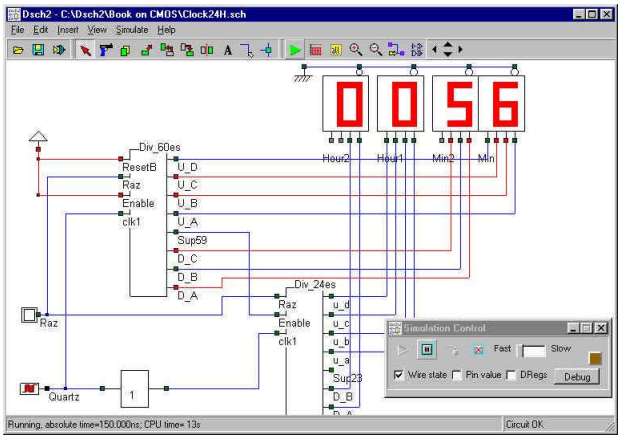
National Institute of Applied Sciences **Toulouse, FRANCE**
Department of Electrical & Computer Engineering

Microwind & Dsch User's Manual Version 2









May 2002

Etienne Sicard
http://intrage.insa-tlse.fr/~etienne

About the author

ETIENNE SICARD was born in Paris, France, in June 1961. He received a B.S degree in 1984 and a PhD in Electrical Engineering in 1987 both from the University of Toulouse. He was granted a Monbusho scholarship and stayed 18 months at the University of Osaka, Japan. Previously a professor of electronics in the department of physics, at the University of Balearic Islands, Spain, E. Sicard is currently an associate professor at the INSA Electronic Engineering School of Toulouse. His research interests include several aspects of design of integrated circuits including crosstalk fault tolerance, and electromagnetic compatibility of integrated circuits. Etienne SICARD is the author of several educational software in the field of microelectronics and sound processing.

Copyright

Address

Etienne Sicard
INSA-DGEI
135, Av de Rangueil
31077 TOULOUSE Cedex 4, FRANCE
Tel : +33.561.55.98.42
Fax: +33.561.55.98.00
e-mail: etienne.sicard@insa-tlse.fr

Web information

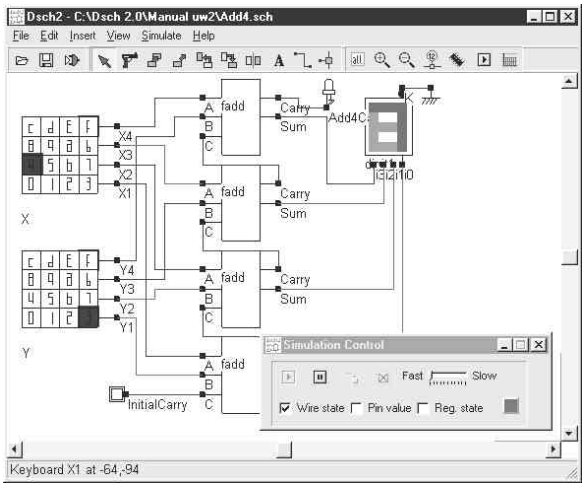
http://intrage.insa-tlse.fr/~etienne

Table of Contents

1 Introduction & Installation

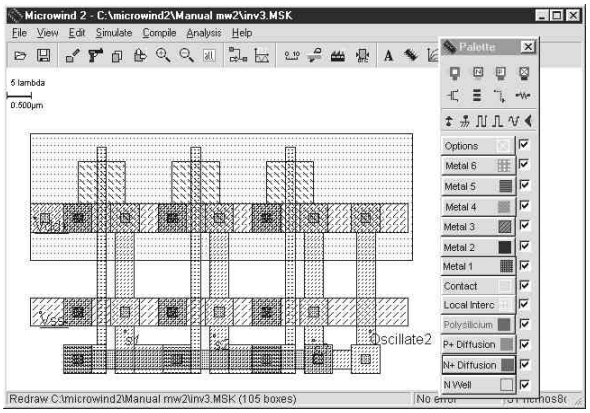
The present manual introduces the design and simulation of CMOS integrated circuits, in an attractive way thanks to user-friendly PC tools Dsch2 and Microwind2.

About Dsch2



The DSCH2 program is a logic editor and simulator. DSCH2 is used to validate the architecture of the logic circuit before the microelectronics design is started. DSCH2 provides a user-friendly environment for hierarchical logic design, and simulation with delay analysis, which allows the design and validation of complex logic structures. A key innovative feature is the possibility to estimate the power consumption of the circuit. Some techniques for low power design are described in the manual.

About Microwind2



The MICROWIND2 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. MICROWIND2 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 describes the technology scale down and the major improvements given by deep sub-micron technologies. Chapter 3 is dedicated to the presentation of the single MOS device, with details on the device modeling, simulation at logic and layout levels. Chapter 4 presents the CMOS Inverter, the 2D and 3D views, the comparative design in micron and deep-submicron technologies. Chapter 5 concerns the basic logic gates (AND, OR, XOR, complex gates), Chapter 6 the arithmetic functions (Adder, comparator, multiplier, ALU). The latches and memories are detailed in Chapter 7.

As for Chapter 8, analog cells are presented, including voltage references, current mirrors, operational amplifiers and phase lock loops. Chapter 9 concerns analog-to-digital and digital to analog converter principles. The input/output interfacing principles are illustrated in Chapter 10.

The detailed explanation of the design rules is in Chapter 11. Electrical rules are described in chapter 12. The program operation and the details of all commands are given in the help files of the programs.

INSTALLATION

From The web

Connect to page http://intrage.insa-tlse.fr/~etienne

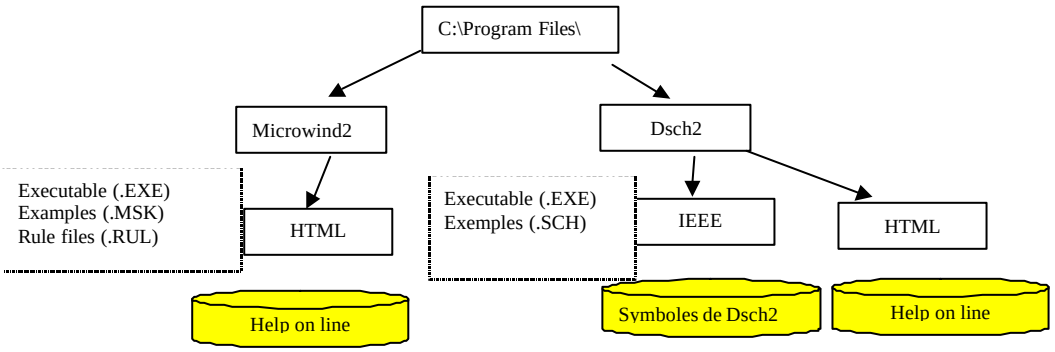
Click "Introduction to microelectronics"

From The CD-ROM

Double-click "index.html"

- Click "Download MICROWIND2 (ZIP file)". In your PC, create manually a directory (Suggested : c:\program files\microwind2). Store the mw.ZIP file in this directory.
- Extract all files with WinZIP in c:\program files\microwind2.
- *Test: double-click MICROWIND2.EXE. Click "File ->Load", select "CMOS.msk". Click "Simulate".*

- Click "Download DSCH2 (ZIP file)". In your PC, create manually a directory (Suggested : c:\program files\dsch2). Store the dsch2.ZIP file in this directory.
- Extract all files with WinZIP in c:\program files\dsch2.
- *Test: double click in DSCH2.EXE. Load "base.sch". Click "Simulate".*



2 Technology Scale Down

The evolution of integrated circuit (IC) fabrication techniques is a unique fact in the history of modern industry. The improvements in terms of speed, density and cost have keep constant for more than 30 years. By the end of 2000, “System-on-Chips” with about 100,000,000 transistors will be fabricated on a single piece of silicon no larger than 2x2 cm. In this chapter, we present some information illustrating the technology scale down.

2.1 Evolution of Microprocessors and Memories

Figure 2-1 describes the evolution of Intel ® microprocessors, figure 2-2 describes the evolution of memory size during the last decades. In figure 3, it is shown that industry has started to produce ICs in deep submicron technology starting 1995. Research has always kept around 5 years ahead mass production.

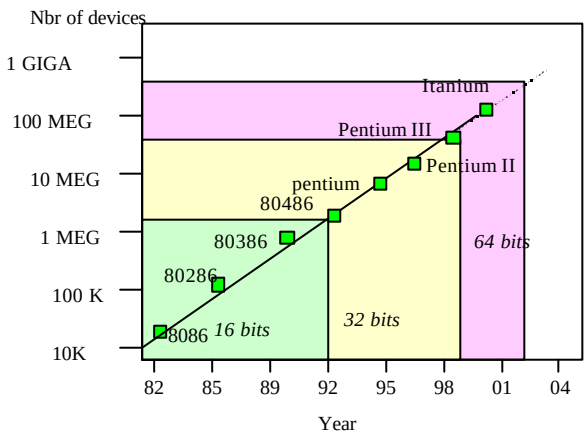


Fig.2-1: Evolution of microprocessors

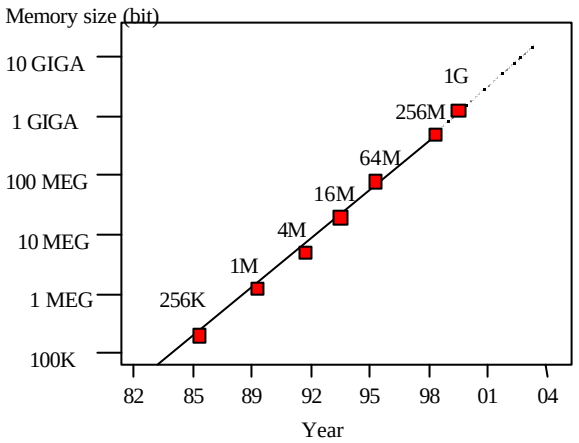


Fig.2-2: Evolution of memories

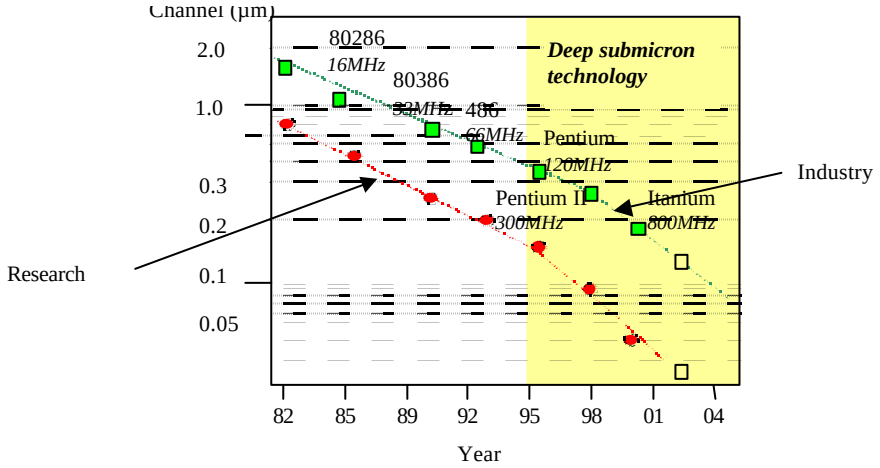


Fig. 2-3: Evolution of lithography

2.2 Frequency Improvements

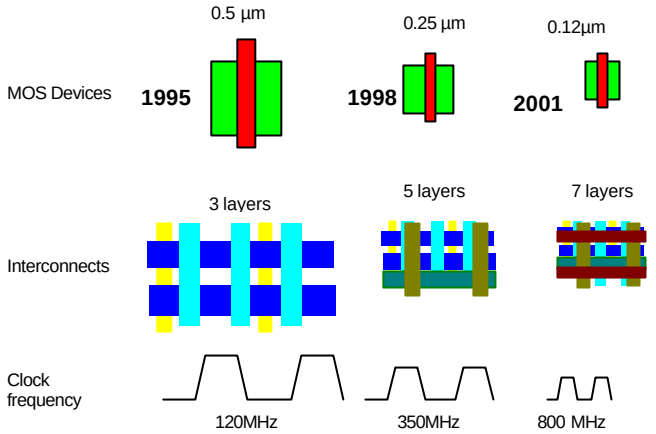


Fig. 2-4: Reduced device features and increased interconnect layers

Figure 2-4 illustrates the main improvements in terms of feature size reduction for MOS devices, increased number of metal interconnects to link MOS together within the chip. Consequently, the clock frequency of the chip has never stopped increasing, with an expected 800MHz in 2001. An illustration is given below (Figure 5), with a ring oscillator made from 3 inverters, simulated with MICROWIND2 using 0.8µm and 0.25µm technologies. Although the supply voltage has been cut by half (VDD is 5V in 0.8µm, 2.5 in 0.25µm), the gain in frequency is close from a factor of five.

0.8 µm

0.25 µm

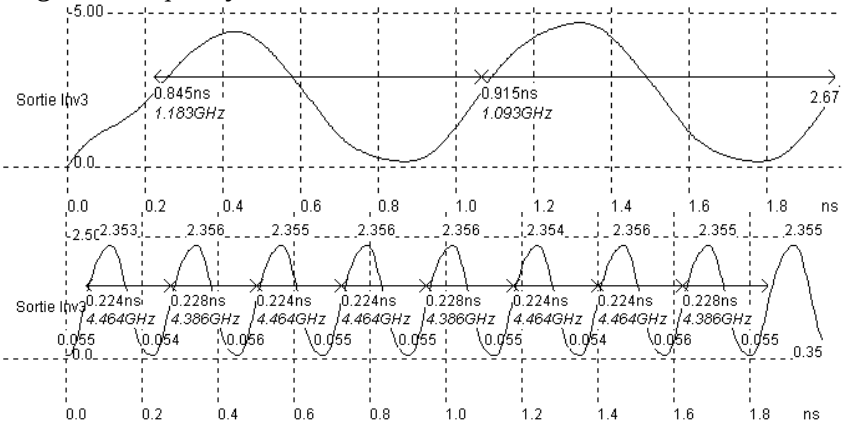


Fig. 2-5: Improvement in speed thanks to deep submicron technology

HOW TO SIMULATE

① Start Microwind2. By default the software is configured with 0.25µm technology. Click "File -> Open".

② Select "INV3". Click "Simulate-> Start Simulation". The oscillation (figure 2-5) appears. Click "Close".

③ Click "File -> Select Foundry". Click "cmos08.rul".

④ Run again the simulation. Observe the change of VDD and the slow down of the oscillating frequency.

2.3 Increased Layers

The table below lists a set of key parameters, and their evolution with the technology. Worth of interest is the increased number of metal interconnects the reduction of the power supply VDD and the reduction of the gate oxide down to atomic scale values. Notice also the slow decrease of the threshold voltage of the MOS device and the increasing number of input/output pads available on a single die.

Lithography	*Year*	*Metal layers*	*VDD supply (V)*	*Oxide (nm)*	*Threshold voltage (V)*	*Input/output pads*	*Microwind2 rule file*
1.2µm	1986	2	5.0	25	0.8	250	Cmos12.rul
0.7µm	1988	2	5.0	20	0.7	350	Cmos08.rul
0.5µm	1992	3	3.3	12	0.6	600	Cmos06.rul
0.35µm	1994	5	3.3	7	0.5	800	Cmos035.rul
0.25µm	1996	6	2.5	6	0.45	1000	Cmos025.rul
0.18µm	1998	6	2.0	5	0.40	1500	Cmos018.rul
0.12µm	2000	7	1.5	4	0.30	1800	Cmos012.rul
0.10µm	2002	8	1.0	3	0.20	2000	Cmos010.rul
0.07µm	2005	8	0.8	2	0.15	3000	Cmos007.rul

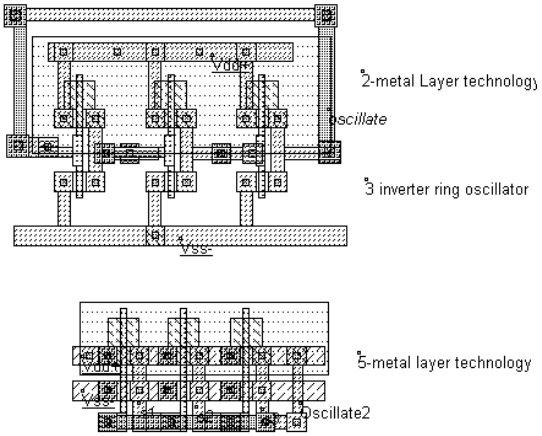


Fig. 2-6: A three-inverter ring oscillator routed with 2-metal layers and 5-metal layers technologies

As can be noticed, the number of metal layers used for interconnects has been continuously increasing in the course of the past ten years. More layers for routing means a more efficient use of the silicon surface, as for printed circuit boards. Active areas, i.e MOS devices can be placed closer from each other if many routing layers are provided (Figure 2-6).

HOW TO SIMULATE

① Start Microwind2. By default the software is configured with 0.25µm technology. Click "File ➔Open".

② Select "INV3". Click "Simulate➔ Process section in 2D".

③ Draw a line representing the location for 2D-process view. The 2D view appears.Click "OK".

④ Click "Simulate ➔ Start Simulation". Observe the oscillator frequency.

⑤ Click "File ➔Select Foundry". Select "ams08.rul" (0.8µm technology).

⑥ Ask again for the 2D view. Observe the change in the process aspect.

⑦ Ask again for analog simulation. Observe the change in frequency and voltage supply.

3 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.

3.1 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 n-channel MOS is built using polysilicon as the gate material and N+ diffusion to build the source and drain. The p-channel MOS is built using polysilicon as the gate material and P+ diffusion to build the source and drain. The symbols for the ground voltage source (0 or VSS) and the supply (1 or VDD) are also reported in figure 3-1.

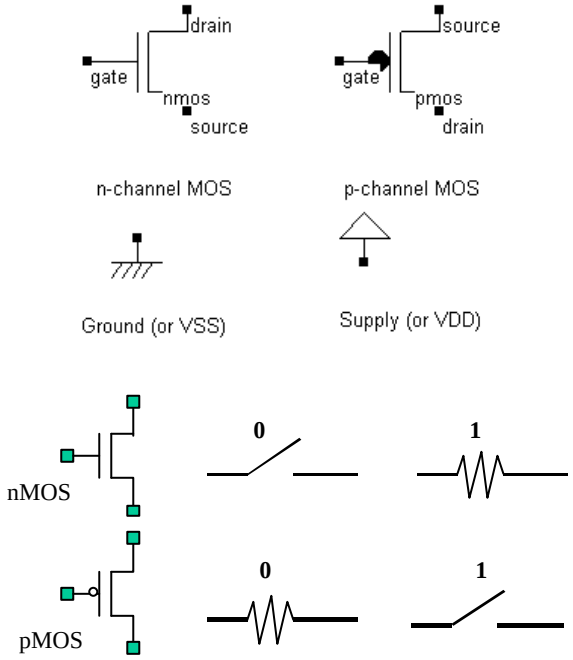


Fig. 3-1: the MOS symbol and switch

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.

3.2 Logic Simulation of the MOS

At logic level, the MOS is considered as a simple switch. Moreover, the logic switch is unidirectional, meaning that the logic signal always flows from the source to the drain. This major restriction has no physical background. In reality, the current may flow both ways. The reason why the logic MOS device enables the signal to propagate only from source to drain is purely a software implementation problem. In the logic simulator of DSCH2, an arrow indicates whether or not the current flows, and its direction (Figure 3.2). When the device is OFF, the drain keeps its last logic value, thus acting as an elementary memory.

Notice that you cannot pass any logic information from the drain to the source. Such a circuit would fail.

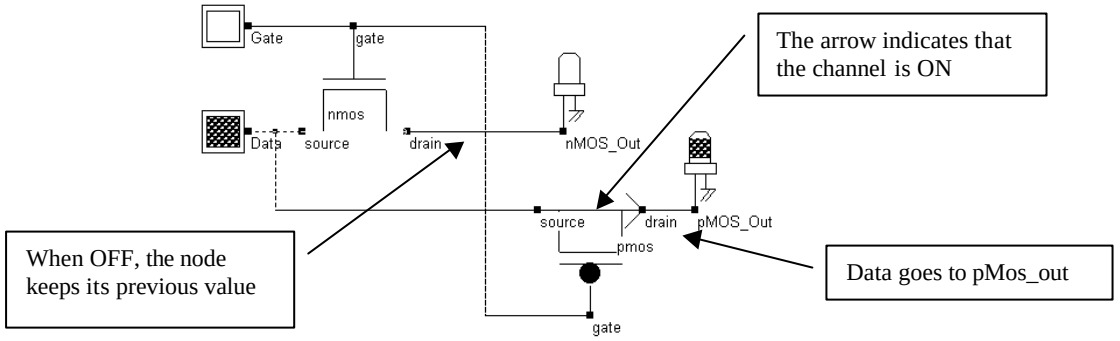


Fig. 3-2: the logic simulation of the MOS device (MosExplain.SCH)

3.3 MOS layout

We use MICROWIND2 to draw the MOS layout and simulate its behavior. Go to the directory in which the software has been copied (By default MICROWIND2). Double-click on the MicroWind2 icon.

The MICROWIND2 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 6-metal layers 0.25µm technology, consequently lambda is 0.125 µm.

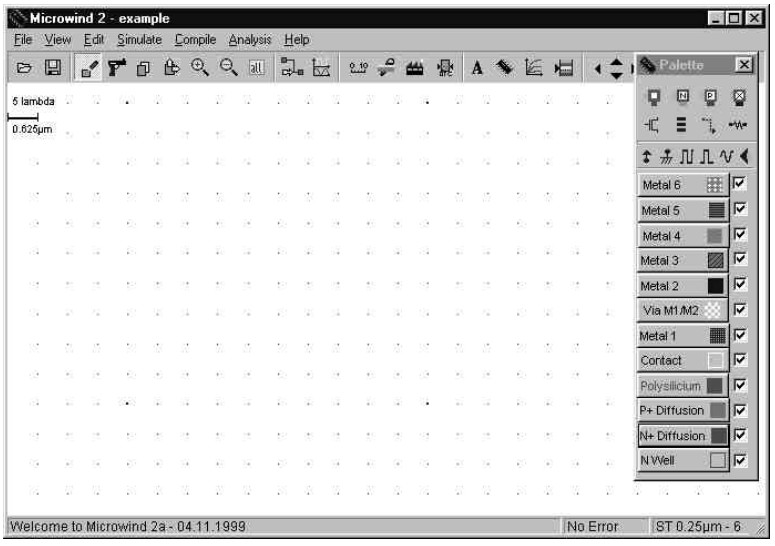


Fig. 3-3 The MICROWIND2 window as it appears at the initialization stage..

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 3-4. 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 3-4. N-diffusion boxes are represented in green. The intersection between diffusion and polysilicon creates the channel of the nMOS device.

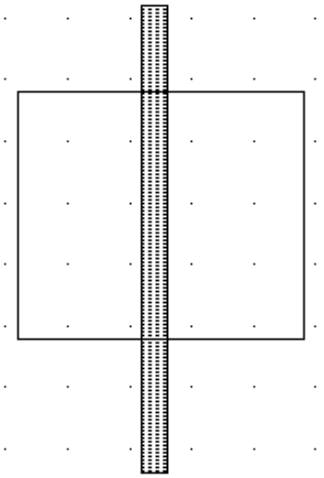


Fig. 3-4. Creating the N-channel MOS transistor

3.4 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.

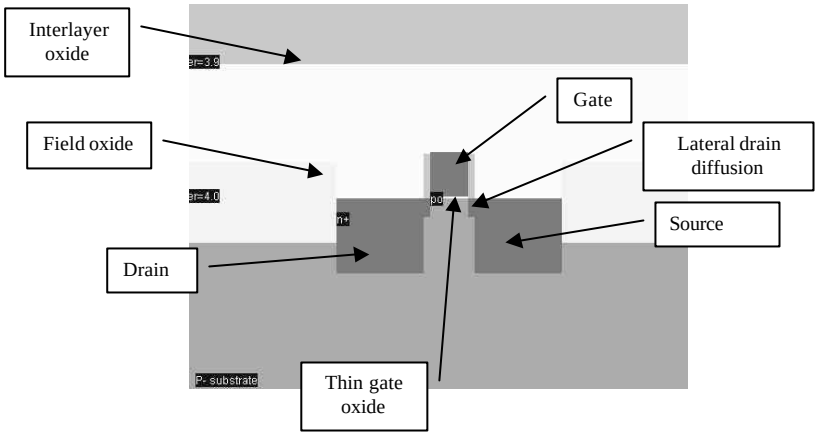


Fig. 3-5. The cross-section of the nMOS devices.

In the example of Figure 3-5, 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.

3.5 Static Mos Characteristics

Click on the *MOS characteristics* icon. The screen shown in Figure 2-6 appears. It represents the Id/Vd static characteristics of the nMOS device.

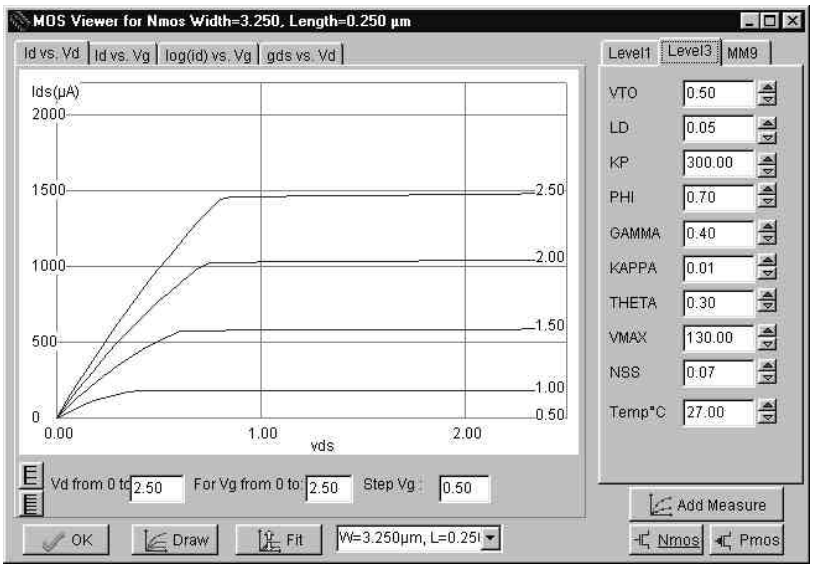
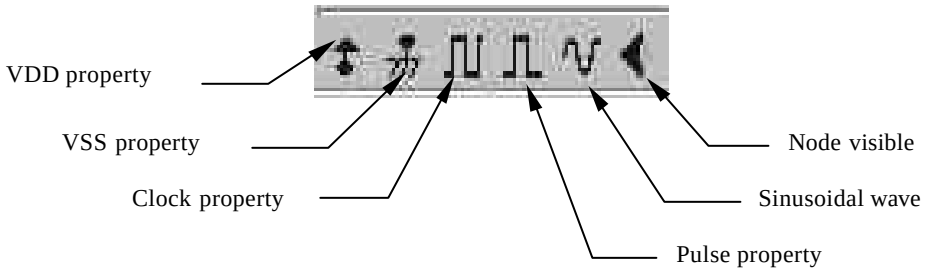


Fig. 3-6. N-Channel MOS characteristics.

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 3-6, the MOS width is 3.25µm and the length is 0.25µm. A high gate voltage (Vg =2.5V) corresponds to the highest Id/Vd curve. For Vg=0, no current flows. A maximum current around 1.5mA is obtained for Vg=2.5V, Vd=2.5V, with Vs=0.0. The MOS parameters correspond to SPICE Level 3. A tutorial on MOS model parameters is proposed later in this chapter.

3.6 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 2.1ns period (1ns at 0, 50ps rise, 1ns at 1, 50ps fall).

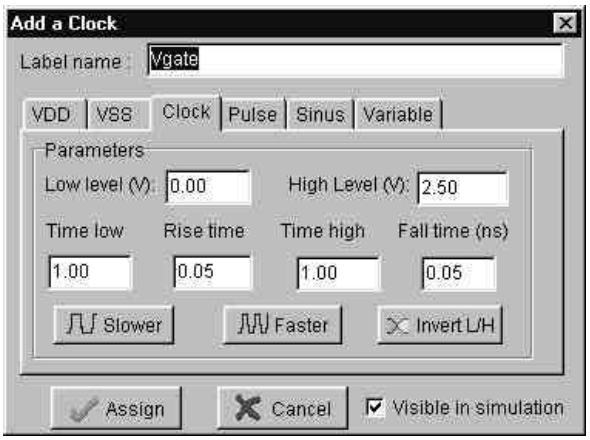


Fig. 3-7. The clock menu.

❶ 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 4.2ns 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 ».

❷ 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.

3.7 Analog Simulation

Click on **Simulate → Start Simulation**. The timing diagrams of the nMOS device appear, as shown in Figure 3-8.

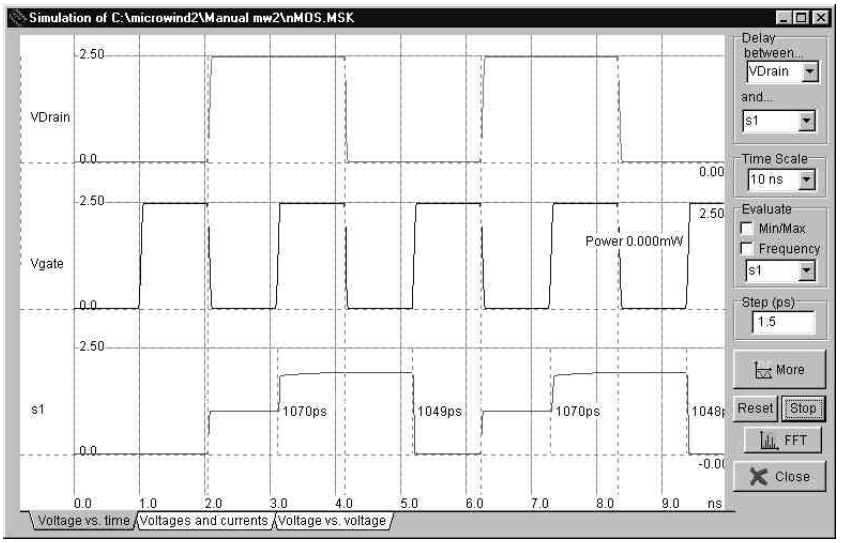


Fig. 3-8. Analog simulation of the MOS device.

When the gate is at zero, no channel exists so the node s1 is disconnected from the drain. When the gate is on, 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 s1 is around 2.0V, 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 3-9. Click on **More** in order to perform more simulations. Click on **Close** to return to the editor.

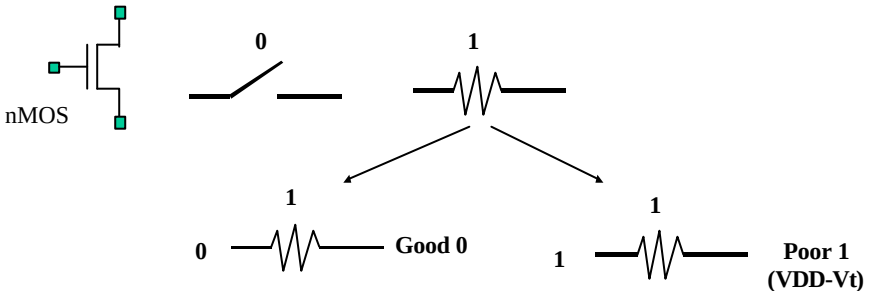


Fig. 3- 9. The nMOS device behavior summary

3.8 Layout considerations

The safest way to create a MOS device is to use the MOS generator. In the palette, click the MOS generator icon. A window appears as reported below. 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.

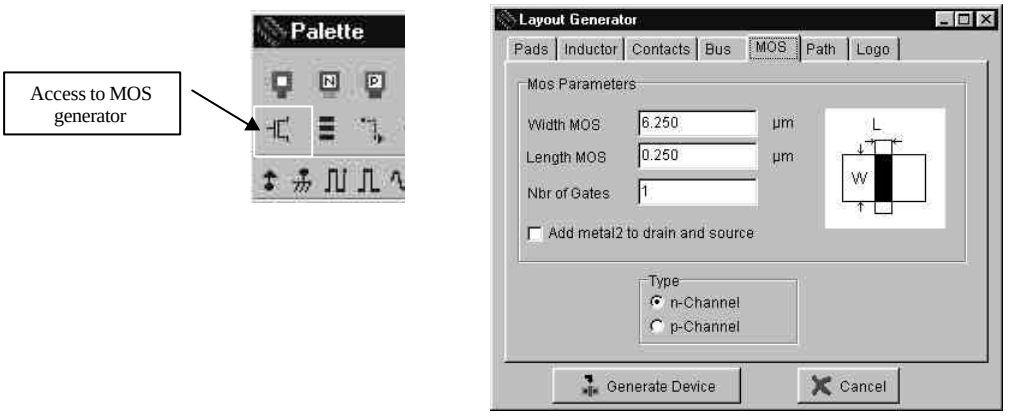


Fig. 3-10. Analog simulation of the MOS device.

3.9 The MOS Model 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 MODEL 1 described below.

MODE	CONDITION	EXPRESSION FOR THE CURRENT IDS
CUT-OFF	Vgs<0	Ids = 0
LINEAR	Vds<Vgs-Vt	$Ids = KP\frac{W}{L}((Vgs-Vt)Vds-\frac{Vds2}{2}))$
SATURATED	Vds>Vgs-Vt	$Ids = KP/2\ \frac{W}{L}\ (Vgs-Vt)^2$

With:

$$vt = VTO + GAMMA + \sqrt{(PHI - vb)} - \sqrt{PHI}$$

MOS MODEL 1 PARAMETERS			
PARAMETER	DEFINITION	TYPICAL VALUE 0.25µm	
		NMOS	pMOS
VTO	Theshold voltage	0.4V	-0.4V
KP	Transconductance coefficient	300µA/V^2	120µA/V^2
PHI	Surface potential at strong inversion	0.3V	0.3V
GAMMA	Bulk threshold parameter	0.4 V$^{0.5}$	0.4 V$^{0.5}$
W	MOS channel width	0.5-20µm	0.5-40µm
L	MOS channel length	0.25µm	0.25µm

Let us compare the simulation and the measurement, for a 10x10µm device.

❶ Click **Simulate → Mos** characteristics (Or the icon)

❷ Click **Add Measure**.

❸ Select the data file **"Nb10x0,25.MES".** The "N" means an n-channel MOS device. The "b" corresponds to a chip called "BETA" fabricated in 0.25µm technology. The values 10x0,25 means W=10µm, L=0.25µm.

❹ Select "Level 1" in the parameter list to compare LEVEL1 simulated characteristics with the measurements.

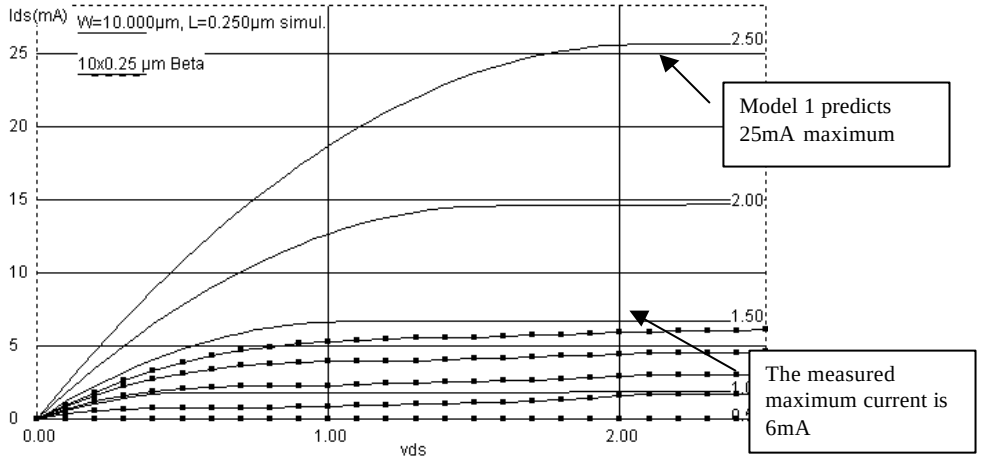


Fig. 3-11: The model 1 predict a current 4 times higher than the measurement

When dealing with sub-micron technology, the model 1 is more than 4 times too optimistic regarding current prediction, compared to real-case measurements, as shown above for a 10x0,25µm n-channel MOS.

3.10 The MOS Model 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 model 3 formulations. The formulations are derived from the model 1 and take into account a set of physical limitations in a semi-empirical way.

CUT-OFF MODE. Vgs<0

$$Ids = 0$$

NORMAL MODE. Vgs>Von

$$Ids = Keff \frac{W}{LEFF} (1+KAPPA\ vds)\ Vde\ ((Vgs-vth)- \frac{Vde}{2}))$$

with

$$von = 1.2\ \ vth$$

$$vth = VTO + GAMMA(\sqrt{PHI - vb} - \sqrt{PHI})$$

$$vde = \min (vds, vdsat)$$

$$vdsat = \ vc + vsat - \sqrt{vc^2 + vsat^2}$$

$$vsat = vgs-vth$$

$$vc = VMAX \frac{LEFF}{0.06}$$

$$LEFF = L - 2\,LD$$

$$Keff = \frac{KP}{(1 + THETA\,(vgs - vth))}$$

SUB-THRESHOLD MODE. Vgs<Von. Vds is replaced by von in the above equations.

$$Ids = Ids(von,vds)\, e^{\frac{q(vgs - von)}{nkT}}$$

TEMPERATURE EFFECTS

$\mu n = \mu n_0 (T-300)\, e^{-1.5}$

$\mu p = \mu p_0 (T-300)\, e^{-1.5}$

$vt = vt_0 - 0.002(T-300)$

MOS MODEL 3 PARAMETERS			
PARAMETER	**DEFINITION**	**TYPICAL VALUE 0.25µm**	
		NMOS	pMOS
VTO	Theshold voltage	0.4V	-0.4V
KP	Transconductance coefficient	300µA/V^2	120µA/V^2
PHI	Surface potential at strong inversion	0.3V	0.3V
LD	Lateral diffusion into channel	0.01µm	0.01µm
GAMMA	Bulk threshold parameter	0.4 V$^{0.5}$	0.4 V$^{0.5}$
KAPPA	Saturation field factor	0.01 V^{-1}	0.01 V^{-1}
VMAX	Maximum drift velocity	150Km/s	100Km/s
THETA	Mobility degradation factor	0.3 V^{-1}	0.3 V^{-1}
NSS	Substhreshold factor	0.07 V^{-1}	0.07 V^{-1}
W	MOS channel width	0.5-20µm	0.5-40µm
L	MOS channel length	0.25µm	0.25µm

The curve shown in Figure 3-12 is used to fit VTO, KP and GAMMA. Act on VTO cursors in order to shift the curves right or left, KP to adjust the slope, and GAMMA to fit the spacing between curves.

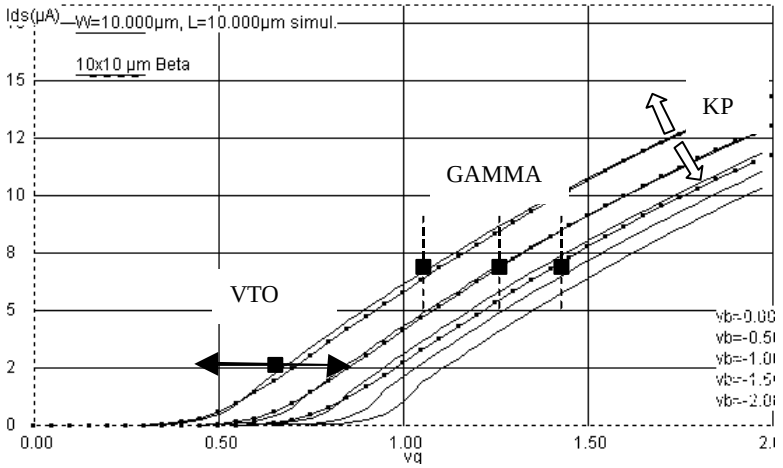


Fig. 3-12. The Id/Vg curves used to fit KP, VTO and GAMMA (Nb10x10.MES)

3.11 The BSIM4 MOS Model

A family of models has been developed at the University of Berkeley for the accurate simulation of sub-micron technology. The Berkeley Short-channel IGFET Model (BSIM) exist in several version (BSIM1, BSIM2, BSIM3). The BSIM3v3 version, promoted by the Electronic Industries Alliance (EIA) is an industry standard for deep-submicron device simulation.

A new MOS model, called BSIM4, has been introduced in 2000. A simplified version of this model is supported by Microwind2, and recommended for ultra-deep submicron 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.

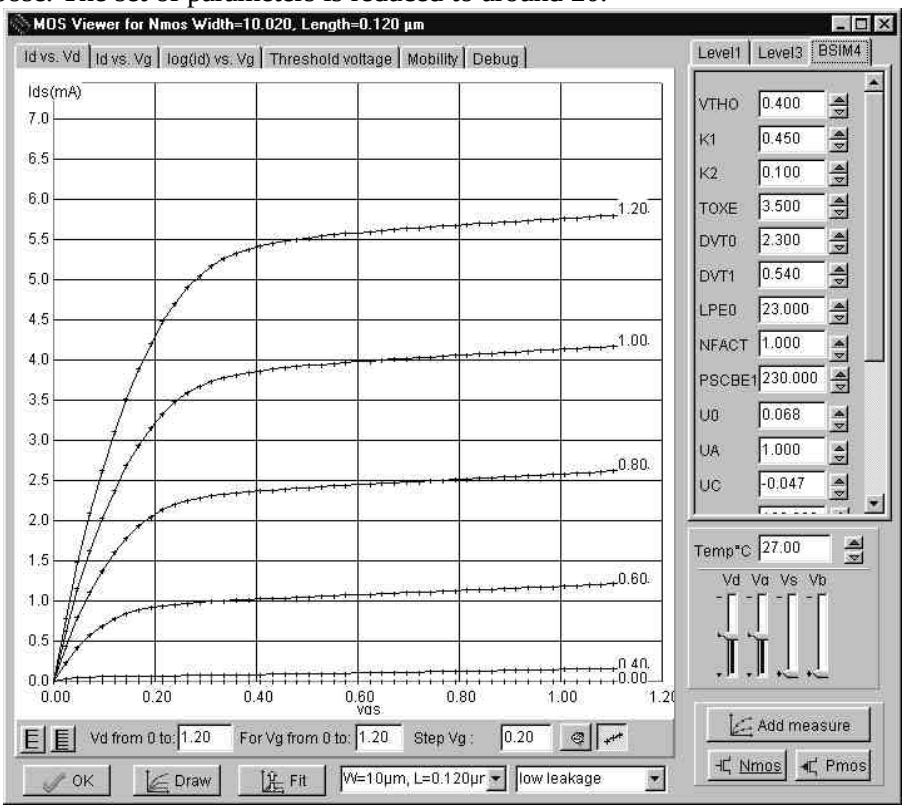


Fig.3-13: Implementation of BSIM4 within Microwind2

The general equation of the threshold voltage is presented below.

$$vth = VTHO + K1.\sqrt{(\Phi_s - Vbs} - \sqrt{\Phi_s}) - K2.Vbs + \Delta Vt_{SCE} + \Delta Vt_{NULD} + \Delta Vt_{DIBL}$$

where VTHO is the long channel threshold voltage at Vbs=0 (Around 0.5V), K1 is the first order body bias coefficient (0.5 $V^{1/2}$), Φs is the surface potential, Vbs is the bulk-source voltage, K2 is the second order body bias coefficient, ΔVt_{SCE} is the short channel effect on Vt, ΔVt_{NULD} is the non-uniform lateral doping effect, and ΔVt_{DIBL} is the drain-induced barrier lowering effect of short channel on Vt.

Concerning the formulations for mobility of channel carriers, he generic parameter is U0, the mobility of electrons and holes. The effective mobility μ_{eff} is reduced due to several effects: the bulk polarization, and the gate voltage. The equation implemented in Microwind2 is the most recent mobility model proposed in BSIM4, reported in 3-xxx.

$$m_{eff} = \frac{U0}{1 + (UA + UC.V_{bseff})(\frac{V_{gsteff} + 2(VTH0 - V_{fb} - f_s)}{TOXE})^{EU}}$$

where

- UA is the low field mobility, in m^2/V-s. Its default value is around 0.06 for n-channel MOS and 0.025 for p-channel MOS.
- UA is the first order mobility degradation coefficient, in m/V. Its default value is around 10^{-15}.
- UC is the body-effect coefficient of mobility degradation, in m/V^2. Its default value is $-0.045x10^{-15}$.
- VFB is the flat band voltage, in V.
- TOXE is the oxide thickness, in m. A typical value for TOXE in 0.12µm is 3nm (3.10^{-9}m).
- EU is a coefficient equal to 1.67 for n-channel MOS, and 1.0 for p-channel MOS.

The current Ids is computed using one single equation, as described below.

$$Ids0 = \frac{Weff}{Leff}\mu eff\ \frac{\varepsilon_0 \varepsilon_r}{TOXE} V_{gsteff}\left(1 - \frac{A_{bulk}V_{dseff}}{(2V_{gsteff} + 4.vt)}\right)\frac{V_{dseff}}{\left(1 + \frac{V_{dseff}}{\varepsilon_{sat} L_{eff}}\right)}$$

Parameter	Description	NMOS value in 0.12µm	NMOS value in 0.12µm
VTHO	Long channel threshold voltage at Vbs = 0V	0.3V	0.3V
VFB	Flat-band voltage	-0.9	-0.9
K1	First-order body bias coefficient	0.45 V1/2	0.45 V1/2
K2	Second-order body bias coefficient	0.1	0.1
LPE0	Lateral non-uniform doping parameter at Vbs = 0	2.3^{e}-10	2.3^{e}-10
DVT0	First coefficient of short-channel effect on threshold voltage	2.2	2.2
DVT1	Second coefficient of short-channel effect on Vth	0.53	0.53
ETA0	Drain induced barrier lowering coefficient	0.08	0.08
NFACTOR	Sub-threshold turn-on swing factor. Controls the exponential increase of current with Vgs.	1	1
U0	Low-field mobility	0.060 m2/Vs	0.025 m2/Vs
UA	Coefficient of first-order mobility degradation due to vertical field	11.0e-15 m/V	11.0e-15 m/V
UC	Coefficient of mobility degradation due to body-bias effect	-0.04650e-15 V-1	-0.04650e-15 V-1
VSAT	Saturation velocity	8.0e4 m/s	8.0e4 m/s

WINT	Channel-width offset parameter	0.01e-6µm	0.01e-6µm
LINT	Channel-length offset parameter	0.01e-6µm	0.01e-6µm
PSCBE1	First substrate current induced body-effect mobility reduction	4.24e8 V/m	4.24e8 V/m
PSCBE2	Second substrate current induced body-effect mobility reduction	4.24e8 V/m	4.24e8 V/m
KT1	Temperature coefficient of the threshold voltage.	-0.1V	-0.1V
UTE	Temperature coefficient for the zero-field mobility U0.	-1.5	-1.5
VOFF	Offset voltage in subthreshold region.	-0.08V	-0.08V
PCLM	Parameter for channel length modulation	1.2	1.2

3.12 Low leakage 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 " low leakage" or "High-Vt" MOS device is available as well as the normal one, recalled "high-speed MOS". The main objective is to reduce significantly the Ioff current, that is the small current that flows from between drain and source with a gate voltage 0 (Supposed to be no current in first order approximation). On the figure below, the low leakage MOS device (right side) has an Ioff current reduced by a factor 50, thanks to a higher threshold voltage (0.45V rather than 0.35V).

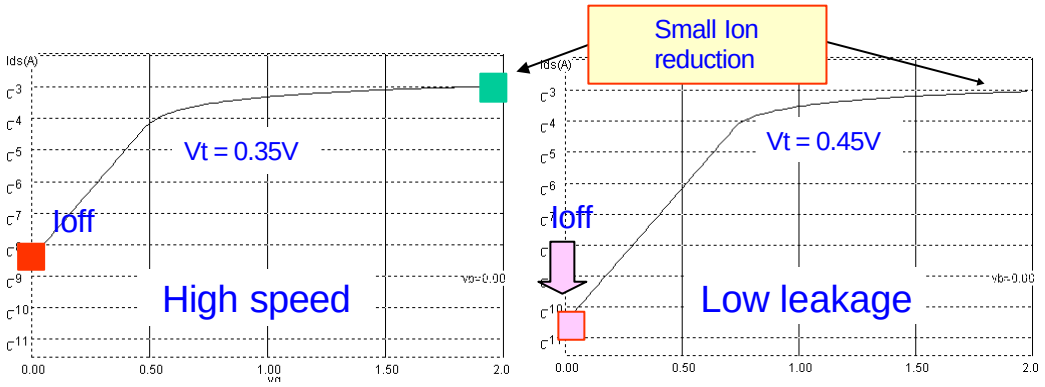


Fig. 3-14: Low leakage MOS for lower Ioff current

The main drawback of the "Low leakage" MOS device is a 30% reduction of the Ion current, leading to a slower switching. High speed MOS devices should be used in the case of fast operation linked to critical nodes, while low leakage MOS should be placed whenever possible, for all nodes where a maximum switching speed is not required.

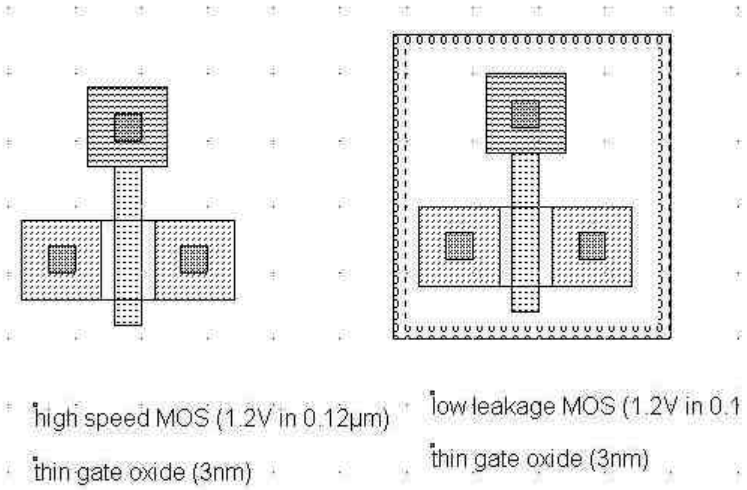


Fig. 3-15: High speed and Low leakage MOS layout. The only difference is the option layer configured for the low leakage option

3.13 High voltage MOS

Integrated circuits with low voltage internal supply and high voltage I/O interface are getting common in deep sub-micron technology. The internal logic of the integrated circuit operates at very low voltage (Typically 1.0V in 0.12µm), while the I/O devices operate in standard voltages (2.5, 3.3 or 5V). The input/output structures work at high voltage thanks to specific MOS devices with thick oxide, while the internal devices work at low voltage with optimum performances.

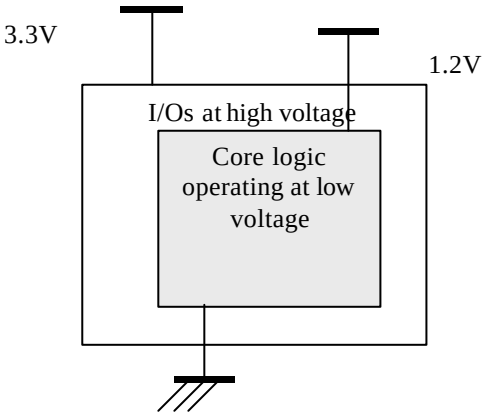


Fig. 3-16: Interfacing low voltage logic signals with high voltage I/Os requires specific circuits operating in high voltage mode.

For I/Os operating at high voltage, specific MOS devices called "High voltage MOS" are used. We cannot use high-speed or low leakage devices as their oxide is too small. A 2.5V voltage would damage the gate oxide of a high-speed MOS in 0.12µm technology. 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.

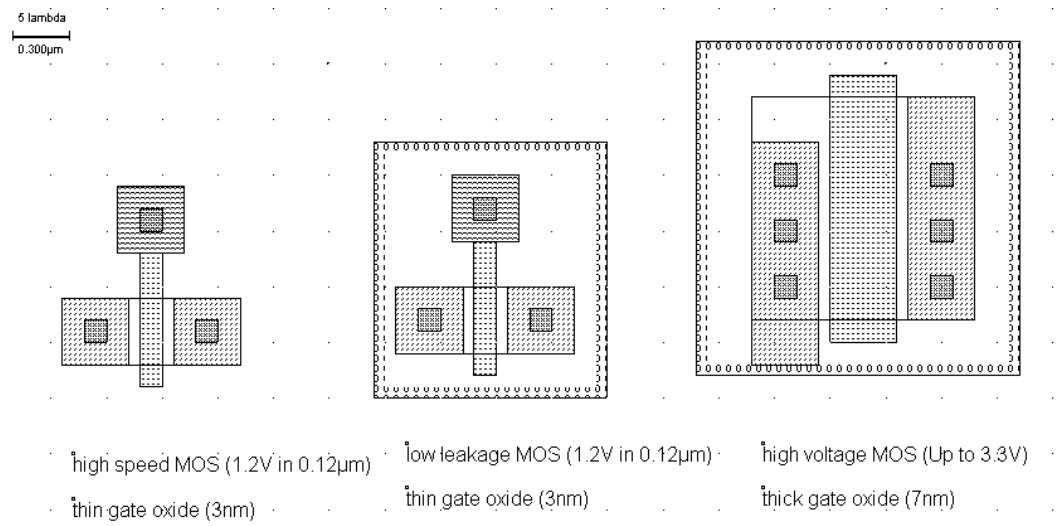


Fig. 3-17: High speed, low leakage and high voltage MOS. The high voltage MOS has a minimum length longer than for the other MOS. The gate oxide is also thicker to handle high voltage operation.

3.14 Temperature effects on the MOS

The MOS device is sensitive to temperature. Two main parameters are concerned: the threshold voltage VTO and the transconductance coefficient KP that decrease with temperature increase. The physical background is the degradation of mobility of electrons and holes when the temperature increase, due to a higher atomic volume of the crystal underneath the gate, and consequently less space for the current carriers. The modeling of the temperature effect is as follows:

$KP(T) = KP(T_0)\ (T- T_0)\ e^{-1.5}$

$VTO(T) = VTO(T_0)-0.002\ (T- T_0)$

With T_0 =300°K generally

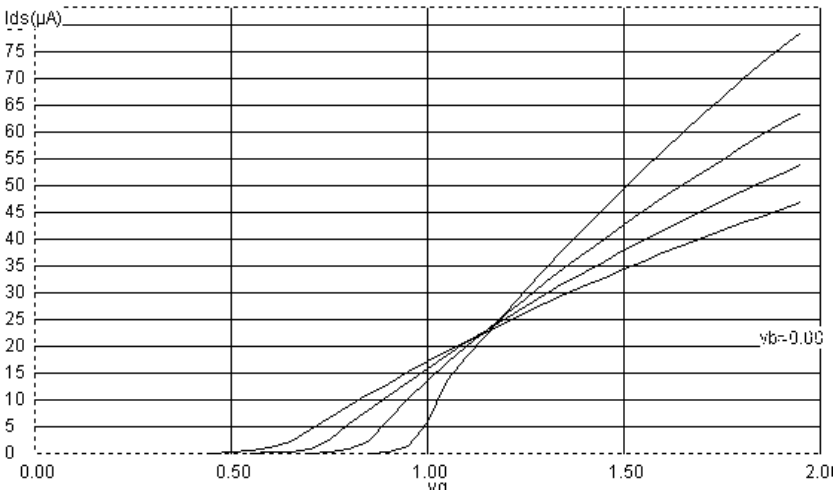


Fig. 3-18. Effect of temperature on the MOS device characteristics

To obtain this curve, proceed as follows:

❶ Click the icon MOS characteristics

❷ Select one MOS in the design or click anywhere

❸ Select the curve Id/Vg

❹ Enable the screen memory mode by a click on this icon.

Change the temperature. The change in the slope is shown. You may reduce the number of Id curves by putting a 0.0 in the field 'For Vb from 0 to :'.

In Microwind2, you can get access to temperature using the command **Simulate -> Simulation Parameters**. The screen below appears. The temperature is given in °C.

3.15 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 3-19. 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.

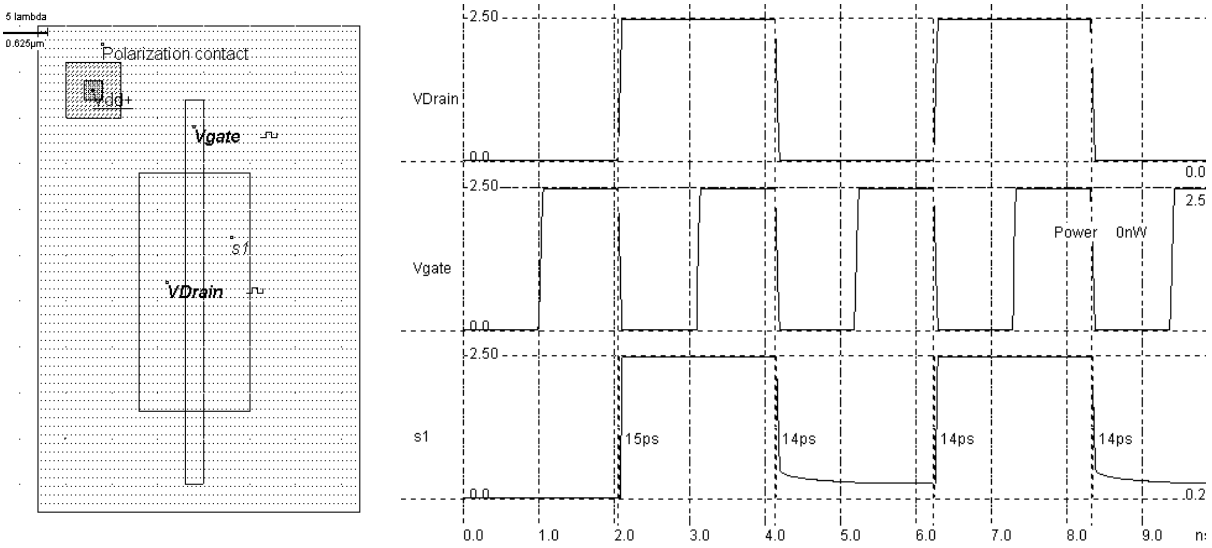


Fig. 3-19. Layout and simulation of the p-channel MOS (pMOS.MSK)

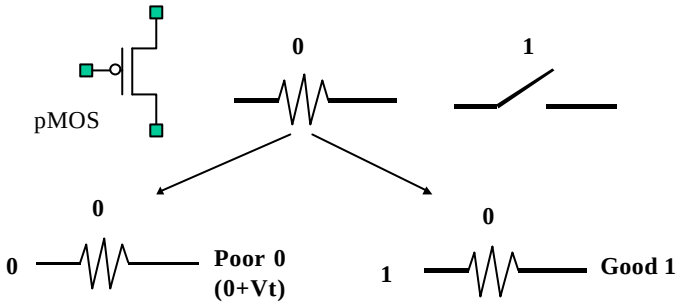


Fig. 3-20. Summary of the performances of a pMOS device

From the simulation of figure 3-19, 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. The summary of the p-channel MOS performances is reported in figure 3-20.

3.16 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 the logic level 0. Thus, a perfect pass gate can be constructed from the combination of nMOS and pMOS devices, leading to improved performances.

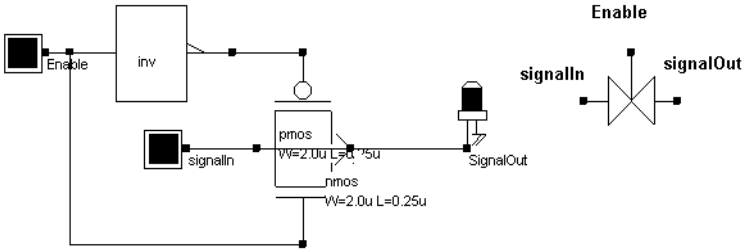


Fig. 3-21. The transmission gate

The transmission gate let a signal flow if Enable is asserted. To pass logic signals well, both a n-channel device and a p-channel device are used, as shown in figure 3-21. The main drawback is the need for two control signals Enable and /Enable, thus an inverter is usually required.

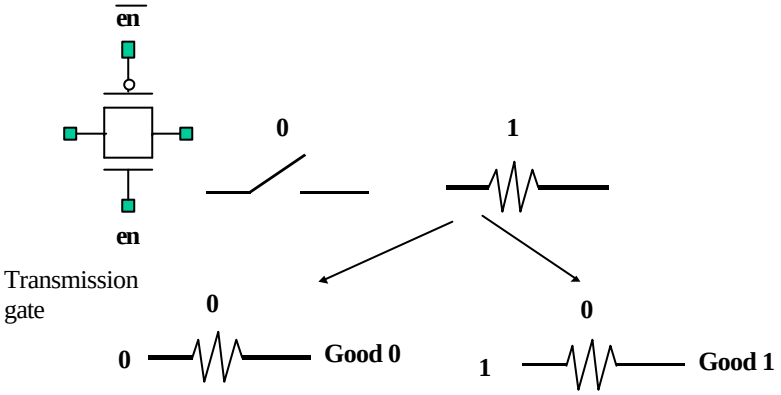


Fig. 3-22 The transmission gate used to pass logic signals

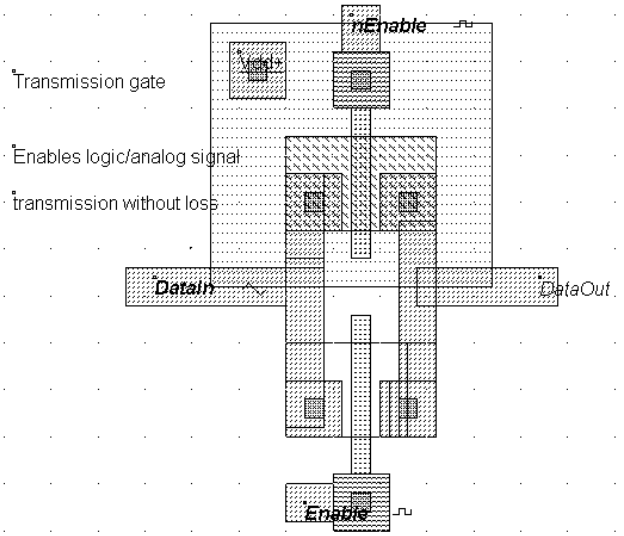


Fig. 3-23. Layout of the transmission gate (TGATE.MSK)

The layout of the transmission gate is reported in figure 3-23. 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 nEnable is the opposite of Enable. The operation of the transmission gate is illustrated in figure 3-24. A sinusoidal wave with a frequency of 2GHz is assigned to DataIn. With a zero on Enable (And a 1 on nEnable), the switch is off, and no signal is transferred. When Enable is asserted, the sinusoidal wave appears nearly identical to the output.

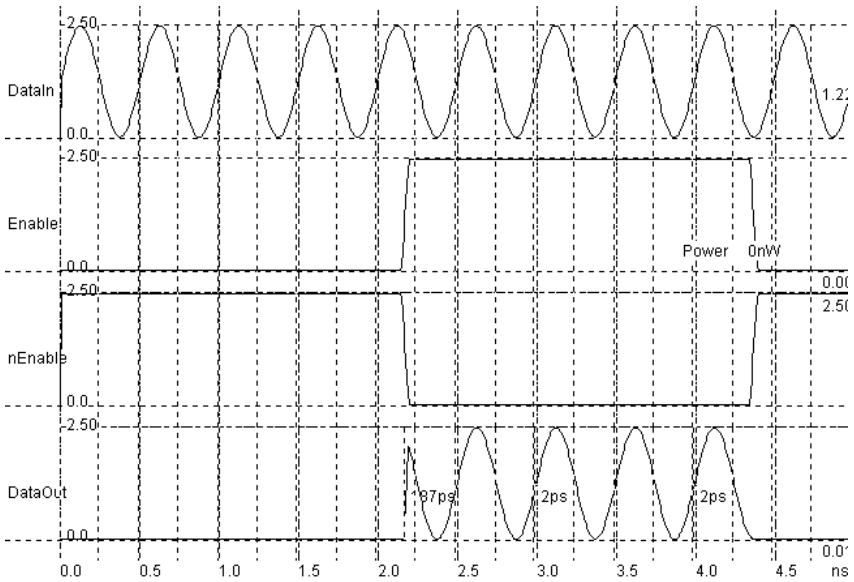


Fig. 3-24. Simulation of the transmission gate (TGATE.MSK)

4 The Inverter

This chapter describes the CMOS inverter at logic level, using the logic editor and simulator DSCH2, and at layout level, using the tool MICROWIND2.

4.1 The LOGIC Inverter

In this section, an inverter circuit is loaded and simulated.

① Click on the icon above to activate the Dsch2 software.

② Click *File→ Open* in the main menu. Select "INV.SCH" in the list. In this circuit are on button situated on the left side of the design, namely A, an inverter and a led.

③ Click *Simulate→ Start simulation* in the main menu.

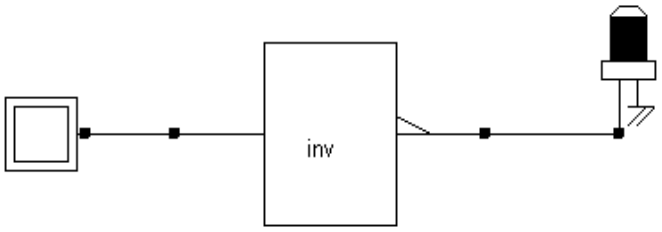


Fig. 4.1: The schematic diagram including one single inverter (Inverter.SCH)

④ Now, click inside the buttons situated on the left part of the diagram. The result is displayed on the lamps. The red value indicates logic 1, the black value means a logic 0.
⑤ Click the button "Stop simulation" shown in the above picture. You are back to the editor.
⑤ Click the above icon to get access to the chronograms of the simulation.

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 (0.06ns in the default 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 0.1ns delay).

4.2 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 (Fig. 4-2 left), the nMOS is switched off while PMOS passes VDD through the output. When the input signal is logic 1 (Fig. 4-3 right), the pMOS is switched off while the nMOS passes VSS to the output.

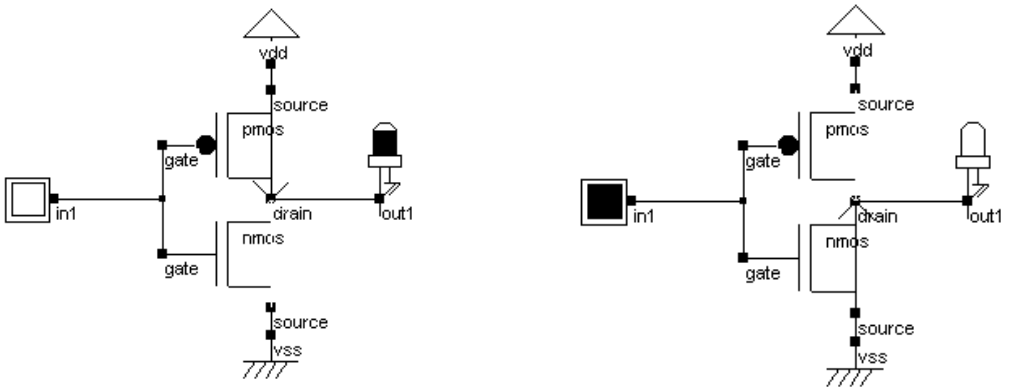


Fig. 4-3: The MOS Inverter (File CmosInv.sch)

The inverter consumes power during transitions, due to two separate effects. The first is short circuit power arising from momentary short-circuit current that flow from *VDD* to *VSS* when the transistor functions in the incomplete-on/off state. This state occurs briefly during transitions of the output, either from 0 to 1 or from 1 to 0 (Fig. 4-4).

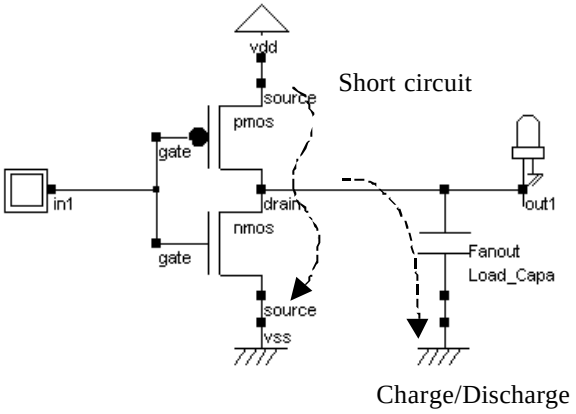


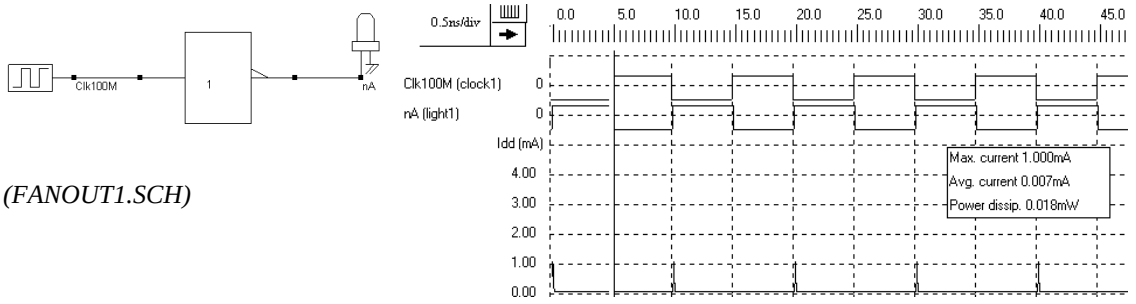
Fig. 4-4: Short circuit current in CMOS inverters

The second is the charging/discharging power, which depends on the output wire capacitance. With small loading, the short circuit current is dominant. But as the number of gates connected to the inverter increase, the load capacity increases. Consequently, the charging and discharging current starts to dominate the short circuit current.

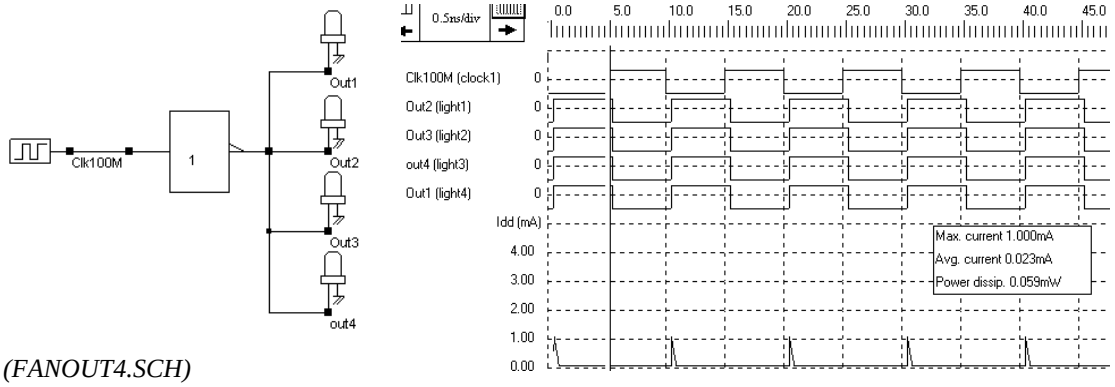
4.3 Fanout effect

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. An inverter circuit is simulated using different clock, fanout and supply conditions. The initial configuration is a 100MHz clock, one output connected to the inverter and a supply voltage 2.5V. To investigate the fanout effect on the consumption, we simulate first the inverter with one single output. In the simulation chronograms, we observe that 0.018 mW with a fanout 1. The corresponding file is FANOUT1.SCH.

Now, we add other lights to the output node, thus increasing the charge capacitance. In the simulation chronograms, both the inverter delay and the power consumption have increased (0.059 mW with a fanout of 4). The power consumption linearly increases with the load capacitance. This is mainly due to the current needed to charge and discharge that capacitance. The corresponding file is FANOUT4.SCH



(FANOUT1.SCH)



(FANOUT4.SCH)

Fig. 4-5: Power consumption increase with the fanout (Fanout1.SCH and Fanout4.SCH)

In summary, three factors contribute to the power consumption: the load capacitance *C*, the supply voltage *VDD* and the clock frequency *f*. For a CMOS inverter, this relation is represented by the equation below. The equation remains valid for more complex gates, although some extra considerations have to be taken into account.

$$P = k\ C\ VDD^2\ f$$

Where:

k: technological factor (close from 1)

C: Output load capacitance (Farad)

VDD: supply voltage (V)

f: Clock frequency (Hz)

4.4 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. The proposes size is 1.25µm for the width, 0.25µm for the length. Simply click “Generate Device”, and click on the middle of the screen to fix the MOS 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 result is displayed in figure 3-7.

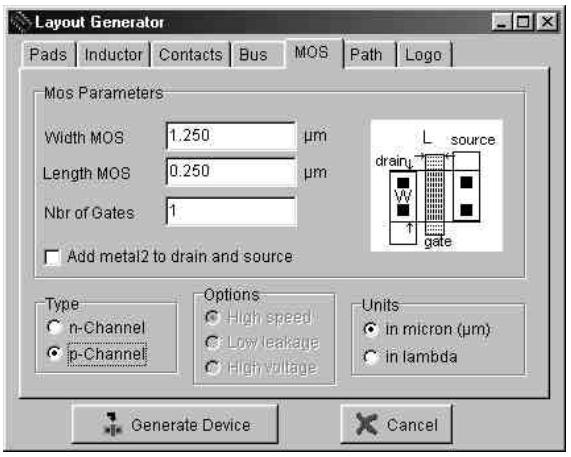


Fig. 4-6. nMOS and pMOS devices placed on the layout

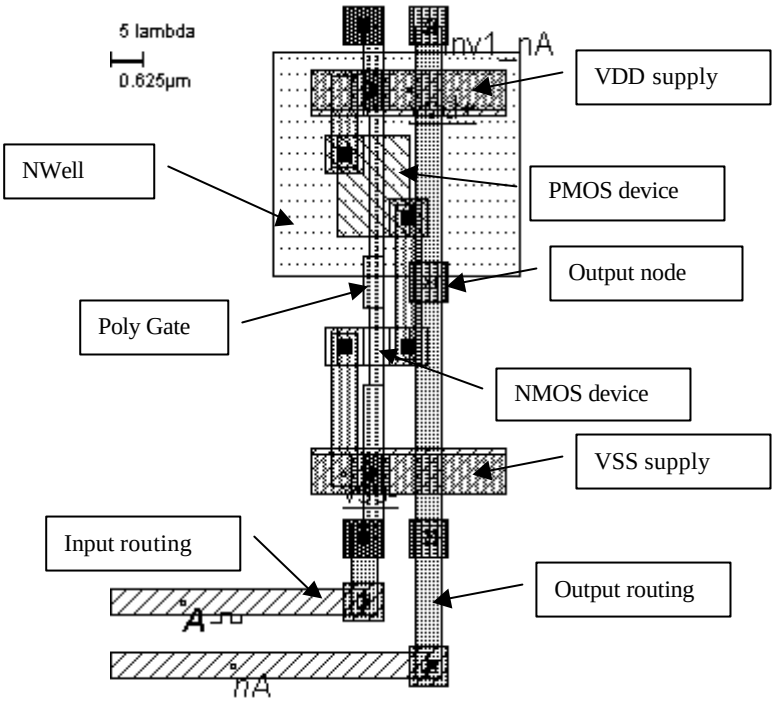


Fig. 4-7. The circuit 'Inverter' compiled into layout

4.5 Analog simulation of the INVERTER

Click *Simulate→ Start Simulation* or the icon above. The simulation of the circuit is performed. You may verify the correct behavior of the inverter cell.

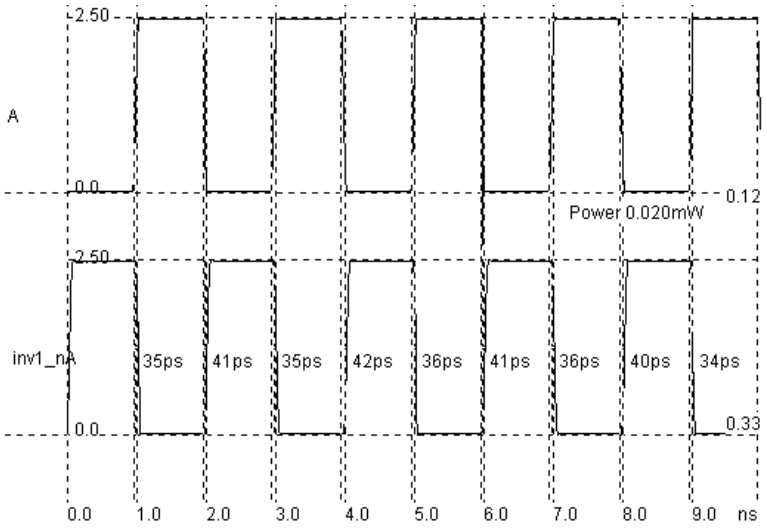


Fig. 4-8: The analog simulation of the circuit 'Inverter.MSK' using Microwind2

4.6 2D View of the Process

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 fabrication principles. A click of the mouse at the first point and the release of the mouse at the second point give the cross-section.

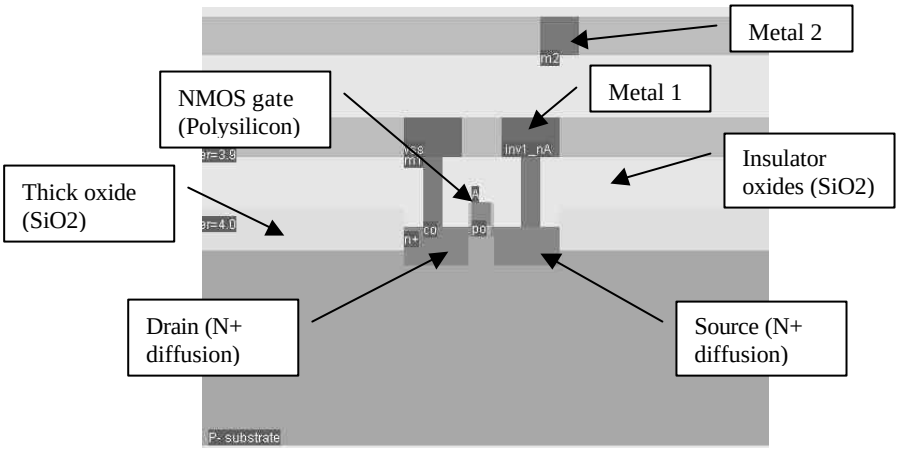


Fig. 4-9 The 2D process section of the 'Inverter' circuit near the nMOS device

4.7 3D View of the Process

Click *Simulate→ Process steps in 3D* or the icon above. The simulation of the CMOS fabrication process is performed, step by step. On figure 4-10, the picture represents the nMOS device, pMOS device, common polysilicon gate and contacts, together with the metal layers stacked on the top of the active devices.

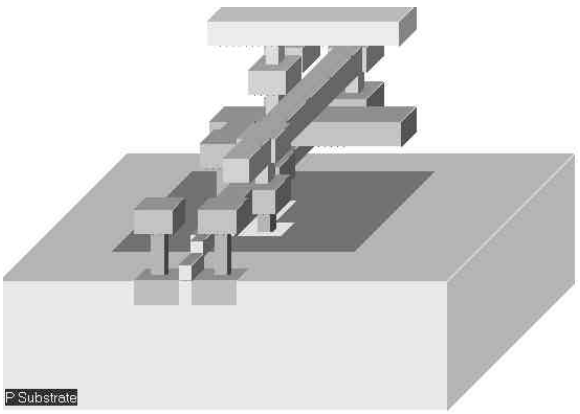


Fig.4-10 The step-by-step fabrication of the 'Inverter' circuit

4.8 3-STATE INVERTER

Until now all the symbols produced the value logic ‘0’ and logic ‘1’. However, if two outputs are connected together, as the left circuit shown below, it will provoke a circuit error. In order to avoid such conflicts, specific symbols are used, featuring the possibility to remain in a ‘high impedance’ state.

The 3-state symbol used below is *Bufif1,* and it consists of the logic buffer and an enable control. There also exists a 3-state inverter *Notif1*. The output remains in ‘high impedance’ as long as the enable ‘En’ is set to level ‘0’. The truth table of the 3-state inverter is reported below.

NOTIF1

In	En	Out
x	0	H
0	1	1
1	1	0

Fig. 4-11 : Truth table of the three state-inverter

5 Basic Gates

5.1 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.

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

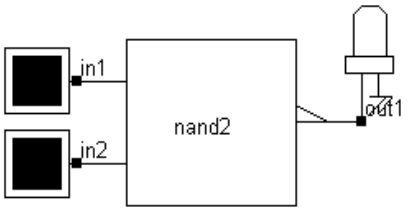


Fig. 5-1. The truth table and symbol of the NAND gate

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. 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”.

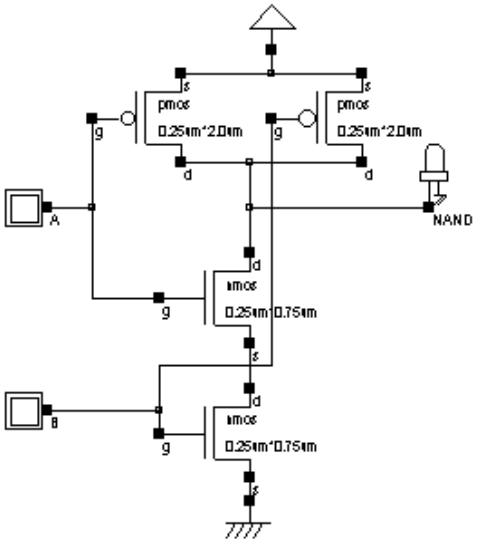
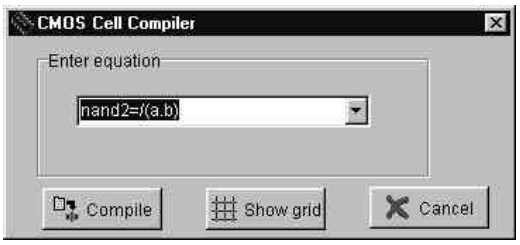
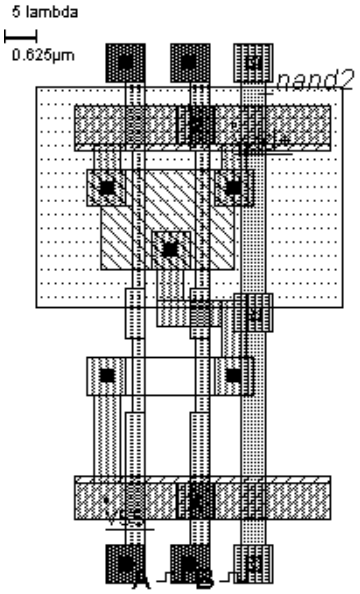


Fig. 5-2. The truth table and schematic diagram of the CMOS NAND gate design.

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 Microwind2. In this case, complete the following procedure:



In Microwind2, 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.

Fig. 5-3. A NAND cell created by the CMOS compiler.

The 2D-process viewer is a useful tool to display the two nMOS in series and the two pMOS in parallel. Select the corresponding icon and draw an horizontal line in the layout in the middle of the nMOS channels. The figure below appears. In fig. 5-4, the output is connected to the VSS supply only if A=1 and B=1.

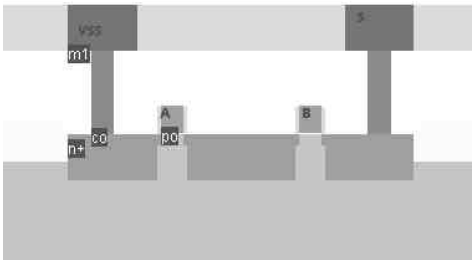


Fig. 5-4. The nMOS devices in serial in the NAND gate

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.

5.2 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 5-5 are significantly higher than for the NAND2 gate alone, due to the inverter stage delay.

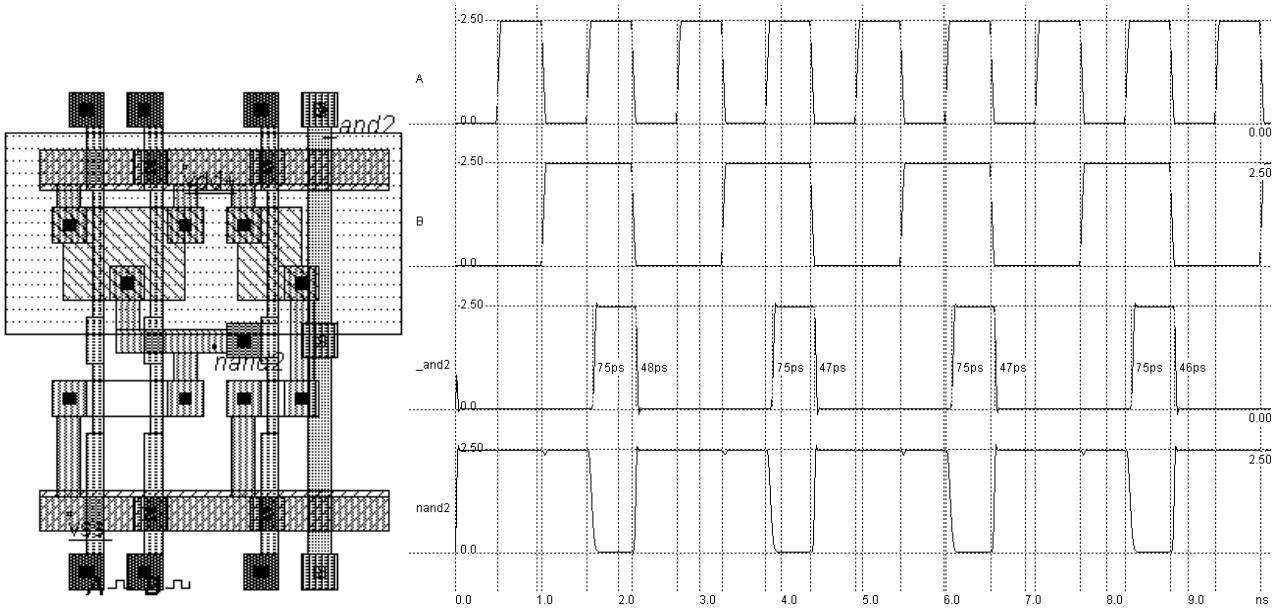


Fig. 5-5: Layout and simulation of the AND gate

5.3 The 3-Input OR Gate

The truth-table and the schematic diagram of the three-input OR gate are shown in Figure 5-6. You may use the DSCH2 logic editor to design a schematic diagram based on the OR gate, generate a Verilog description, and compile the text file in Microwind2. As can be seen again in the final layout, the OR gate is the sum of a NOR3 gate and an inverter.

OR 3 Inputs

A	B	C	Or3
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

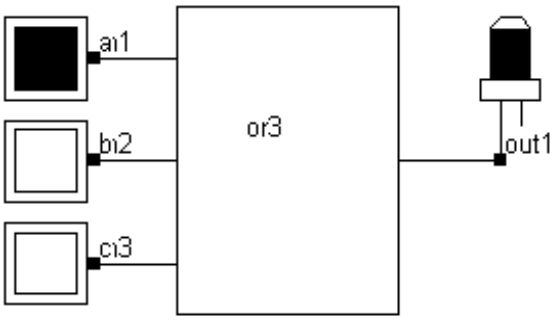


Fig. 5-6. The truth table and symbol of the OR3 gate

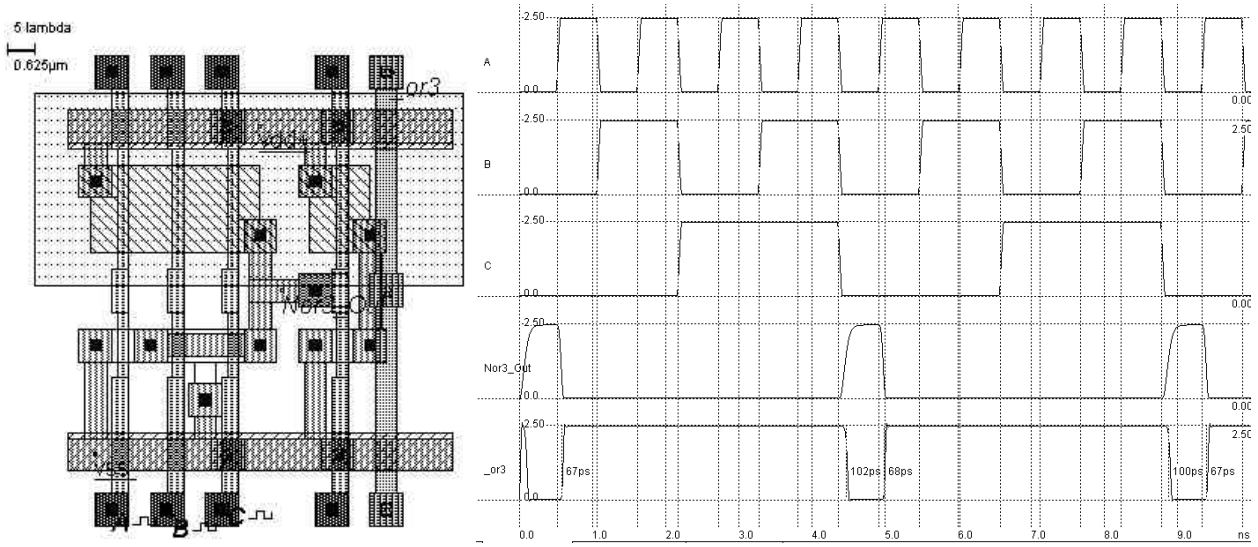
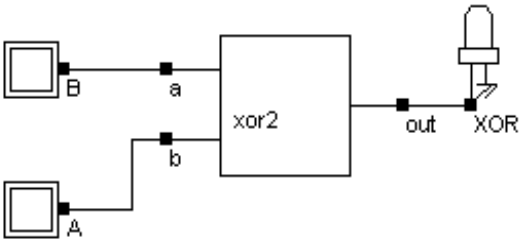


Fig. 5-7. Layout and simulation of the OR3 gate (OR3.MSK).

5.4 The XOR Gate

XOR 2 inputs

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



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.

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.

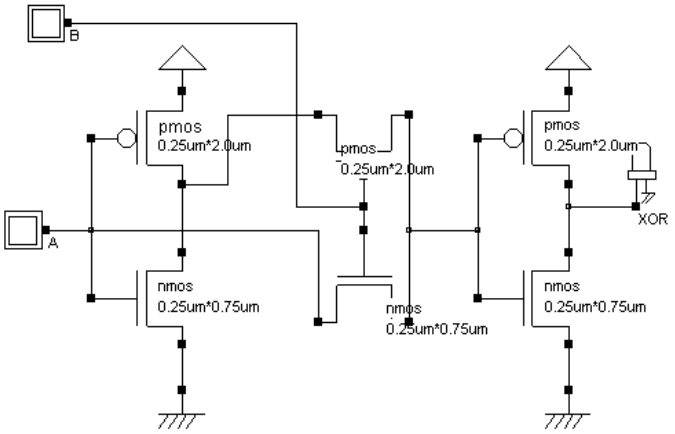


Fig. 5-8. The schematic diagram of the XOR gate (XORCmos.SCH)

You may use DSCH2 to create the cell, generate the Verilog description and compile the resulting text. In Microwind2, the Verilog compiler is able to construct the XOR cell as reported in Figure 5-9. 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.

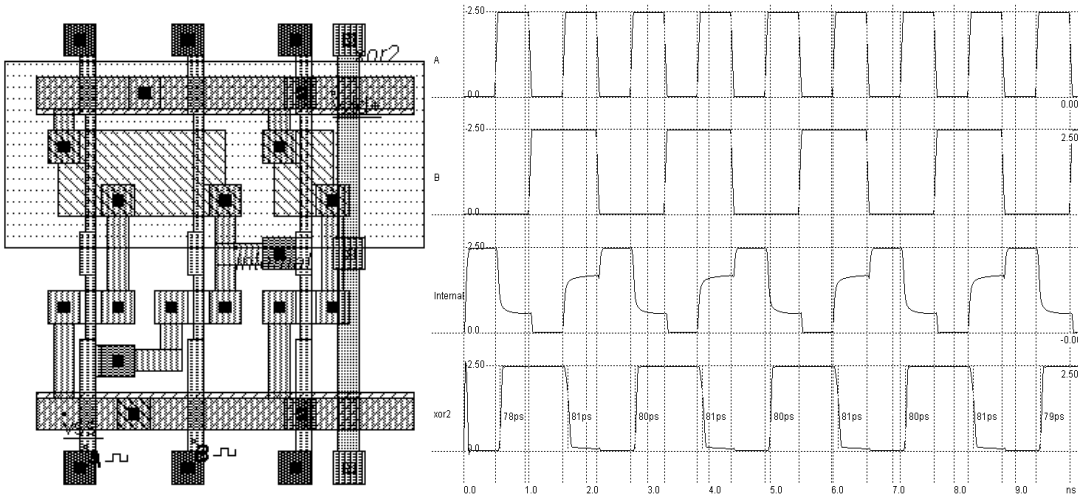


Fig. 5-9. Layout and simulation of the XOR gate (XOR.MSK).

5.5 Complex Gates

The complex gate design technique applies for any combination of operators **AND** and **OR**. The technique produces compact cells with higher performances in terms of spacing and speed than conventional logic circuits. To illustrate the concept of complex gates, let us take the example of the following Boolean equation:

F= /(A+(B.C))

The logic circuit corresponding to this equation is reported below. The circuit is built using a 2-input NOR and a 2-input AND cell, that is 10 transistors and three delay stages.

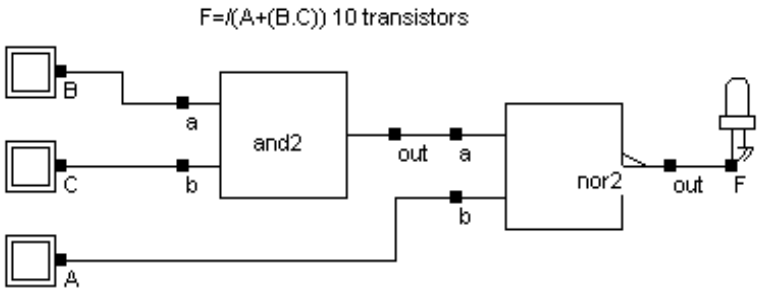


Fig. 5-10: The conventional schematic diagram of the function F=/(A+(B.C))

A much more compact exists in this case (Figure 5-11), consisting in the following steps:

1. For the nMOS network, translate the AND operator '.' into nMOS in series , and the OR operator '+' in nMOS in parallel.
2. For the pMOS network, translate the AND operator '.' into pMOS in parallel, and the OR operator '+' in pMOS in series.
3. If the function is non-inverting , as for 'F=A+(B.C)', an inverter is mandatory.

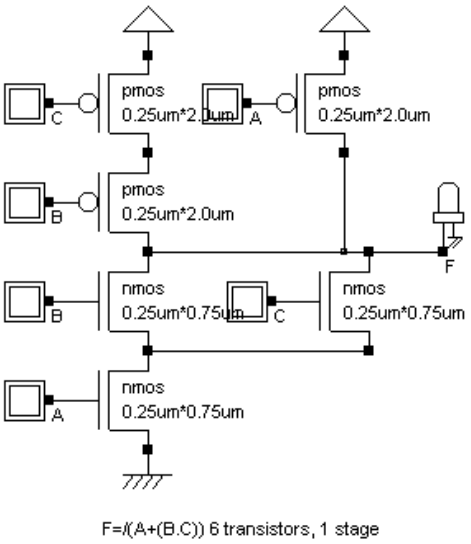


Fig. 5-11: The complex gate implementation of the function F=/(A+(B.C))

Microwind2 is able to generate the CMOS layout corresponding to any description based on the operators **AND** and **OR**, using the command **Compile -> Compile one line**. Using the keyboard, enter the cell equation, or modify the items proposed in the list of examples. In the one-line equation, the first parameter is the output name. In the present case that name is **s**. The sign '=' is obligatory. The '~' sign corresponds to the operation NOT and can be *used only* right after the '=' sign. The parenthesis '(' ')' are used to build the function, where '&' is the AND operator , '|' is the OR operator, and '^' is the XOR operator.

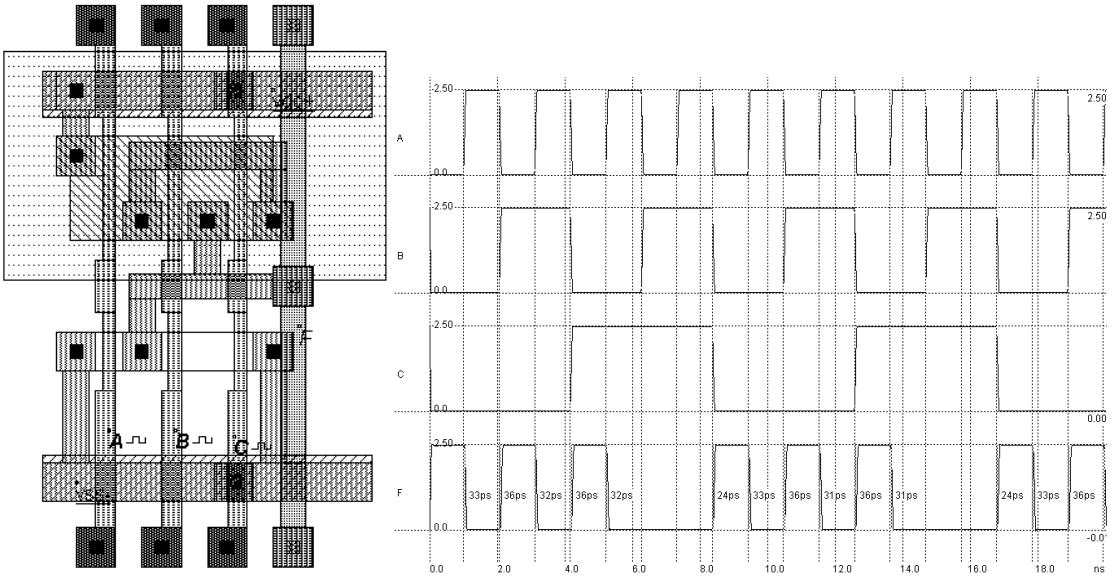


Fig. 5-12. A compiled complex gate and its analog simulation (ComplexABC.MSK)

5.6 Multiplexor

Multiplexing means transmitting a large amount of information through a smaller number of connections. A digital multiplexer 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 multiplexer is a basic cell called the transmission gate. The transmission gate let a signal flow if Enable is asserted. Remember that the n-channel MOS is only good for low signals, and the p-channel MOS is only good for high signals. To pass logic signals well, both a n-channel device and a p-channel device must be used, as shown in figure 5-13. The main drawback is the need for two control signals Enable and /Enable, thus an inverter is required.

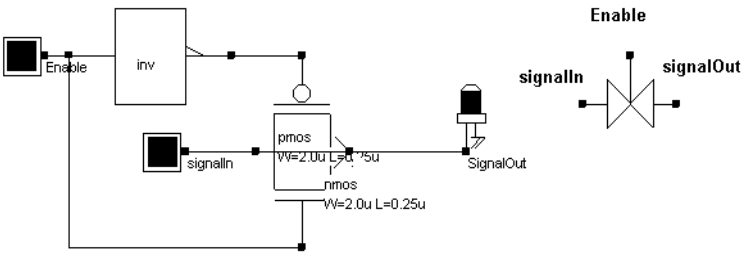


Fig. 5-13. The transmission gate used as a multiplexor

In DSCH2, a transmission gate symbol exists. 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.

5.7 4 to 1 Multiplexer

The multiplexer is a very useful function and has a multitude of application. The selection of a particular input line is controlled by a set of selection lines. Normally, there are 2^n input lines and n selection lines whose bit combinations determine which input is selected. Figure 5-14 shows the transmission gate implementation of the 4 to 1 multiplexer. In the configuration S1=1, S2=0, the input 'C' is connected to the output.

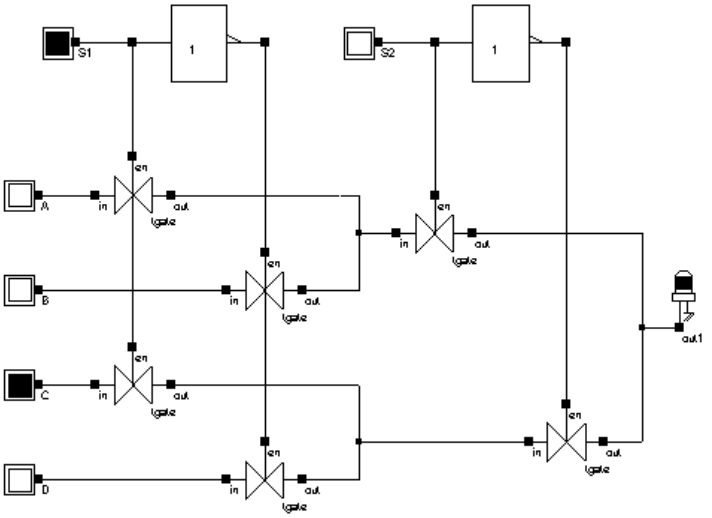


Fig. 5-14: 4 to 1 multiplexing based on transmission gates (Mux4to1.sch)

5.8 Keyboard multiplexor

Figure 5-15 gives an example of 2 multiplexed hexadecimal keyboards sharing the same hexadecimal display, using transmission gates. We use a clock to generate an alternative selection of the keyboard information.

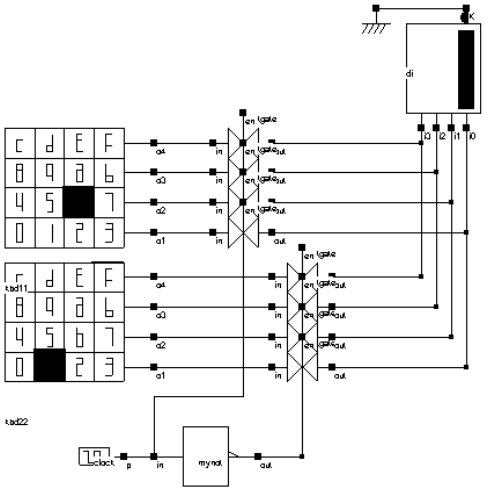


Fig. 5-15 Keyboard multiplexing based on transmission gates (Mux2Kbd.sch)

6 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.

6.1 Half-Adder Gate

The Half-Adder gate truth-table and schematic diagram are shown in Figure 6-1. 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

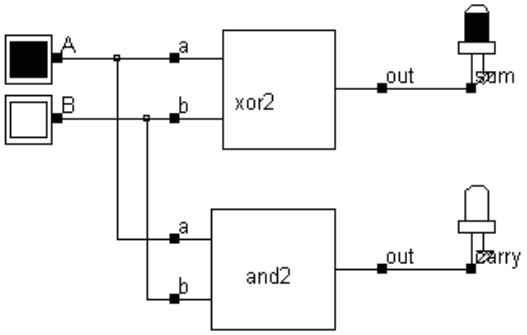


Fig. 6-1. Truth table and schematic diagram of the half-adder gate (HADD.MSK).

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 through the **File -> Open** and **HADD.MSK** sequence.

VERILOG COMPILING. Use DSCH2 to create the schematic diagram of the half-adder. Verify the circuit with buttons and lamps. Save the design under the name 'hadd.sch' using the command **File -> Save As.** Generate the Verilog text by using the command **File -> Make Verilog File**. In Microwind2, click on the command **Compile -> Compile Verilog File**. Select the text file 'hadd.txt'.

```
module Hadd( B,A,sum,carry);
 input B,A;
 output sum,carry;
 xor xor1(sum,B,A);
 and and1(carry,A,B);
endmodule
```

Click on **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 6-2 appear and you should verify the truth table of the half-adder. Click on **Close** to return to the editor

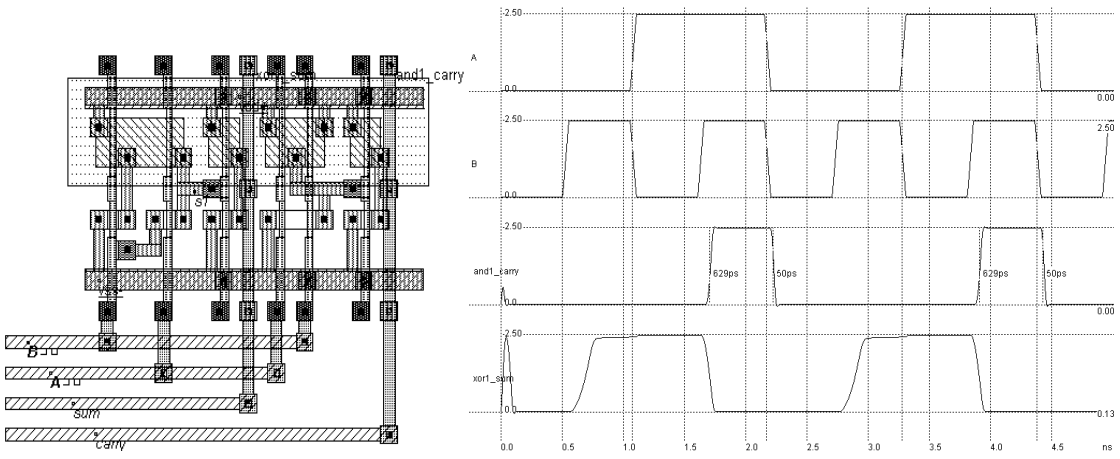


Fig. 6-2. Compiling and simulation of the half-adder gate (Hadd.MSK)

6.2 Full-Adder Gate

The truth table and schematic diagram for the full-adder are shown in Figure 6-3. 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

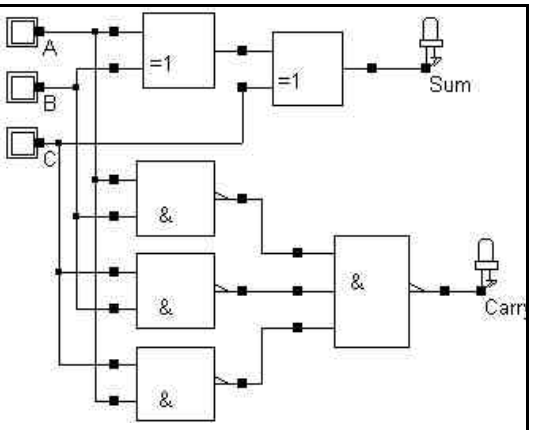


Fig. 6-3. The truth table and schematic diagram of a full-adder(FADD.SCH)

6.3 Full-Adder Symbol in DSCH

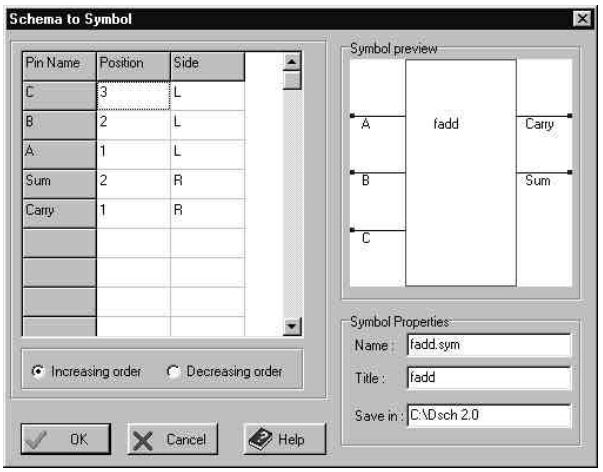


Fig. 6-4. Create a symbol from the schematic diagram

In order to build hierarchical designs using the adder, we detail the procedure to generate the symbol of the full-adder from its schematic diagram. In DSCH2, click the above icon, the screen of figure 6-4 appears. Simply click 'OK. The symbol of the full-adder is created, with the name 'Fadd.sym' in the current directory. Use the command "Insert -> Symbol" to include this symbol into a new circuit.

6.4 Full-Adder Layout

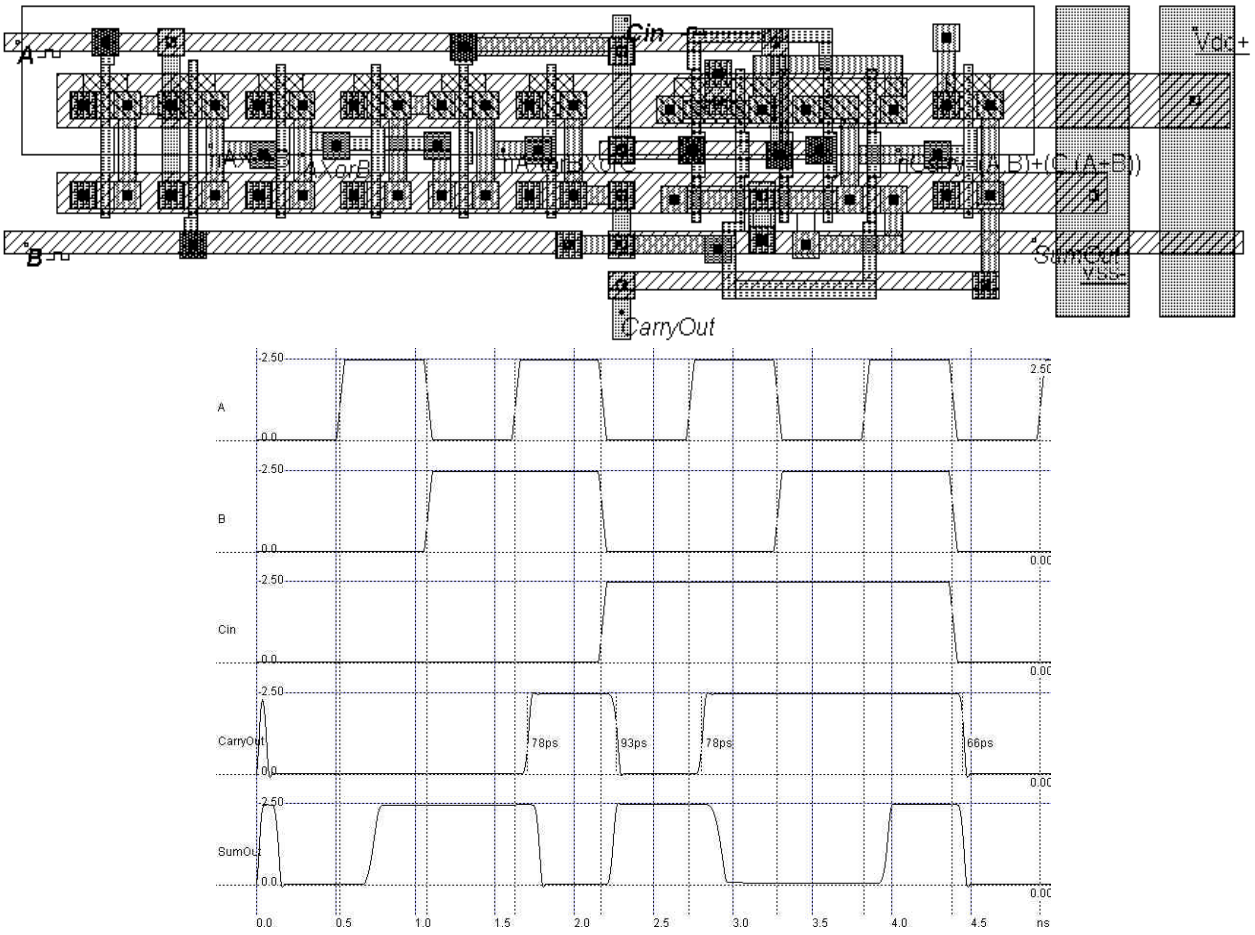


Fig. 6-5. The full-custom implementation of the full-adder and its simulation (FullADD.MSK).

You may create the layout of the full-adder by hand in order to create a compact design. Notice that the AND/OR combination of cells may be replaced by a complex gate. An example of full-custom layout of the full-adder is proposed in Figure 6-5. Notice that the carry propagates vertically within the cell to ease multiple addition. The typical delay is less than 100ps in 0.25µm technology.

```
module fulladd(sum,carry,a,b,c);
  input a,b,c;
  output sum,carry;
  wire sum1;

  xor xor1(sum1,a,b);
  xor xor2(sum,sum1,c);
  and and1(c1,a,b);
  and and2(c2,b,c);
  and and3(c3,a,c);
  or  or1(carry,c1,c2,c3);
endmodule
```

Alternatively, you may use DSCH2 to create the schematic diagram of the full-adder and compile it directly into layout. Verify the circuit with buttons and lamps. Save the deign under the name ‘fadd.sch’ using the

command **File -> Save As.** Generate the Verilog text by using the command **File -> Make Verilog File**. In Microwind2, click on the command **Compile -> Compile Verilog File**. Select the text file 'fadd.txt'. Click on **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. Click on **Simulate ->Start Simulation**. The timing diagrams appear (Figure 6-5) and you should verify the truth table of the full-adder.

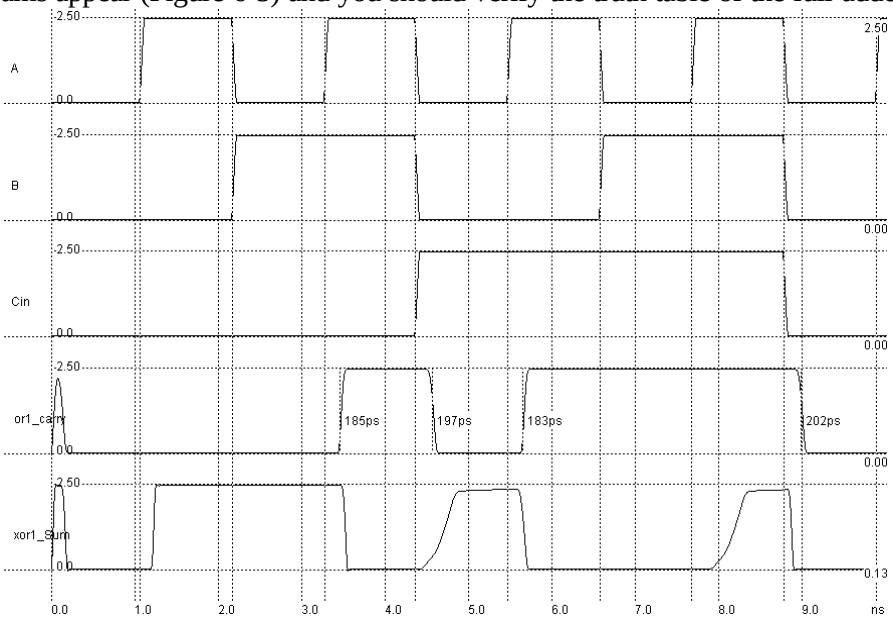


Fig. 6-5. Simulation of a full-adder (File FADD.MSK).

6.5 Four-Bit Adder

The four-bit adder circuit includes adders in serial to perform the arithmetic addition. The result of each stage propagates to the next one, from the top to the bottom. The circuit allows a four-bit addition between two numbers A3,A2,A1,A0 and B3,B2,B1,B0. Insert the user-defined 'Fadd.sym' symbol using the command **Insert -> User Symbol**. In DSCH2, the A and B numbers are generated by keyboard symbols, as reported below. Also notice the hexadecimal display with a ground connected to the K input to activate the display.

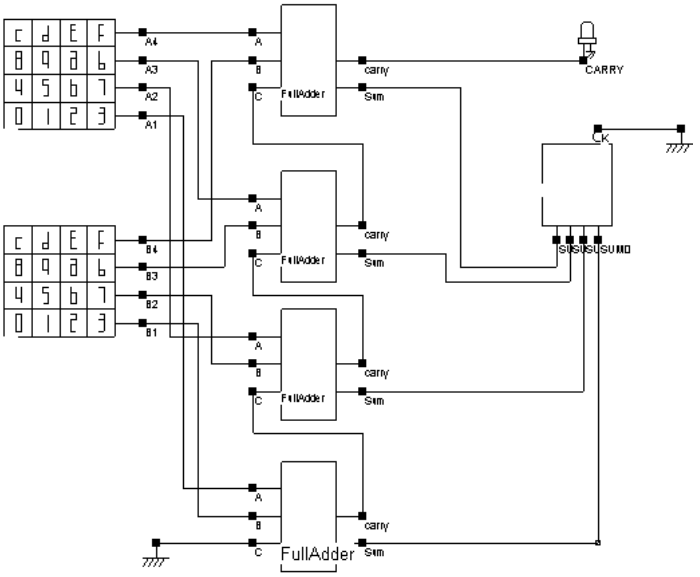


Fig. 6-6. Schematic diagram of the four-bit adder (ADD4.SCH).

Figure 6-7 details the four-bit adder layout based on the full-custom cell design, with the corresponding simulation. In Microwind2, the command **Edit -> Duplicate X,Y** has been used to duplicate the full-adder layout vertically.

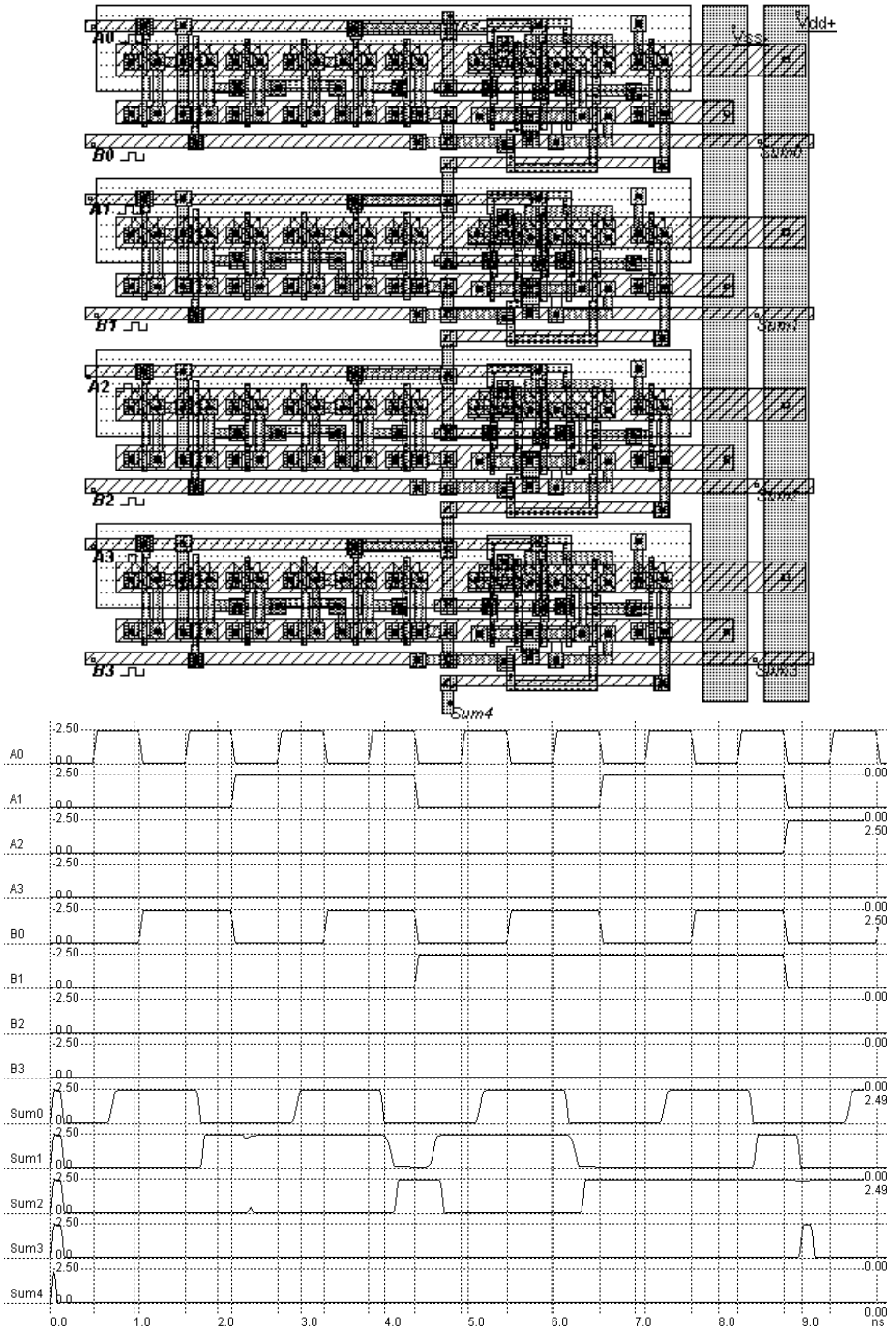


Fig. 6-7. Design and simulation of the four-bit adder (ADD4.MSK).

6.6 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

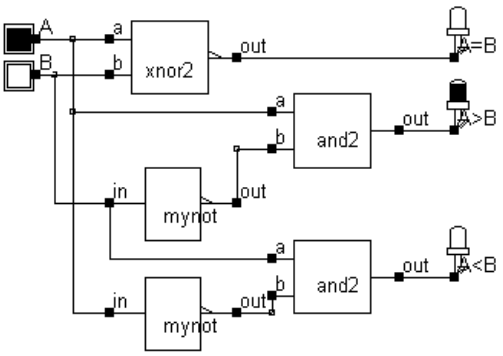


Fig. 6-8 The truth table and schematic diagram of the comparator (COMP.SCH).

Using DSCH2, the logic circuit of the comparator is designed and verified at logic level. Then the conversion into Verilog is invoked (**File -> Make verilog File**). Microwind2 compiles the verilog text into layout. The simulation of the comparator is given in Figure 6-9. 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.

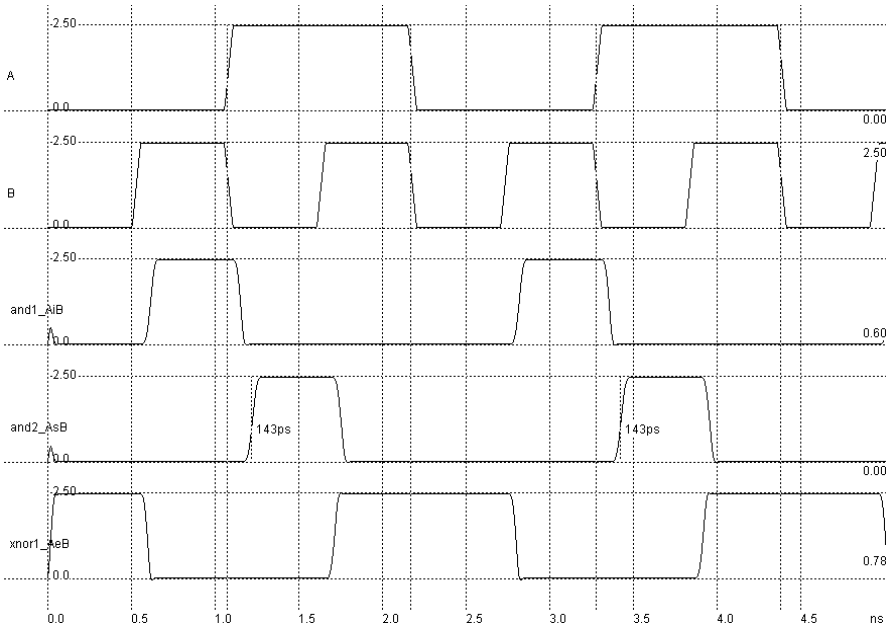


Fig. 6-9. Simulation of a comparator (COMP.MSK file).

6.7 Arithmetic and logic Units (ALU)

The digital function that implements the micro-operations on the information stored in registers is commonly called an arithmetic logic unit (ALU). The ALU receives the information from the registers and performs a given operation as specified by the control.

A very simple ALU design is proposed to illustrate its principle. The control unit is made up of a 4-1 multiplexor. The operation part consists of four kinds of operations listed as follows: and, or, addition and subtraction. The 'and' and 'or' operation are realized by using the basic logic gates. The addition and subtraction are realized using the ADDER user symbols. A digital multiplexer made from MOS devices selects one of the 4 operations results and directs it to a single output line « Result ».

S1	S0	Operation
0	0	or
0	1	and
1	0	full substraction
1	1	full addition

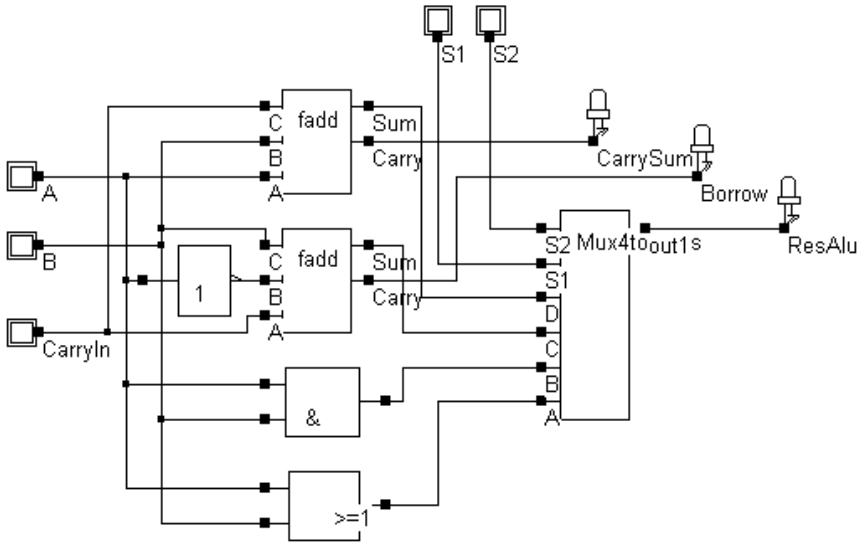


Fig. 6-10. The 1-bit ALU operates the and, or, addition and substraction (ALU1bit.SCH)

6.8 Critical Path

In DSCH2, a specific command computes the critical path, that is the path between one input and one output with the longest switching delay. In the example reported in figure 6-11, the critical path between inputs in1,in2,in3 and the outputs is marked using dot lines. The estimated maximum delay is 0.38ns (bottom output) while the other branch exhibits a delay of 0.30ns.

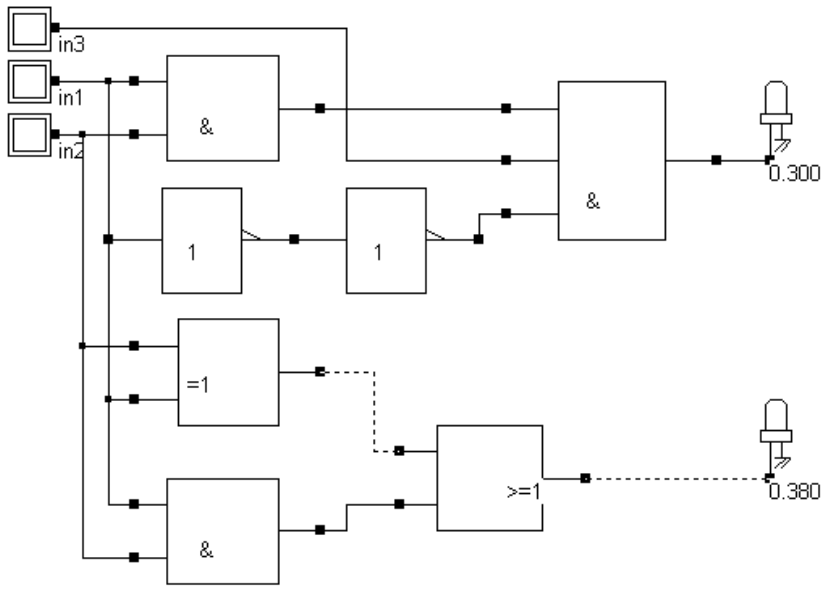


Fig. 6-11. Displaying the critical path between one input and one output.

7 Latches & Memories

This chapter details the structure and behavior of latches and memory circuits. The RS Latch, the D Latch and the edge-sensitive register are presenter. Then , the concepts of ROM, static RAM and dynamic RAM memories are introduced, together with simulations.

7.1 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 NAND gates. In that case, the Reset and Set inputs are active low. The memory state corresponds to Reset=Set=1. The combination Reset=Set=0 should not be used, as Q=nQ=1. Furthermore, the simultaneous change from Reset=Set=0 to Reset=Set=1 provokes what is called the metastable state, that corresponds to a parasitic ring effect that may jeopardize the behavior of the whole circuit.

RS Latch (NAND)

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

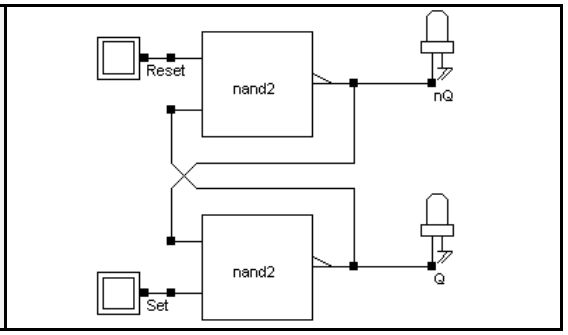


Fig. 7-1. The truth table and schematic diagram of a RS latch made (RSNor.SCH)

FULL CUSTOM LAYOUT. You may create the layout of RS latch manually. The two NAND gates may share the VDD and VSS supply achieving continuous diffusions. The internal routing may also save routing area, leading to the layout shown in Figure 6.2.
VERILOG COMPILING. Use DSCH2 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 by using the command **File -> Make Verilog File**. In Microwind2, click on the command **Compile -> Compile Verilog File**>. Select the text file

'RS.txt'. 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 "RESET" node.
2. Click the brush to clear the existing pulse properties of the pulse.
3. Enter the desired sequence, for example 01000. An click "INSERT". A piece-wise-linear sequence is generated in the table, describing the 01000 waveform in an analog way.
4. Repeat the same procedure to change the clock into a pulse for node "SET". This time the sequence must be 000100 to delay the pulse.
5. Click on **Simulate ->Start Simulation**. The timing diagrams of figure 7-2 appear. Click on **Close** to return to the editor.

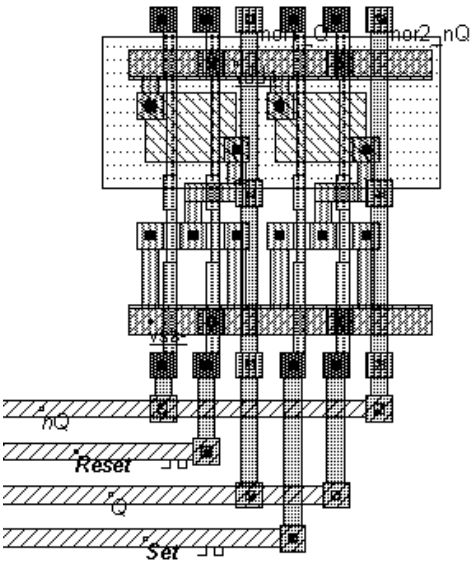


Fig. 7-2. Layout of the RS latch made (RSNor.MSK)

In the simulation of Figure 7-3, 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.

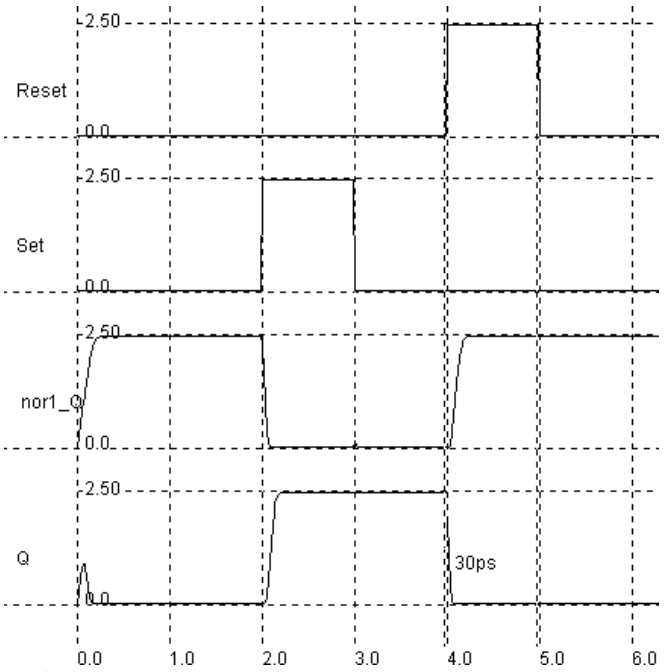


Fig. 7-3. Simulation of the RSNOR latch (RSNor.MSK)

7.2 D Latch

The truth table and schematic diagram of the static D latch, also called Static D-Flip-Flop, are shown in Figure 7-4. The data input D is transferred to the output if the clock input is at level 1. When the clock returns to level 0, the latch keeps its last value.

D Latch (NOR)

D	Clock	Q	nQ
0	0	Q	nQ
0	1	0	1
1	0	Q	nQ
1	1	1	0

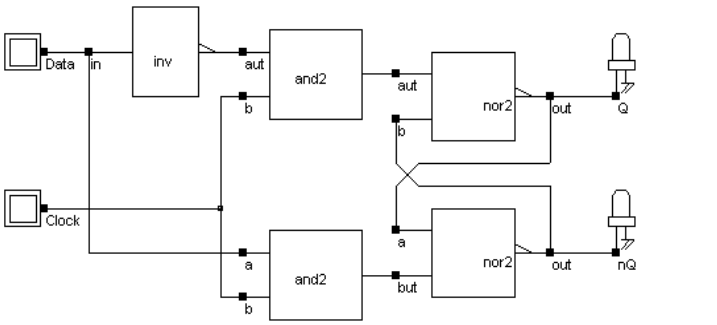


Fig. 7-4. The truth table and schematic diagram of a D Latch (File DLATCH.SCH).

MANUAL DESIGN. Note that the NOR2-AND combination can be implemented in a complex-gate style. You may find useful to invoke the one line compiler to create successively one inverter **nd=~d**, and two complex gates which include the AND/NOR cells using the syntax **Q=~(nQ|(nd&h))** and **nQ=/(Q|(d&h)**. Build the interconnections and run the Design Rule Checker.

Assign a clock to CLK and a clock to DATA. An example of such an implementation can be found in the file "DLatchLevel.MSK". Its layout an corresponding simulation are illustrated in figure 7-5.

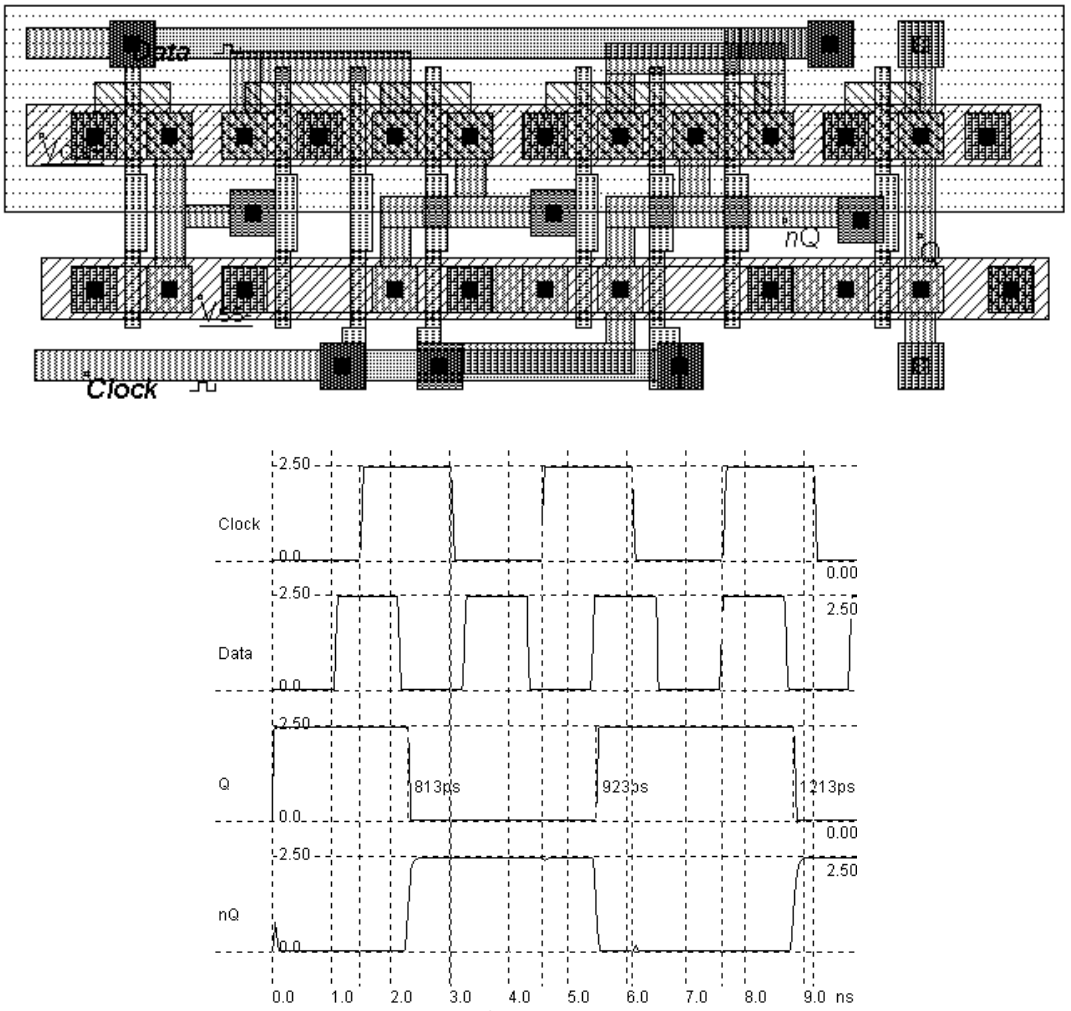


Fig. 7-5 Implementation and simulation of the D Latch (File DLatchLevel.MSK)

Edit the schematic diagram called "DLATCH.SCH" using DSCH2. Generate the Verilog text by using the command **File -> Make Verilog File**. In Microwind2, click on the command **Compile -> Compile Verilog File.** Select the text file 'DLATCH.txt'. Click on **Compile.**

7.3 Edge Trigged Latch

The most common example of an edge-trigged flip flop is the JK latch. Anyhow, the JK is rarely used, a more simple version that features the same function with one single input D is preferred. This simple type of 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. 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 pass 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.

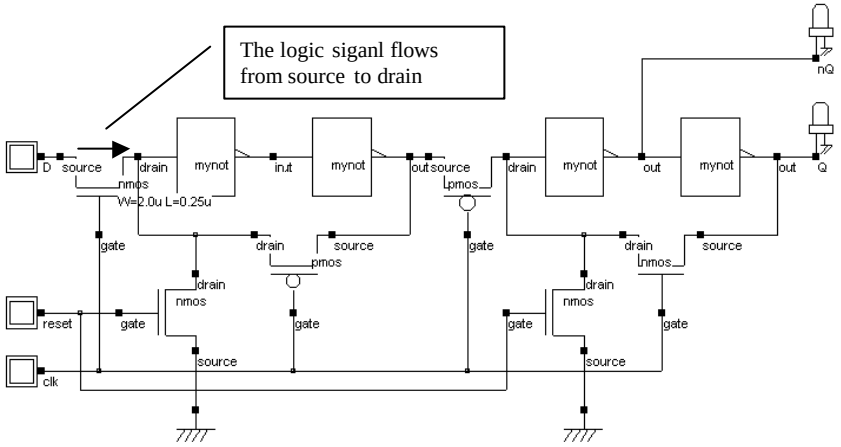


Figure 7-6 : The edge-trigged latch and its logic simulation (Dreg.MSK)

Notice that the logic model of the MOS device is not working the same way as for the real MOS switch. In the case of the logic implementation, the logic signal flows only from the source to the drain. This is not the case of the real switch where the signal can flow both ways.

Use the Verilog compiler to generate the edge-trigged latch, using the following text (dreg.txt), or by creating a schematic diagram including the "D" register symbol, in the symbol palette of DSCH2. 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 "rst" automatically.

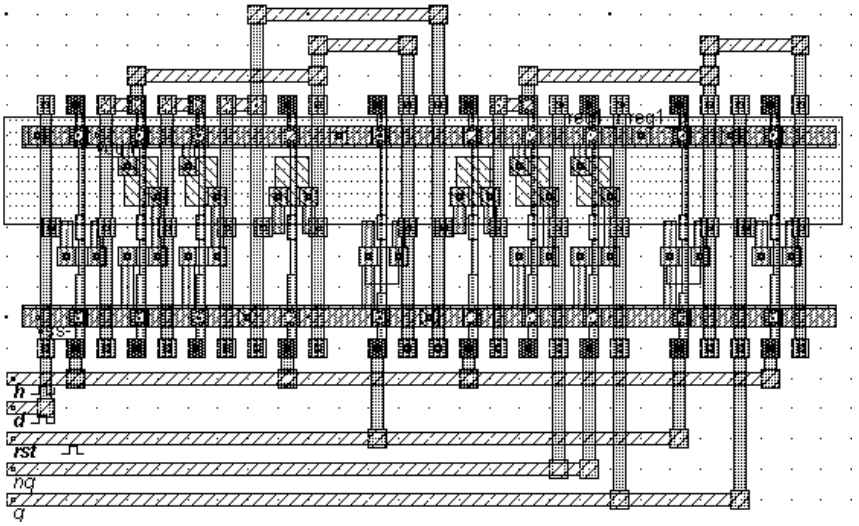


Fig. 7-7: Compiled version of the Edge-trigged D Flip Flop

The simulation of the edge-trigged latch is reported in figure 7-8. 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.

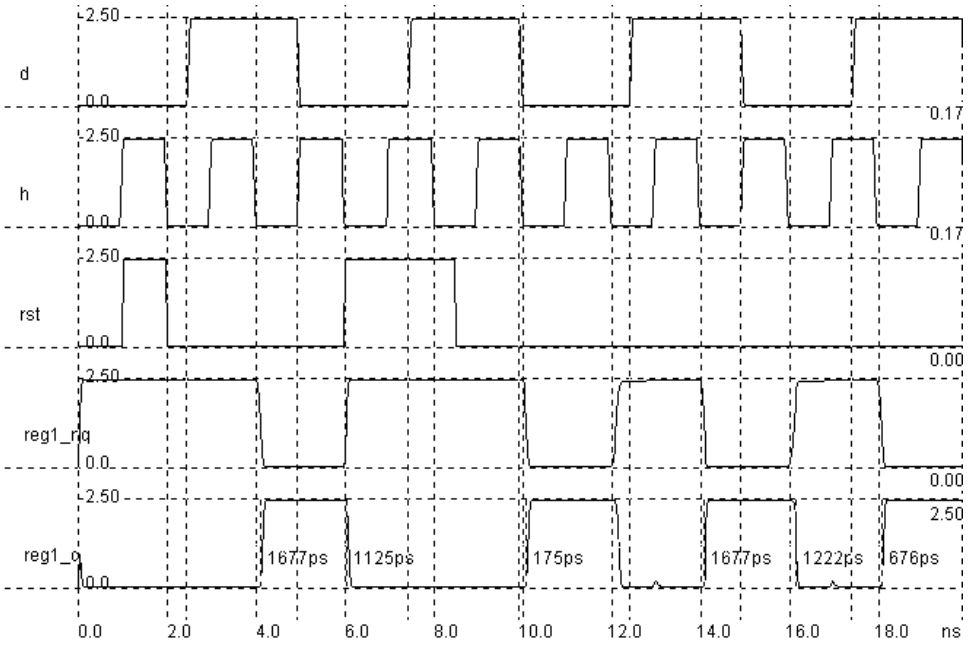


Fig. 7-8 Simulation of the DREG cell (DREG.MSK)

7.4 Counter

The one-bit counter is able to produce a signal featuring half the frequency of a clock. The most simple implementation consists of a D flip-flop where the output nQ is connected to D, as shown in figure 7-9. In the logic simulation shown in figure 7-9, the clock ”Clock1” changes the state of “Clock_Div_2” at each fall edge. The “RESET” is active high, and stuck the output to 0.

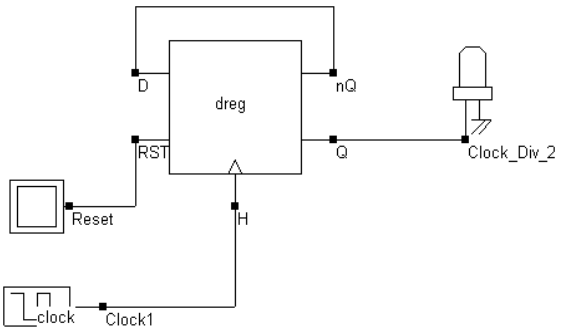


Fig. 7-9. Schematic diagram of the 2-bit counter (DivFreq.MSK).

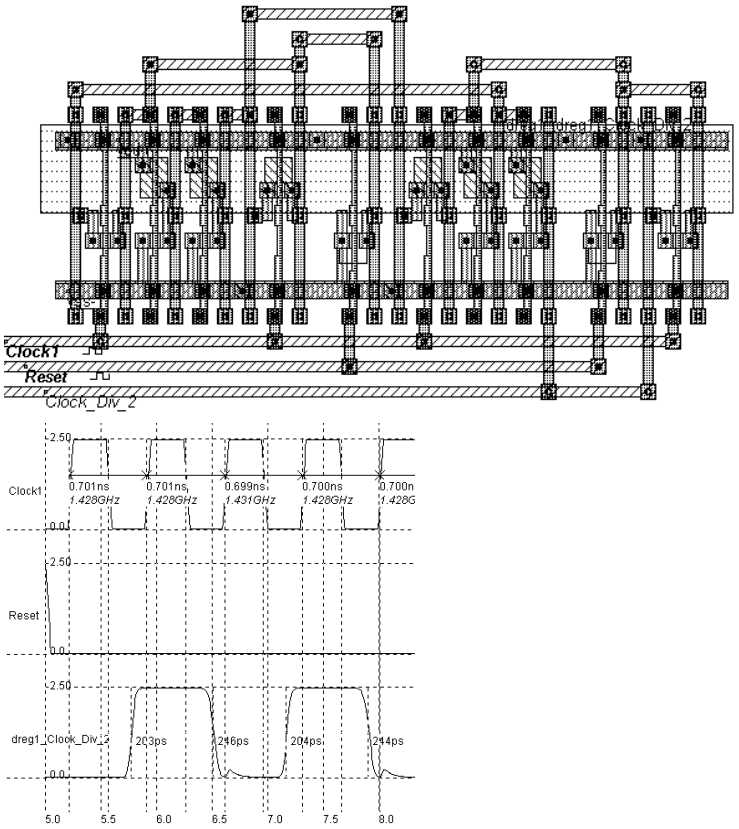


Fig 7-10. Layout and analog simulation of the divider-by-two (ClockDiv2.MSK)

7.5 RAM Memory

The schematic diagram of the static memory cell used in High Capacity Static RAMs is given in figure 7-11. The circuit consists of 2 cross-coupled inverters and two nMOS pass transistors. 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 x 256 cells or higher. An arrangement of 4x4 RAM cells is also shown in figure 6-14. The selection line **Sel** concerns all the cells of one row. The lines **Data** and **nData** concern all the cells of one column.

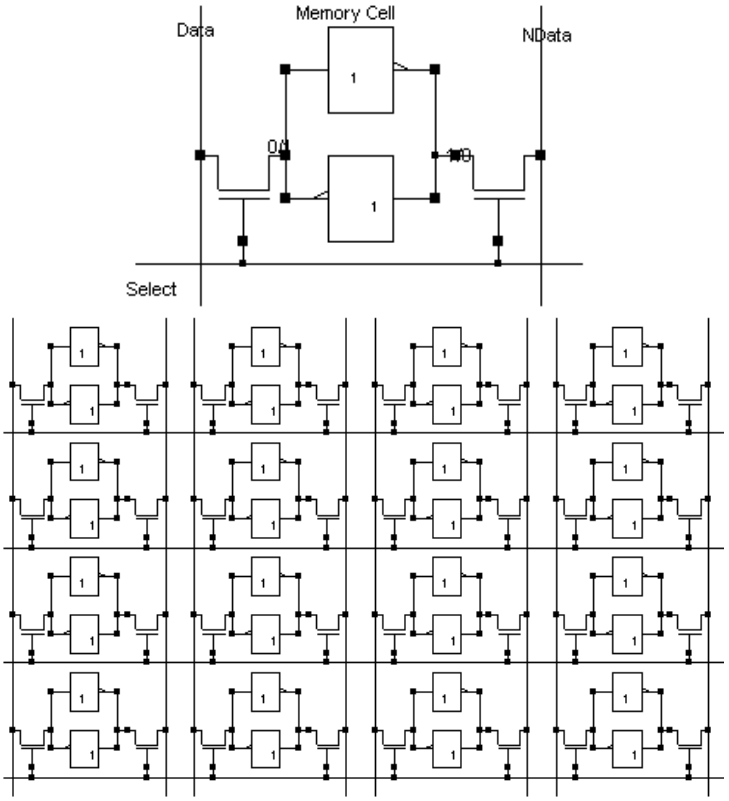


Fig. 7-11 The schematic diagram of the static RAM cell (RAM1.SCH).

The RAM layout is given in Figure 7-12. Click on **File → Open → RAM.MSK** to read it. The **Data** and **nData** 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. The cross-section shows the nMOS devices and the connection to VSS using metal3, situated on the middle of the cell. The Data and nData lines, in metal2 are on both sides.

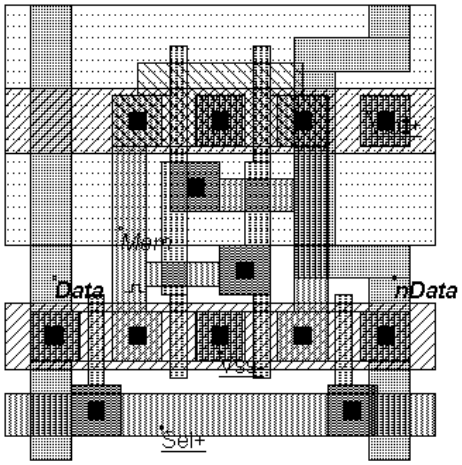


Fig 7-12. The layout of the static RAM cell (RAM1.MSK).

WRITE CYCLE. Values 1 or 0 must be placed on **Data**, and the data inverted value on **nData**. Then the line **Sel** goes to 1. The two-inverter latch takes the Data value. When the line **Sel** returns to 0, the RAM is in a memory state. See figure 7-13 for the analog simulation of the WRITE cycle.

READ CYCLE. In order to read the cell, the line **Sel** must be asserted. The RAM value propagates to **Data**, and its inverted value propagates to **nData**.

SIMULATION. The simulation parameters correspond to the write cycle in the RAM. The simulation steps describe din figure 6-16 are as follows:

- ❶ **Mem** reaches 1, after an unstable period (unpredicatable value).
- ❷ **Data** gets to value 0 and **nData** to value 1.
- ❸ **Sel** is asserted. The memory cell **Mem** goes down to 0.
- ❹ **Data** gets to a value of 1 and **nData** gets to a value of 0.
- ❺ **Sel** is still asserted. The memory cell fights against Data=1 and surrenders (Mem=1).
- ❻ **Sel** is inactive. The RAM is in a memory state.

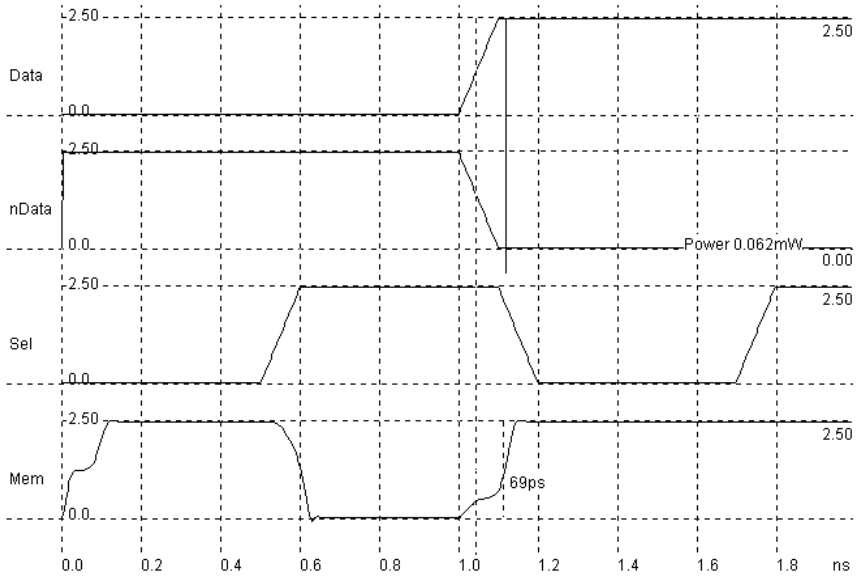


Fig. 7-13. Write cycle for the static RAM cell (RAM1.MSK).

7.6 RAM Array

You can duplicate the RAM cell into a 4x4 bit array using the command **Edit -> Duplicate XY**. Select the whole RAM cell and a new window appears. Enter the value « 4 » for X and « 4 » for Y into the menu. Click on « **Generate** ». The result is shown below.

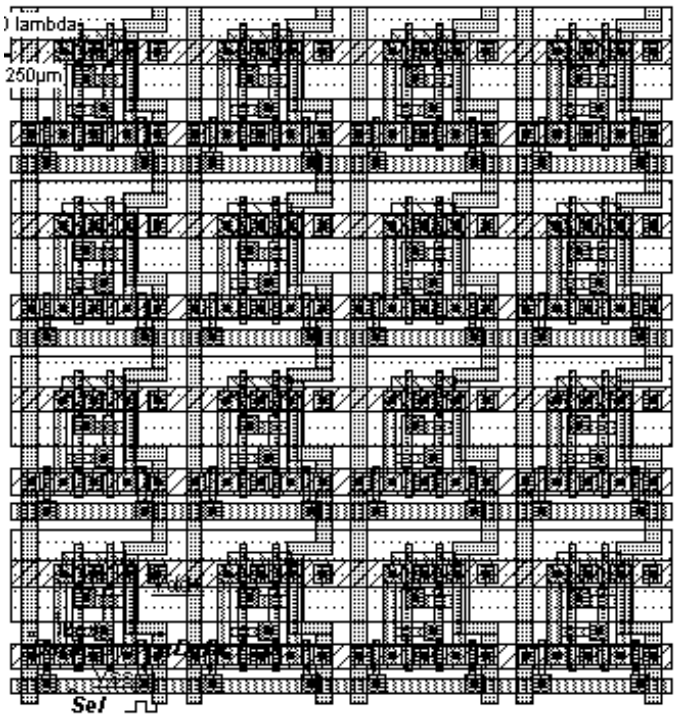


Fig. 7-14 Duplicating the RAM Cell in X and Y

7.7 RAM Line decoder

The line decoder is based on the following schematic diagram. One line is asserted while all the other lines are at zero. In this circuit one line was picked out from a choice of four lines. Using AND gates would be an easy solution, but in order to save the inverter, we choose NOR gates with inverted inputs.

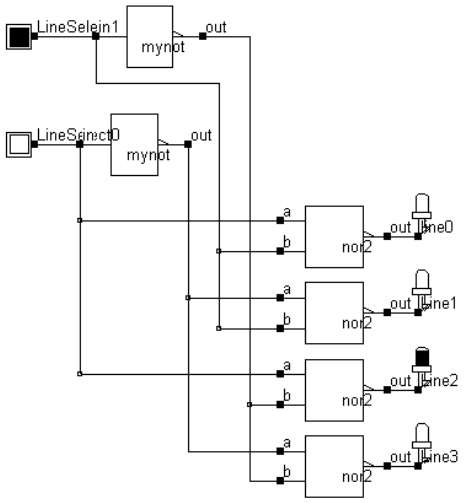


Fig. 7-15. A line selection circuit

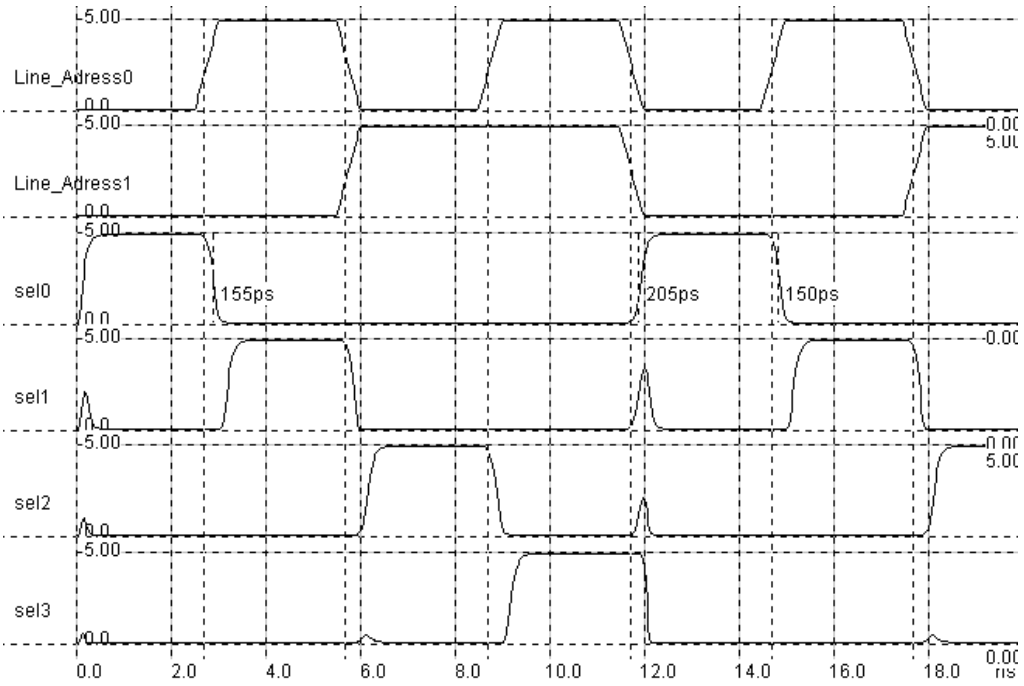


Fig. 7-16 A line selection layout and its corresponding simulation (RamLineSelect.MSK)

The NOR gate height should be adjusted to that of the RAM cell height. When making the final assembly between blocks, the command **Edit -> Move Area** is very important. This command helps to move a selected block with a lambda step.

7.8 RAM Column Selection

The column selection circuit is based on the same principles as those of the line decoder. The major modification is that the data flows both ways, that is firstly from the cell to the read circuit (Read cycle) and secondly from the write circuit to the cell (Write cycle). Fig. 7-17 proposes an architecture for this.

The n-channel MOS device is used as a switch controlled by the column selection. When the n-channel MOS is on and **Write** is asserted, the data issued from DataIn is amplified by the buffer, flows from the bottom to the top and reaches the memory. If **Write** is off, the 3-state inverter is in high impedance, which allows one to read the information.

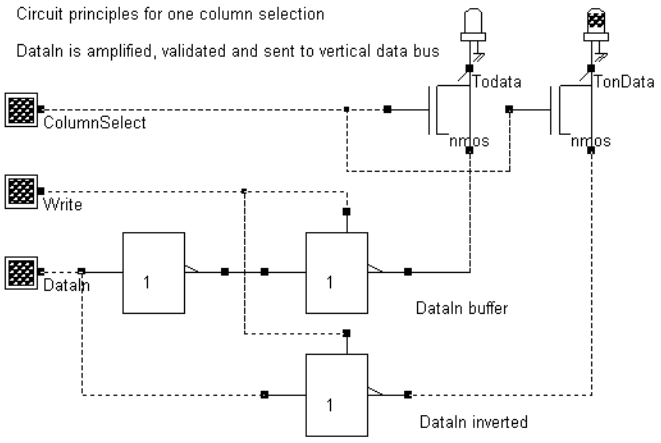


Fig. 7-17. Row selection and Read/Write circuit (RamColumn.SCH)

7.9 Dynamic RAM Memory

The dynamic RAM memory uses a single MOS device with a parasitic junction capacitance as a storage element. In figure 7-18, a set of 4x4 dynamic RAM cells are reported. The gates are connected horizontally while the drains are connected vertically. Furthermore, the sources are connected to a polarization node.

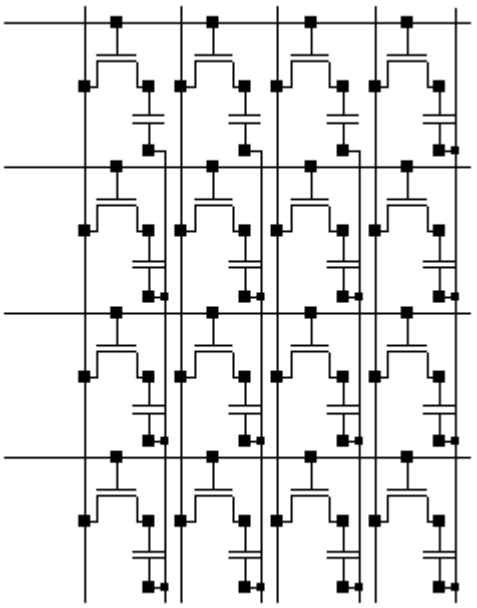


Fig. 7-18. An array of 4x4 dynamic cells (Dram4x4.SCH)

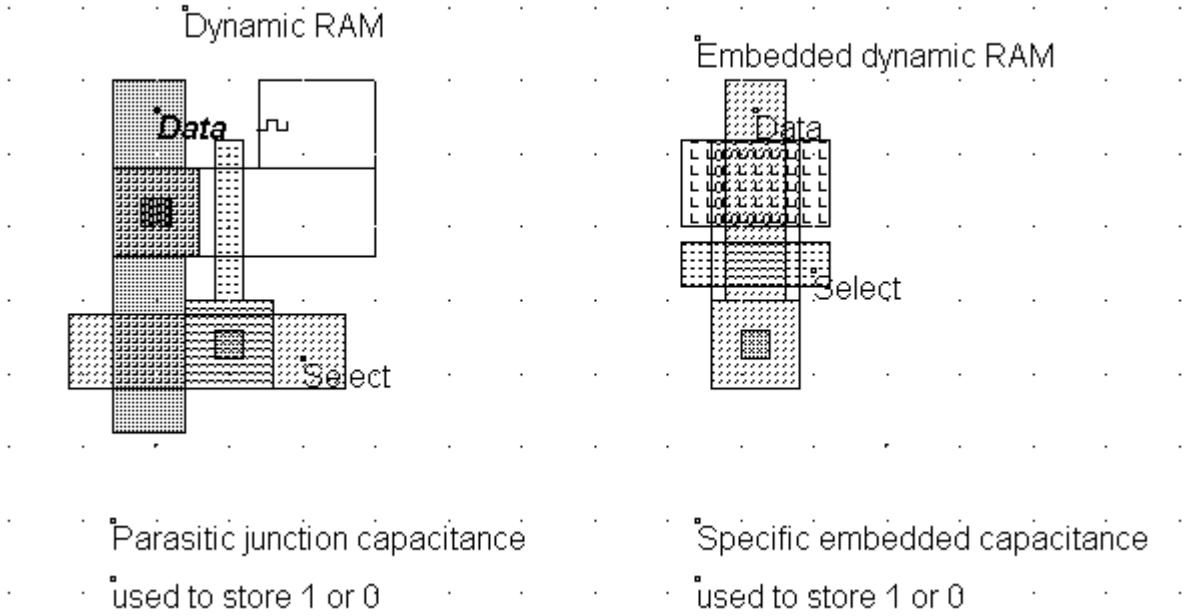


Fig. 7-19. The dynamic ram cell may be a single MOS device with enlarged source region or an embedded capacitance.

In figure 7-19, two layout implementation are proposed. In the left part, the source area of the MOS is significantly enlarged to increase artificially the parasitic junction capacitance. A typical target capacitance value C_{cell} for large DRAM arrays is 3fF. In this layout, C_{cell} is 0.2fF. In the layout of figure 7-19 right, a specific option layer is added, with a property of embedded capacitance with a very high value.

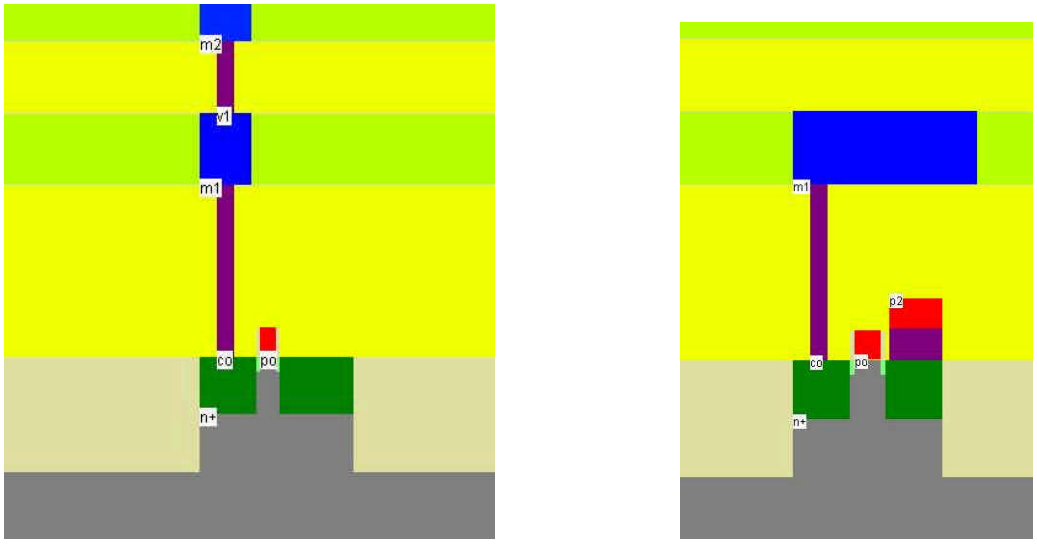


Fig. 7-20. Compared to the regular MOS (left), the embedded DRAM uses poly2 and a second specific layer (violet) to create an efficient capacitor.

Thanks to a specific option layer available in embedded RAM process, C_{cell} is increased to 3fF, while reducing the cell area (Figure 7-21 right). Three 8x8 DRAM arrays have been created by duplicating (Command Edit Duplicate XY in Microwind) the basic DRAM cell. See the impact of embedded capacitor on the reduction of silicon area. Also notice that joining two cells at the common contact saves silicon area. The fixed voltage of the embedded capacitor is driven by the 2nd layer of polysilicon.

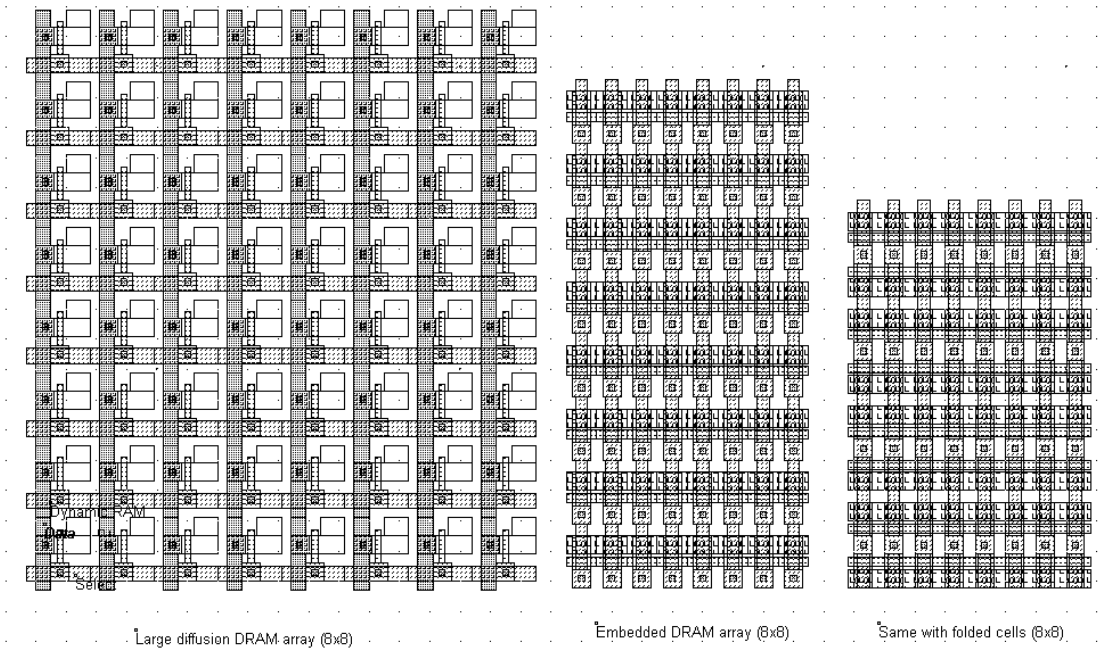


Fig. 7-21. Impact of embedded capacitor on silicon area on a 8x8 array

7.10 EEPROM

The EEPROM memory includes 2 poly gates, with the bottom polysilicon floating, isolated by oxide (Figure 7-22) . The programming of a double-poly transistor involves the transfer of electrons from the source to the floating gate through the thin oxide. The erasure involves the transfer of electrons from the floating gate back to the source.

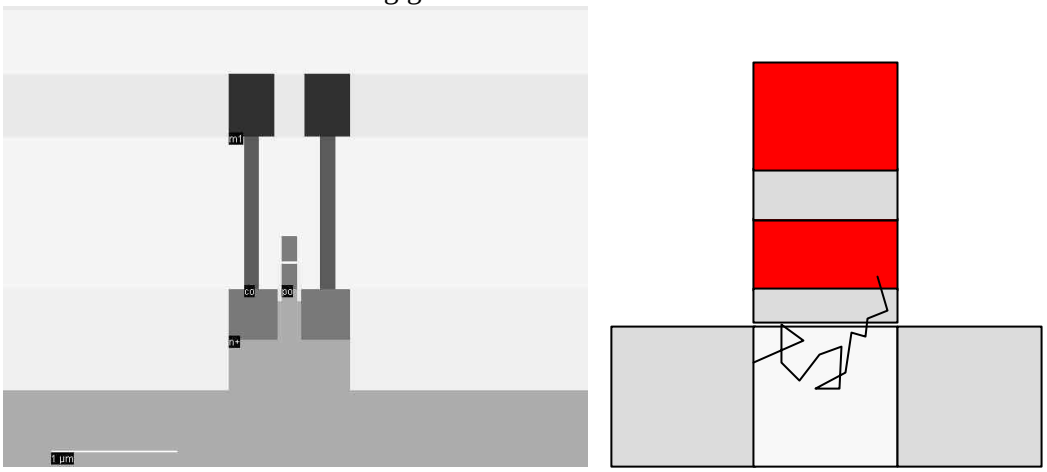


Fig. 7-22: The double-poly MOS device used as a non-volative memory

The programming operation is performed using a very high gate voltage on poly2, usually around 10V. The mechanism for electron transfer from the grounded source to the floating polysilicon gate is called tunneling. With a sufficiently positive voltage on the poly2 gate, the voltage difference between poly and source is high enough to enable electrons to pass through the thin oxide. The electron transfer mechanism is called hot electron injection.

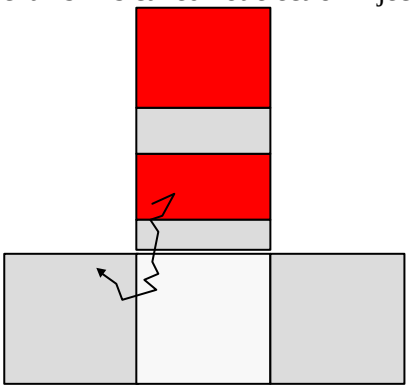
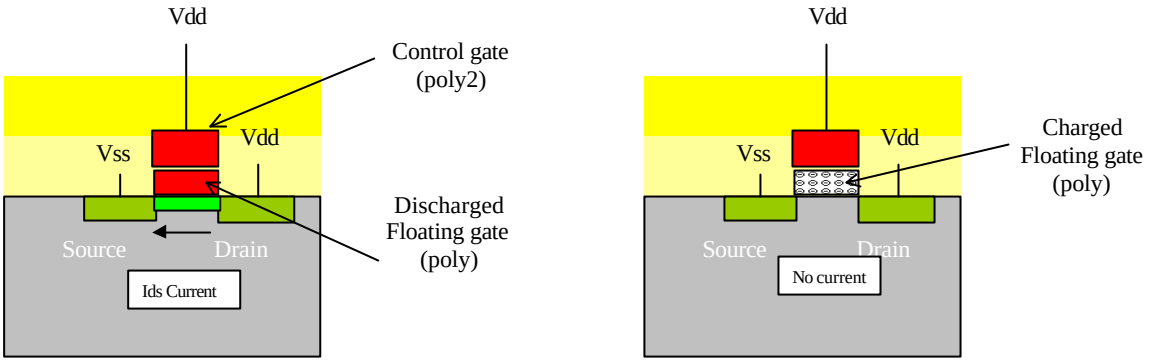


Fig. 7-23: Erase operation to remove electrons

For the erase operation (Figure 7-23), the poly2 gate is grounded and a high voltage (Around 10V) is applied to the source. Electrons are pulled off the floating gate thanks to the high electrical field between the source and the floating gate. This charge transfer is called Fowler-Nordheim electron tunneling.

From an operational point of view, the double-poly MOS device works as a normal MOS when the floating gate is discharged: a VDD gate voltage is high enough to turn the device on. Consequently, a Ids current flows between the drain and grounded source. If the floating gate is charged with electrons, the threshold voltage is very high, and a VDD gate voltage is not sufficient to turn the MOS on. Almost no Ids current is flowing.



(a) Discharged gate, low Vt , small current (b) Charged gate, high Vt, no current

Fig. 7-24: Read operation with a double-poly MOS device

.

8 Analog Cells

8.1 Diode-connected MOS

The schematic diagram of the diode-connected MOS is proposed in figure 8-1. The question rises: is this device a capacitance, a diode or a resistance? The answer is: a capacitance for Vk<Vt, a diode with an interesting high resistance when Vk>Vt, where Vt is the threshold voltage of the device. The main application of this circuit is the design of a big resistance in a very small silicon area.

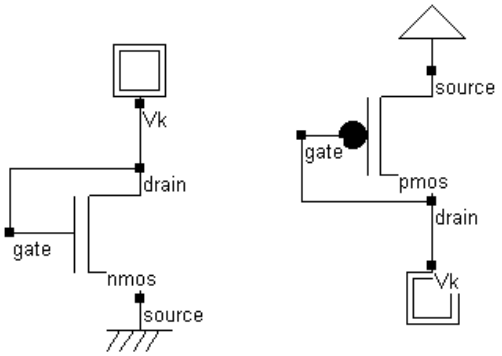


Figure 8-1 : MOS connected as a diode

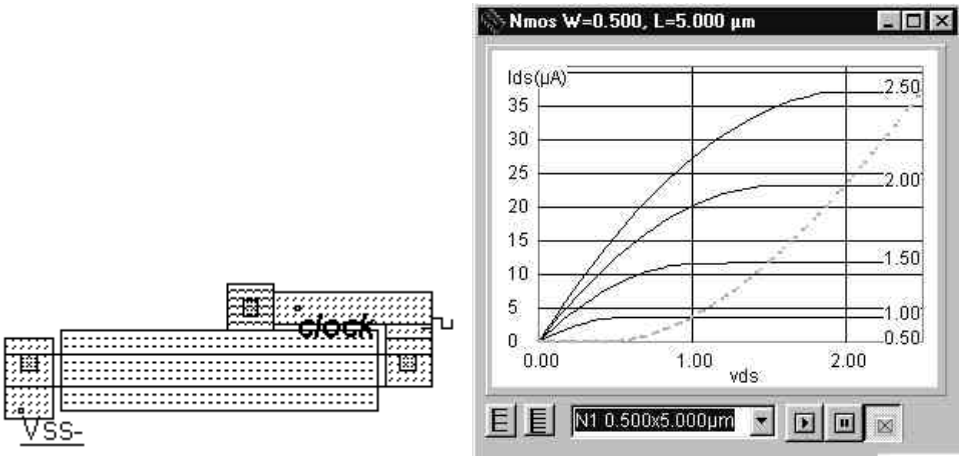


Figure 8-2 : Layout and simulation of the MOS connected as a diode

❶ In the palette, click the icon "MOS generator".

❷ Enter a large length and a small width. For example, enter W=0.5µm, L=5µm. This sizing corresponds to a long and narrow channel, featuring a very high resistance channel with poor current performances.

❸ 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.

❹ Click **Simulation on Layout.** In a small window, the MOS characteristics are drawn, with the functional point drawn as a color dot (Figure 7-2). It can be seen that the I/V characteristics correspond to a diode. The resistance varies with Vk but can be estimated around 30KΩ)

The resistance obtained using such a circuit can reach easily 100KΩ in a very small silicon area. The same resistance can be drawn in poly but would require a much larger area.

8.2 Voltage Reference

The voltage reference is usually derived from a voltage divider made from resistance. The main problem is that the value of the resistance must be high to keep the short cut current low, to avoid wasted power consumption. A key idea is to use MOS devices rather than polysilicon or diffusion resistance to keep silicon area very small.

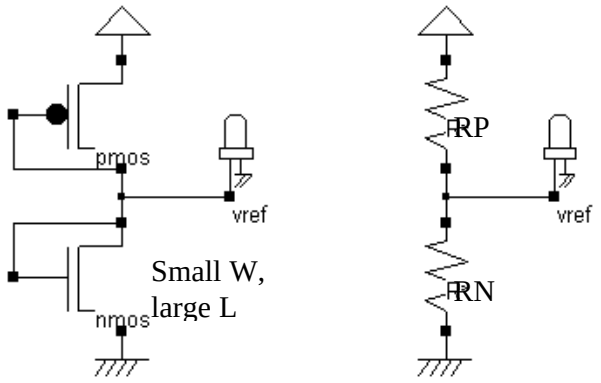


Figure 8-3 : Voltage reference using PMOS and NMOS devices as large resistance

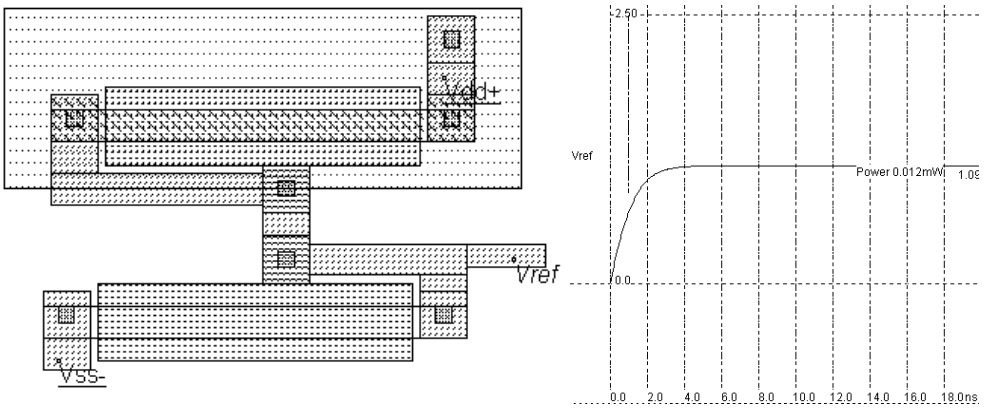


Figure 8-4 : Voltage reference of 1V (Vref.MSK)

In the layout of figure 8-5, the PMOS and NMOS have the same size. Due to lower PMOS mobility, the resulting Vref is not VDD/2 but 1V. You may change the temperature (Simulate -> Simulate Options) and see how the voltage reference is altered by temperature.

8.3 Current Mirror

The current mirror is one of the most useful basic blocs in analog design. In its most simple configuration, it consists in two MOS devices, as presented in figure 8-5. A current I1 flowing through the nMOS device Master is copied to the MOS device Slave. If the size of Master and Slave are identical, in most operating conditions, the currents are the same. The remarkable point is that the current is almost independent of the drain voltage of the slave V2. If the ration W/L of the Slave is 10 times the ratio of the Master, the current on the right branch is 10 times the current on the left branch.

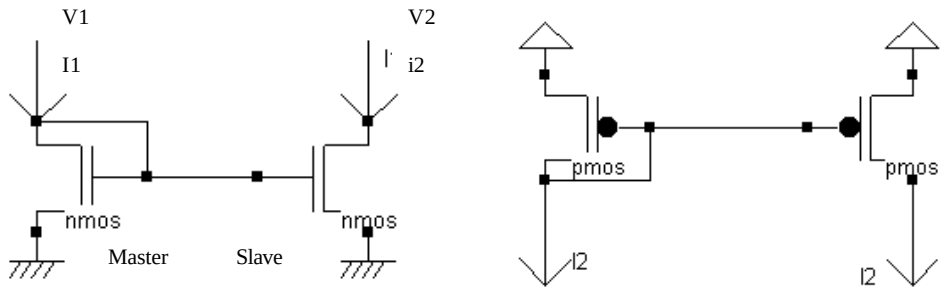


Figure 8-5: Current mirror principles in nMOS and pMOS versions

The illustration of the current mirror behavior is performed on the layout of figure 8-6. The circuit includes a voltage reference, using N1 and P1 as described above, a device N2 which has an identical size as N1, and a device N3 with L=0.5µm, leading to a ratio equal to 10 W/L of N1. What we expect is a current I2 equal to I1 in most operating conditions and a current I3 =10xI1.

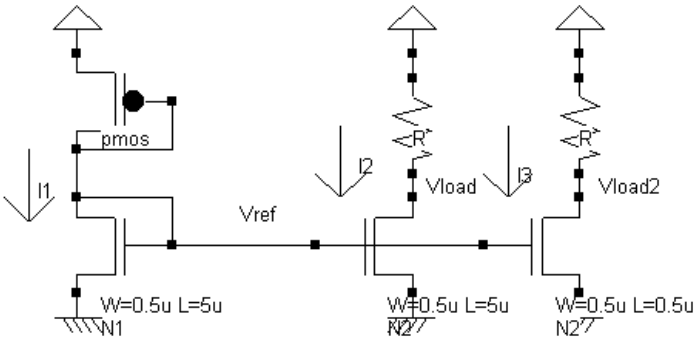
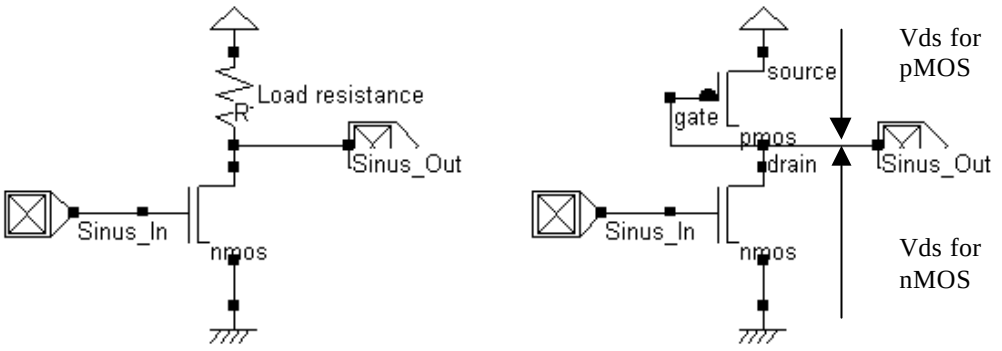


Figure 8-6: Illustration of the current mirror principles (Mirror.MSK)

You may observe each MOS characteristics by using the command **Simulate ® Simulate on Layout**. During the transient simulation, the functional point of the select MOS device appears in the characteristics, which provides a valuable aid to understand the current mirror behavior. For most values of Vload, N2 produces the same current as N1, except when Vload is lower than 0.5V.

8.4 Single Stage Amplifier

The single-stage amplifier is described in figure 8-7. It consists of a MOS device (we choose here a n-channel MOS) and a load resistance. The resistance can be made from polysilicon or diffusion. As the gain of this amplifier is proportional to the load resistance, a MOS device with gate and drain connected, as shown in figure 8-7 could replace the resistance. This is called an active resistance. Using a small silicon area, high resistance can be obtained, meaning high amplifier gains.



(a) (b)

Figure 8-7: Single stage amplifier with passive resistance (a) and active resistance (b).

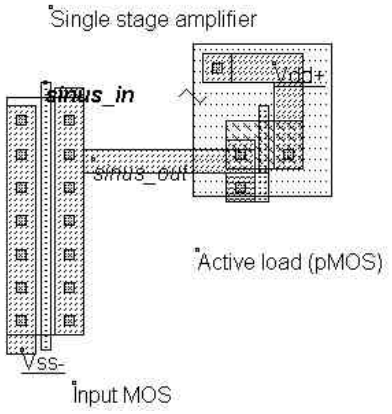


Figure 8-8: Single Stage amplifier layout (AmpliSingle.MSK)

In the simulation window, click “Voltage vs voltage” and “More”, to compute the static response of the amplifier (Figure 8-9). The range of voltage input that exhibits a constant gain appears clearly. For V_{DS} higher than 0.6V and lower than 0.8V, the output gain is around 5. Therefore, an optimum offset value is 0.7V. Change the parameter "Offset" of the input sinusoidal wave to place the input voltage in the correct polarization.

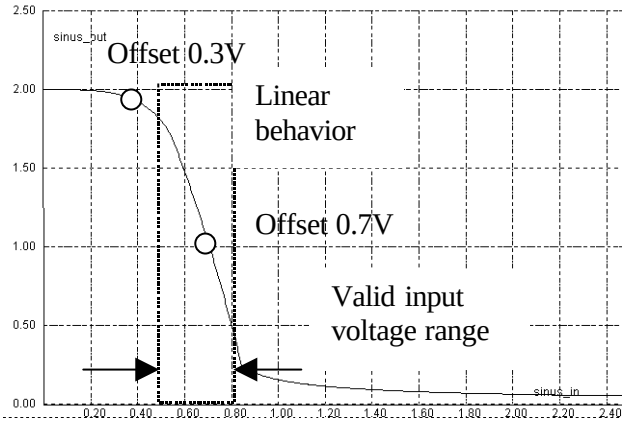


Figure 8-9: Single Stage amplifier static response

We change the sinusoidal input offset and start again the simulation. A gain of 5 is observed as predicted from the static simulation when the offset is 0.8V (input 100mV peak to peak, output 482mV peak to peak).

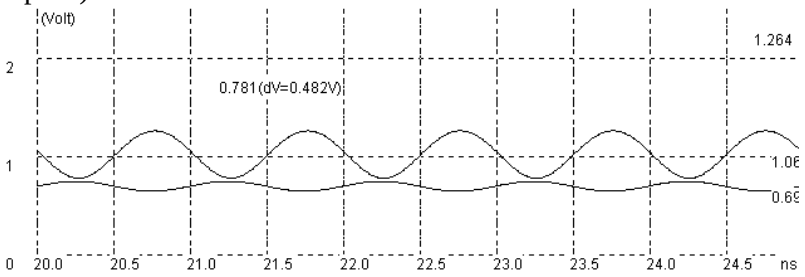


Figure 8-10: Single Stage amplifier with high gain

8.5 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. 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.

$$Vout = K(Vp - Vm)$$

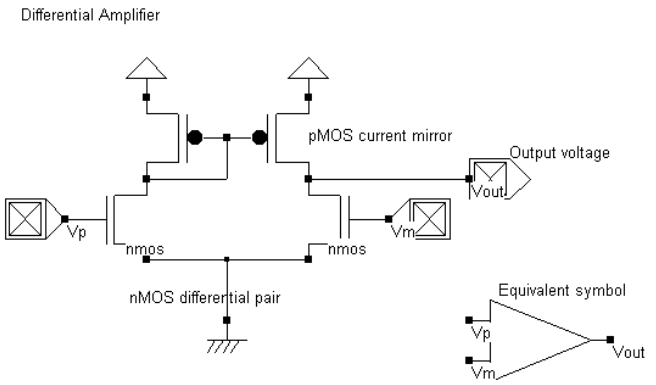


Figure 8-11: Schematic diagram of the differential amplifier

The differential amplifier layout is reported in figure 8-12. The differential pair is built from n-channel MOS devices. Their size must be identical, and drawn with the same orientation, to minimize the offset generated by transistor mismatch. In the simulation, it can be seen that a small voltage difference between V+ and V- induces the saturation of the output either near VSS and VDD.

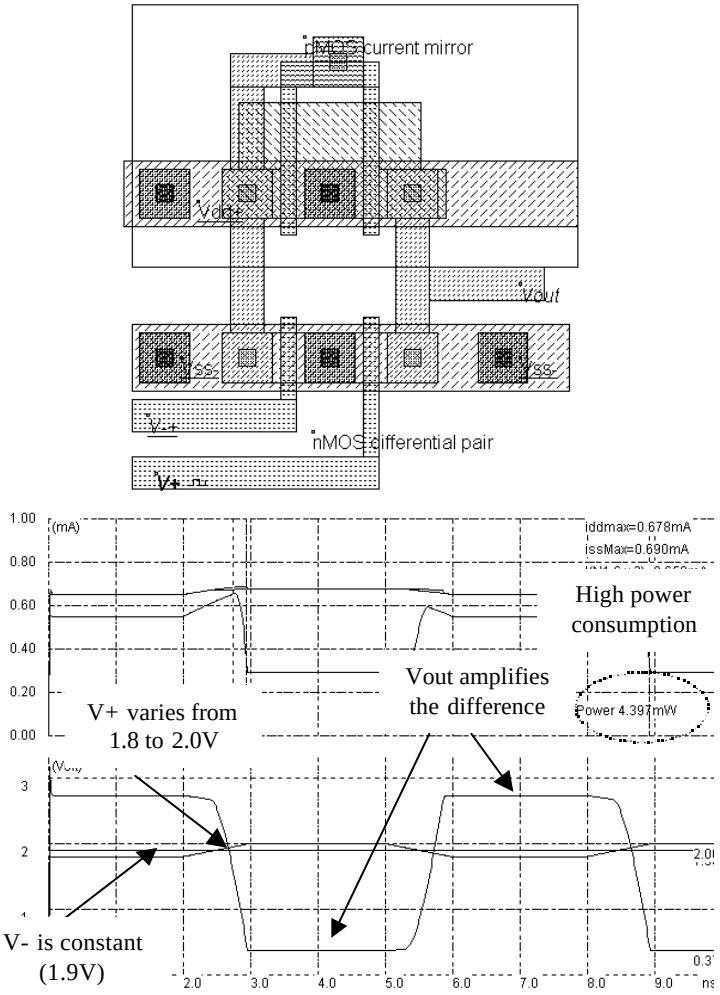


Figure 8-12: Layout and transient simulation of the differential amplifier (AmpliDiff.MSK)

MEASURE THE GAIN

1. Click « Voltage vs. Voltage » to select static characteristics mode.
2. Select « Slope » in the « Evaluate » menu.
3. Click « More » to compute the static characteristics of the differential amplifier. The value of the gain is added on the simulation window.

MEASURE THE INPUT RANGE

The best way to measure the input range is to connect the differential amplifier as a follower, that is Vout connect to V-. In this case, a slow ramp is applied on the input V+ and the result is observed on the output. The valid input range [0.5,1.9V] is the value of V+ for which the output copies the same voltage in a reasonable time.

Wide-Range Amplifier

The wide-range amplifier is built using a voltage comparator and a power output stage. Its schematic diagram is reported in Fig. 8-13. The difference between V+ and V- is amplified and it produces a result, codified : Vout. The gain near 2.5V is very high (more than 100). Use the **Voltage vs. Voltage** simulator mode to get the transfer characteristics Vout/V+.

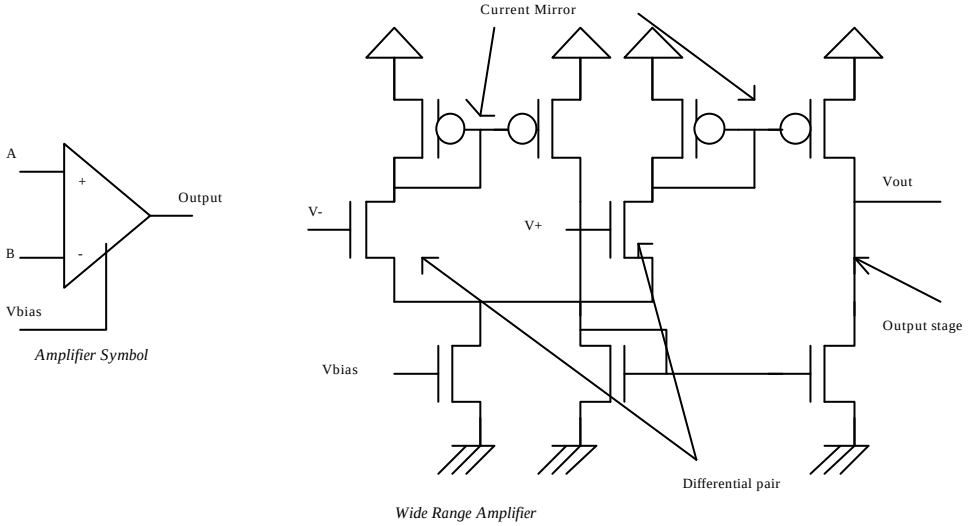


Fig 8-13. Node description and schematic diagram of the analog amplifier (AMPLI2.MSK).

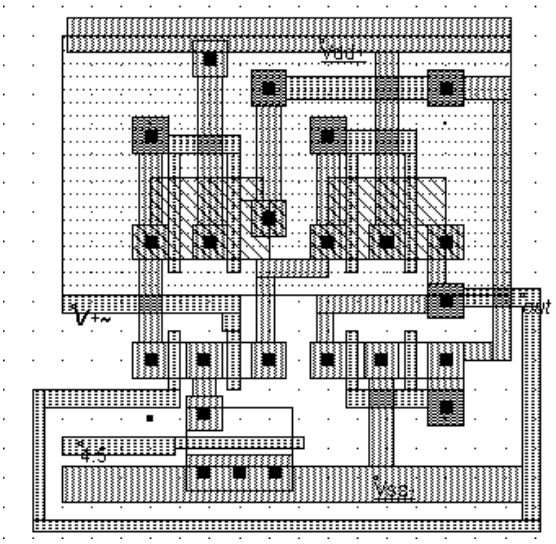


Fig. 8-14. Design of the analog amplifier (AMPLI2.MSK).

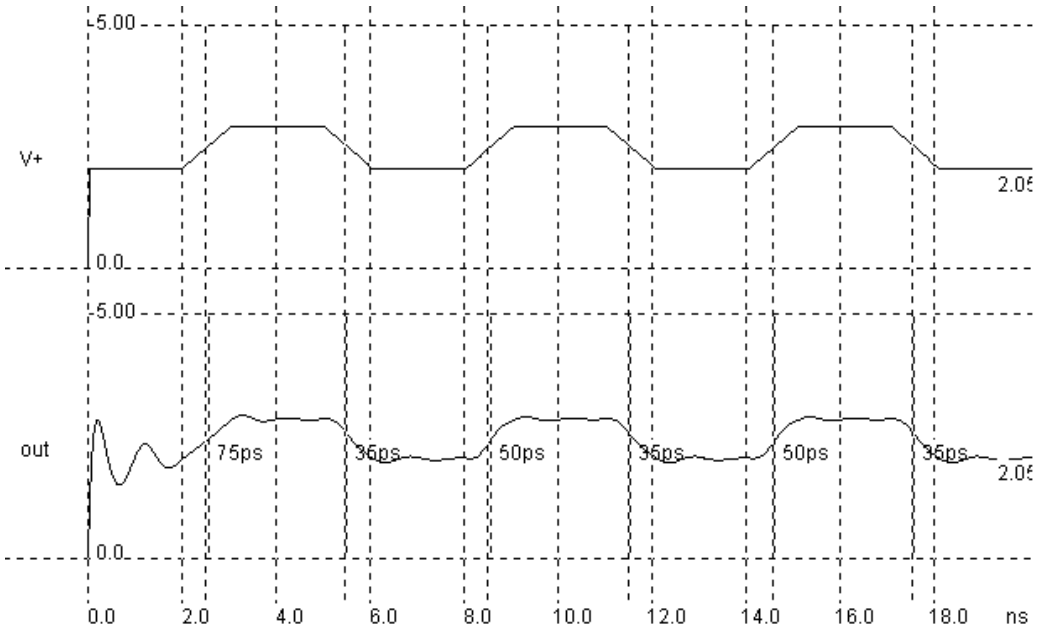


Fig. 8-15. Transient simulation of the analog amplifier (AMPLI2.MSK) connected as a follower

You can easily build a follower by designing an extra connection from Vout to V-. This layout is shown in Figure 8-14. The output stage is not strong enough to be able to drive large loads such as output pads.

8.6 Voltage Controlled Oscillator

The voltage controlled oscillator is able to produce a square wave with a frequency varying depending on an analog control Vc. Ideally, the frequency dependence with Vc should be linear. One example of voltage controlled oscillator is given in figure 8-16. It consists of a ring oscillator with three stages. Vc acts on the resistance of the supply path, which acts on the speed response of the inverters.

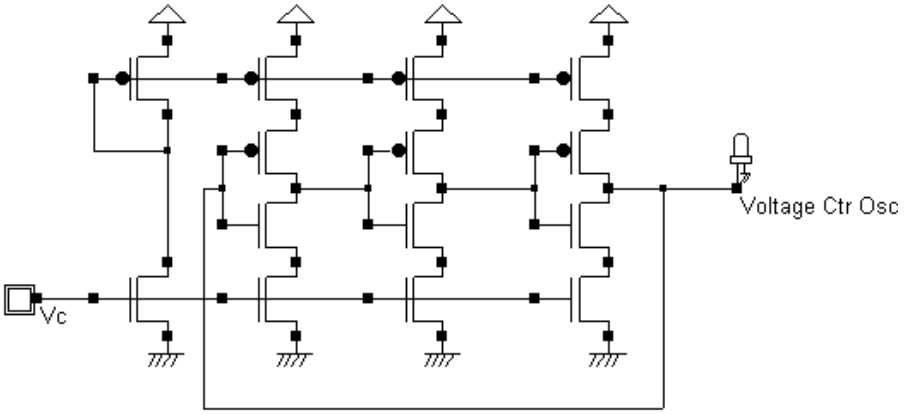


Fig. 8-16. Schematic diagram of a voltage controlled oscillator

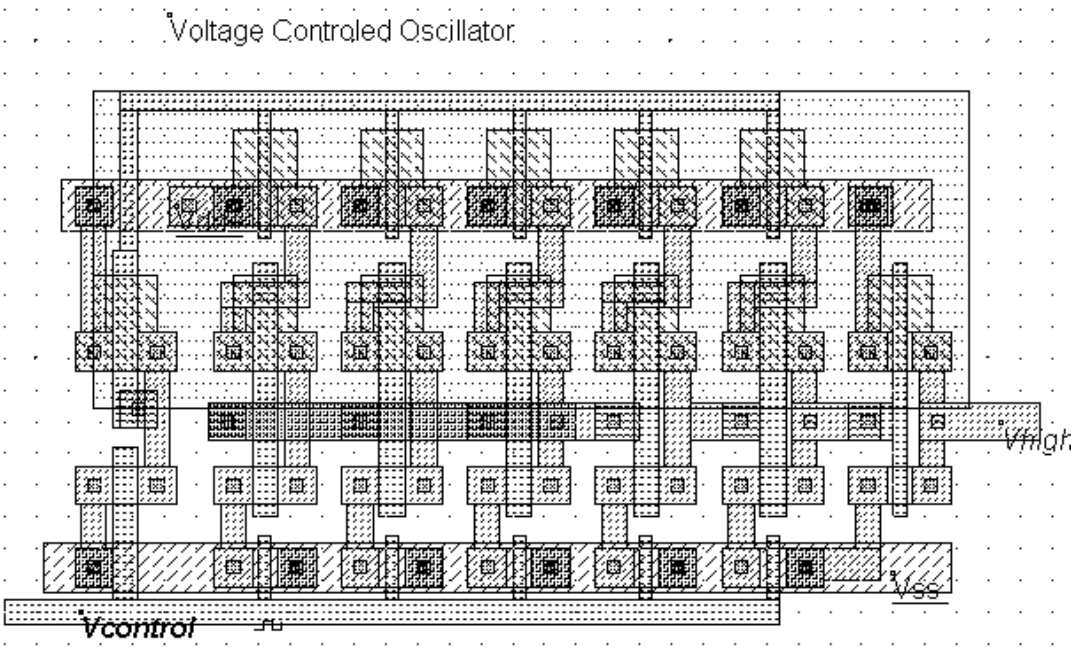


Fig. 8-17. Implementation of a voltage controlled oscillator based on a 5-stage ring oscillator

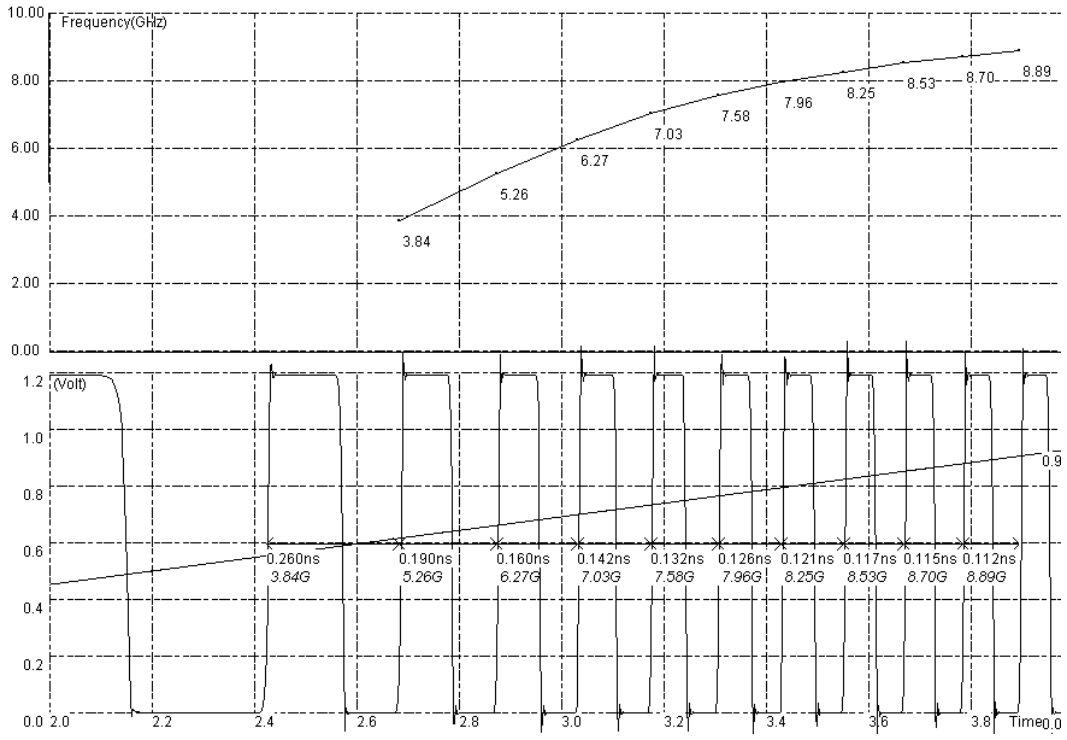


Fig. 8-18. Simulation of the voltage controlled oscillator

In the simulation of figure 8-18, we use the specific mode "Frequency and Voltages" to plot the frequency variation with Vc. The VCO output is a frequency-varying square wave. Its dependence with Vc is not linear.

9 Converters

9.1 Analog-Digital Converter

The analog-digital converter converts an analog value **Vin** into a two-bit digital value called A0,A1. The flash converter uses three converters and a coding logic to produce A0 and A1 (Figure 9-1). A very complex logic circuit and 255 comparators would be used for an ADC eight-bit flash.

The polysilicon has a high resistance (50 per square) and can be used as a resistor network (Left of Figure 9-2), which generates intermediate voltage references used by the voltage comparators located in the middle. The resistance symbol is inserted in the layout to indicate to the simulator that an equivalent resistance must be taken into account for the analog simulation. Open-loop amplifiers are used as voltage comparators. The comparisons address the decoding logic situated to the right and that provides correct A0 and A1 coding.

Analog Input Vin	**C0**	**C1**	**C2**	**A1**	**A0**
Vin<1.25V	0	0	0	0	0
1.25<Vin<2.5V	1	0	0	0	1
2.5<Vin<3.75V	1	1	0	1	0
Vin>3.75	1	1	1	1	1

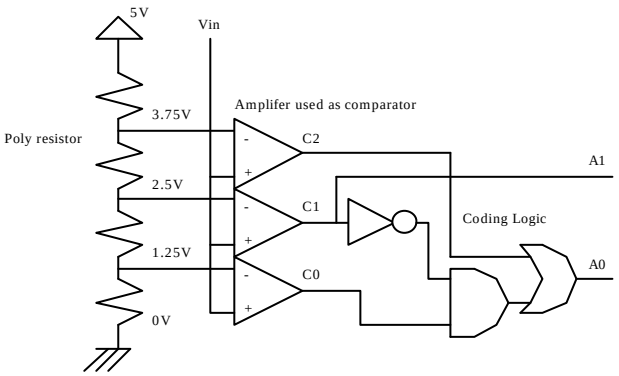


Fig. 9-1. Node description and schematic diagram of the analog-digital converter (ADC.MSK).

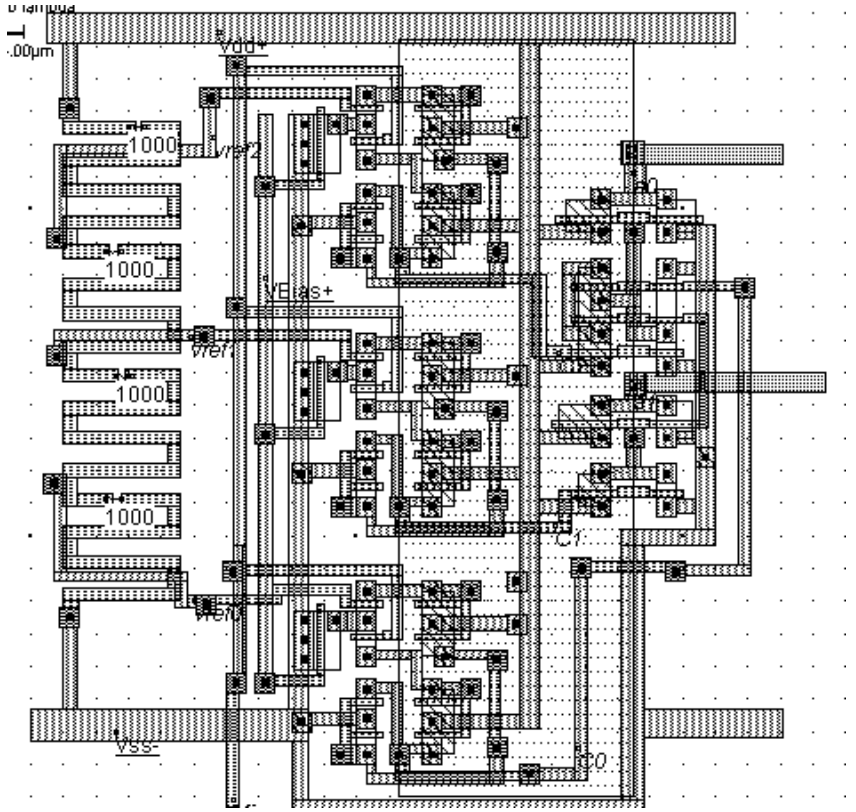


Fig. 9-2. Design of the analog-digital converter (ADC.MSK).

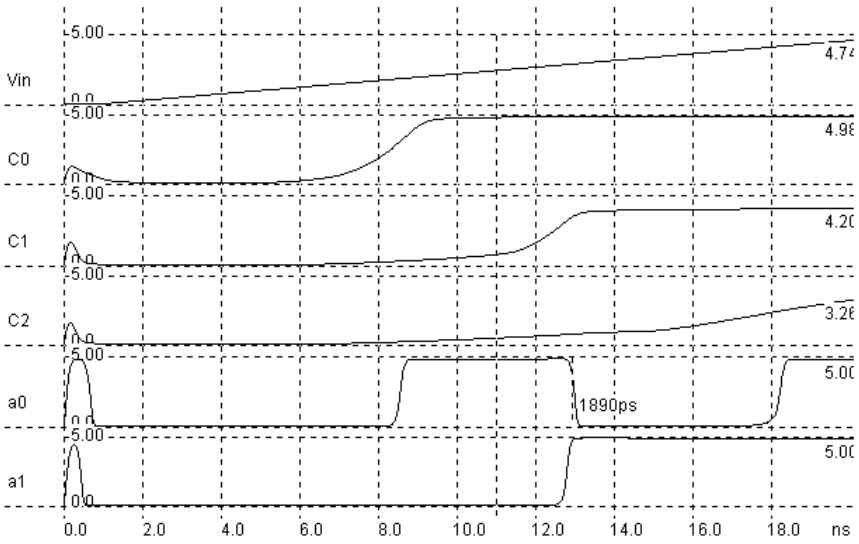


Fig. 9-3. Simulation of the analog-digital converter (ADC.MSK).

In the simulation shown in Figure 9-3, the comparators C0 and C1 work well but the comparator C2 is used in the upper limit of the voltage input range. The generation of combinations 00,01 and 10 is correct.

9.2 Digital-Analog Converter

The digital-analog converter converts a digital three-bit input (A0,A1,A2) into an analog value **Vout**. The polysilicon resistive net gives intermediate voltage references which flow to the output via a transmission gate net. The resistance symbol is inserted in the layout to indicate to the simulator that an equivalent resistance must be taken into account for the analog simulation. The schematic diagram and layout of the digital-analog converter are shown in Figure 9-4.

A2	A1	A0	Analog output Vout (V)
0	0	0	0.0 V
0	0	1	0.625
0	1	0	1.25
0	1	1	1.875
1	0	0	2.5
1	0	1	3.125
1	1	0	3.75
1	1	1	4.375

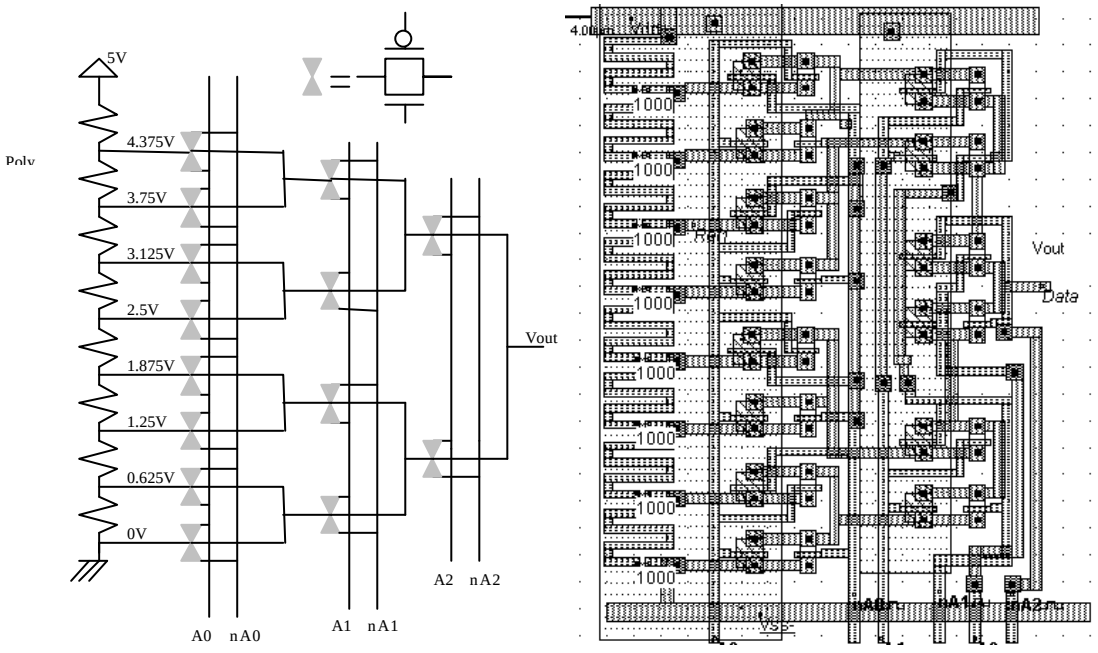


Fig. 9-4. Schematic diagram and implementation of the digital-analog converter (DAC.MSK).

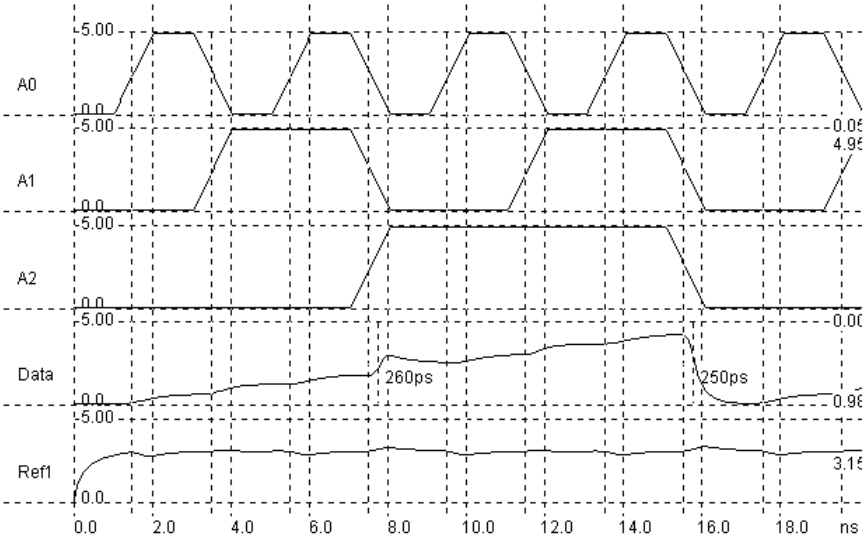


Fig. 9-5. Simulation of the digital-analog converter (DAC.MSK).

The simulation of the DAC (Fig. 9-5) shows a regular increase of the output voltage **Vout** with the input combinations, from 000 (0V) to 111 (4.375V). Each input change provokes a capacitance network charge and discharge.

9.3 Sample and Hold circuit

During the conversion from analog to digital, the input signal must be kept constant. This operation is called sample-and-hold. The transmission gate can be used as a sample and hold circuit. The layout of the transmission gate is reported below.

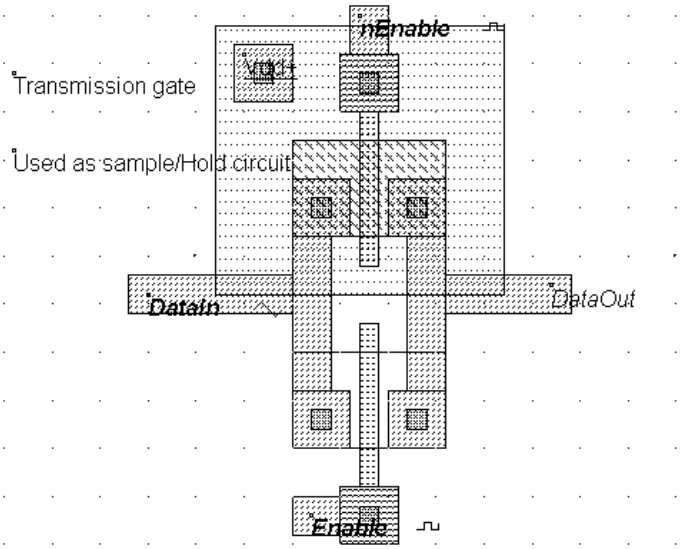


Fig. 9-6. The transmission gate used to sample analog signals (SampleHold.MSK)

The effect of sample and hold is illustrated in figure 9-7. When sampling, the transmission gate is turned on so that the sampled data DataOut reaches the value of the sinusoidal wave DataIn. When the gate is off, the value of the sampled data remains constant. This is mainly due to the parasitic capacitance of the node, which is the order of several fF.

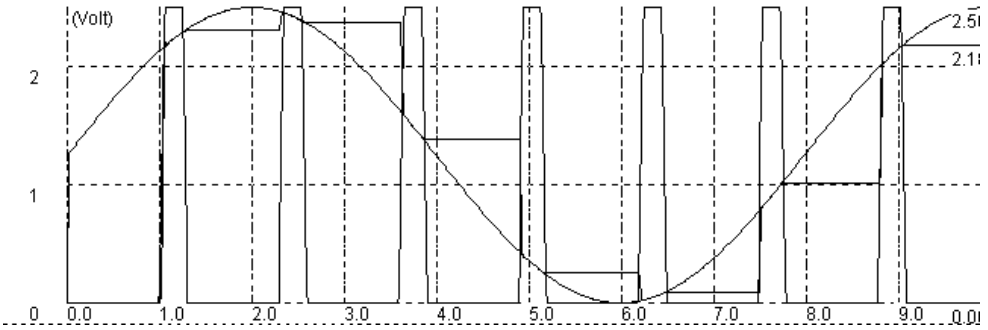


Fig. 9-7. Effect of sampling on a sinusoidal wave (SampleHold.MSK)

10 Input/Output Interfacing

10.1 Create a Pad Ring

Click on the chip library icon and click on **Pads**. Enable the "pad ring" option. A pad ring with 3 pads in X and 3 pads in Y is generated by a click on **Generate Pad**. In that case, a set of pads is added to your circuit. The VSS pad is situated at the bottom, and the VDD pad at the top with the associated power rings.

10.2 VDD/VSS Floor-planning

The supply network of a typical integrated circuit is shown in figure 10-1. Bars of metal wires cross the circuit to supply the active parts of the circuit. The metal wires are designed very large to enable strong currents to flow within the supply interconnects.

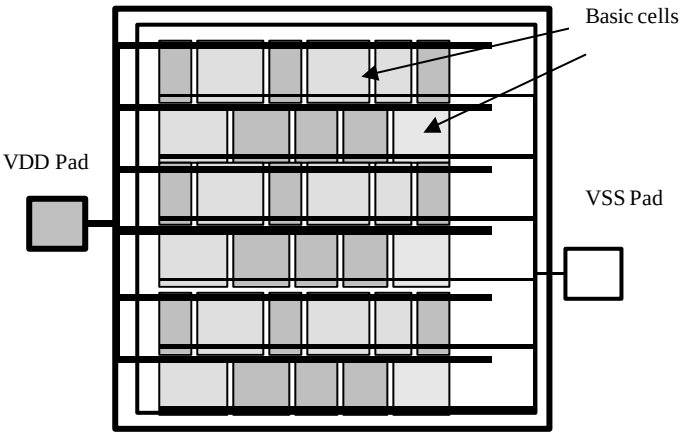


Fig. 10-1. Supply of an integrated circuit

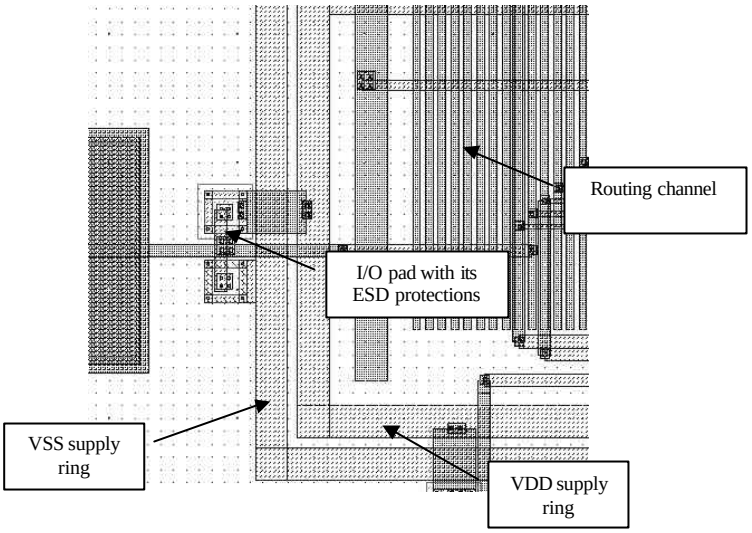


Fig. 10-2: Supply network in a real case circuit

10.3 High Voltage MOS

For interfacing with input/output, specific high voltage MOS are introduced. These MOS devices are called high voltage MOS. They use a thick gate oxide to handle the high voltage of the I/Os. An example of high voltage MOS device is reported below.

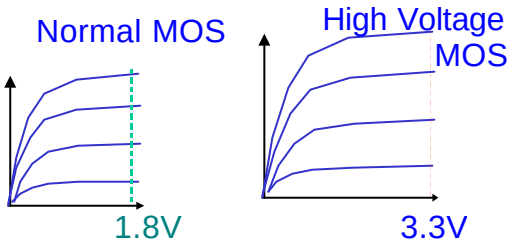


Fig. 10-3: The High voltage MOS device is used in 3.3V I/Os

The high voltage MOS uses a gate width which is slightly larger than the one of the regular MOS. As the high voltage MOS device is generally used in I/O structures, the MOS width is usually large, even sometimes very large (100-500µm).

10.4 I/O Pad

We give here some details about input-output pad structure. The basic bonding pad size is 100x100µm. The pad consists of a sandwich of **metal** layers. For advanced technologies, all metal layers are stacked on the top of each other. The passivation oxide has been removed from over the pad, so that a gold connection can be fixed upon it.

The input-output pad contains one input stage with a polysilicon resistor and two protection diodes. The output stage contains a chain of inverters. The last stage is a 3-state inverter so that the buffer can be turned off.

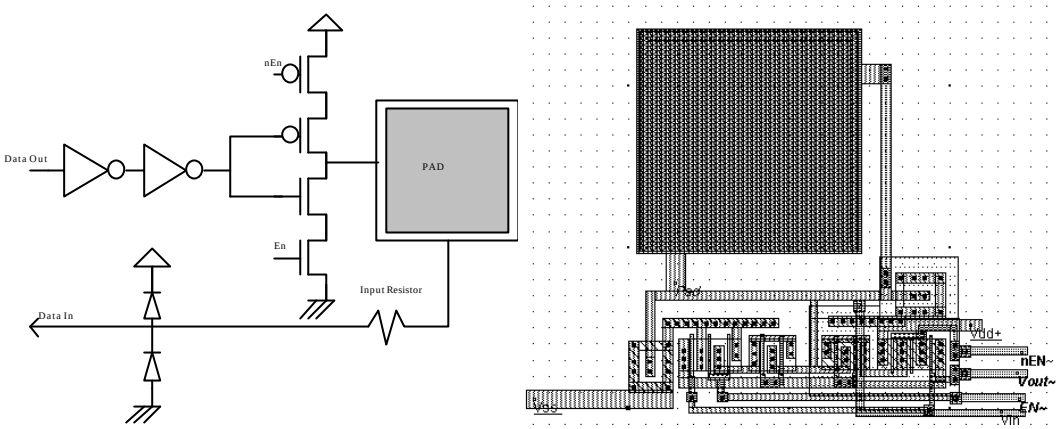


Fig. 10-4. Design of an input-output pad (PAD.MSK)

10.5 ESD Protections

The input pad includes some voltage boosting and under voltage protections linked with problems of electrostatic discharge (ESD). Such protections are required as the oxide of the gate connected to the input could be destroyed by over voltage. One of the most simple ESD protection is made up of a set of two diodes and a resistance (Fig. 10-5). One diode handles the negative voltage flowing inside the circuit (N+/P substrate), the other diode (P+/N well) handles the positive voltage.

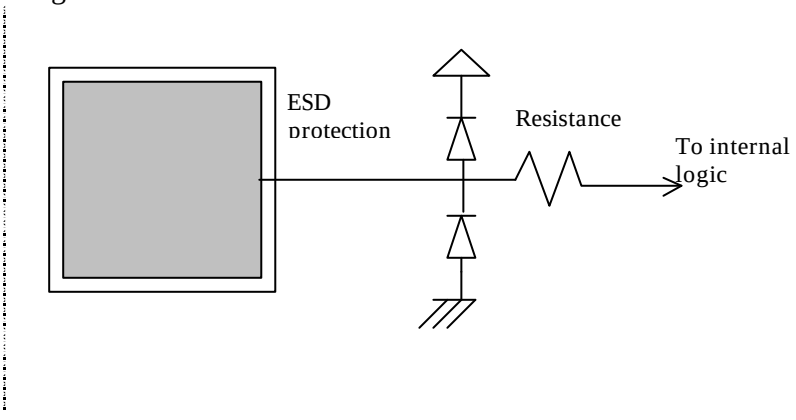


Fig. 10-5: Diodes for electrostatic discharge protection

10.6 I/O Pad description using Ibis

IBIS is a standard for electronic behavioral specifications of integrated circuit input/output analog characteristics. In order to enable an industry standard method to electronically transport IBIS Modeling Data between semiconductor vendors, simulation vendors, and end customers, a format has been proposed by the IBIS group [www.eia.org/ibis]. The intention of the IBIS to specify a consistent format that can be parsed by software, allowing simulation vendors to derive models compatible with their own products.
The version 3.2 of IBIS was finalized by an industry-wide group of experts representing various companies and interests. Regular "EIA IBIS Open Forum" meetings were held to accomplish this task. See [www.eia.org/ibis] for the complete backup of slides and meeting notes for the latest IBIS open forum.

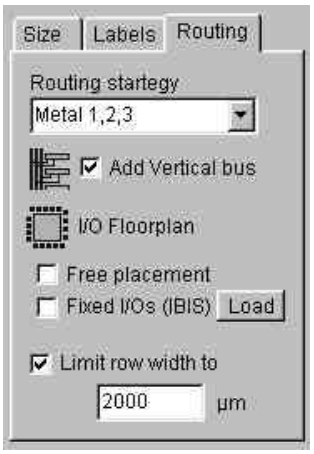


Fig. 10-6: Controlling the I/O pin assignment by an IBIS description file

Microwind2 uses IBIS to pilot the generation of the I/O pads, when compiling a Verilog file. Click the button "Load" in front of the check box "Fixed I/Os", in the Verilog menu. The default IBIS file is "default.IBS". The following screen appears:

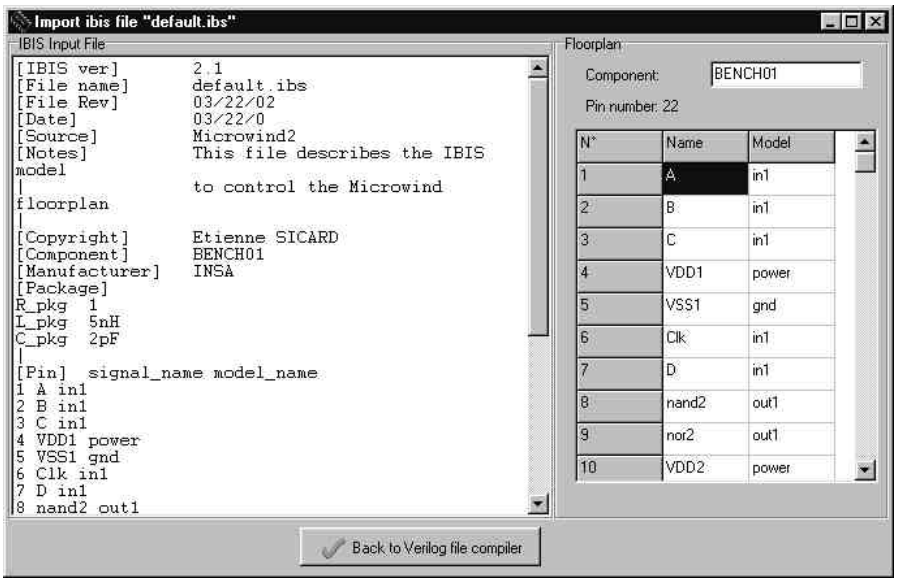


Fig. 10-6. The IBIS description file loaded for controlling the pin assignment

It can be seen that IBIS is a text file, with a simple structure based on keywords. We only use a very reduced set of the available keywords:

[IBIS Ver]	Specifies the IBIS template version. This keyword informs electronic parsers of the kinds of data types that are present in the file.
[File Rev]	Tracks the revision level of a particular .ibs file. Revision level is set at the discretion of the engineer defining the file.
[Component]	Marks the beginning of the IBIS description of the integrated circuit named after the keyword.
[Manufacturer]	Specifies the manufacturer's name of the component. Each manufacturer must use a consistent name in all .ibs files.
[Package]	Defines a range of values for the default packaging resistance, inductance, and capacitance of the component pins. Sub-Parameters are named R_pkg, L_pkg, C_pkg
[Pin]	Associates the component's I/O models to its various external pin names and signal names. Each line must contain either three or six columns. A pin line with three columns only associates the pin's signal and model. Six columns can be used to override the default package values. In that case headers R_pin, L_pin, and C_pin appear.

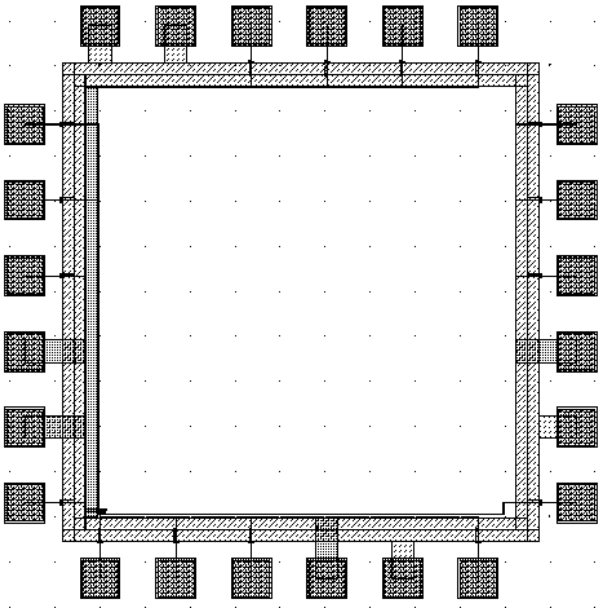


Fig. 10-7. The I/O pad generation constructed using the IBIS file default.IBS

Furthermore, Dsch2 uses IBIS to ease the generation of user symbols. You may load an IBIS file using the command File->Open, select the IBIS format "*.IBS", and click Open. If you click "Generate Symbol", the following screen appears.

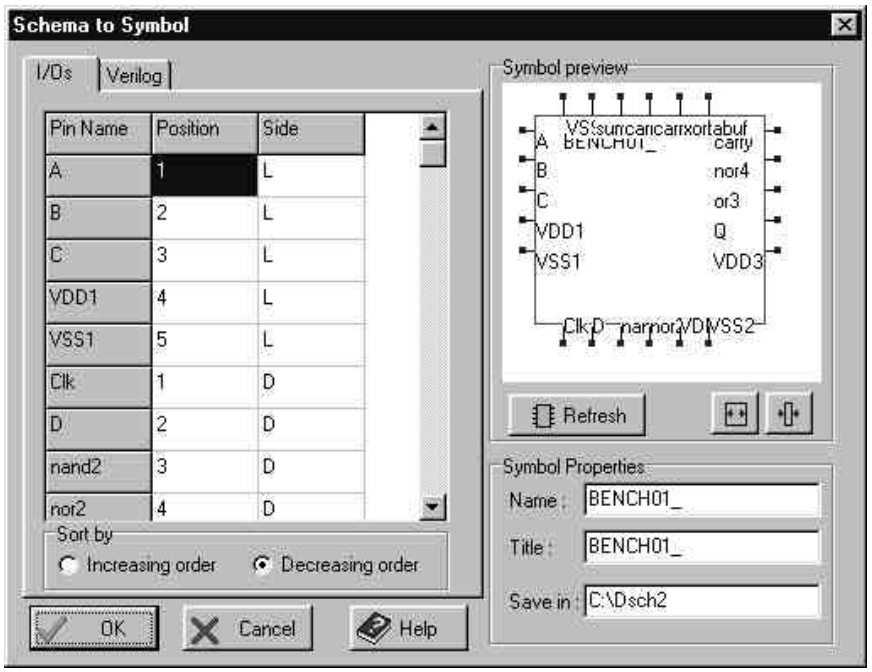


Fig. 10-8: User symbol generated from the IBIS description.

11 Design Rules

11.1 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 6-metal 0.25µm CMOS process. The default file is CMOS025.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 Files -> Properties , 'Set as Default Technology'.

11.2 Start Microwind with a specific design Rule File

To start Microwind with a specific design rule file, click with the right button of the mouse on the Microwind icon, select the "Properties" item, then the target. The default target may be:

```
C:\microwind2\Microwind2.exe
```

The command line may include two more parameters:

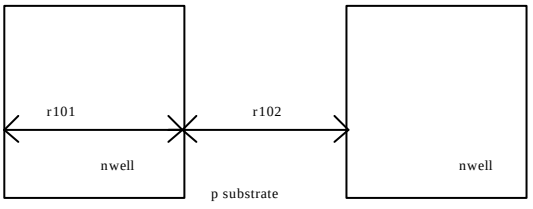
- The *First* parameter is the default mask file loaded at initialization
- The *Second* parameter is the design rule file loaded at initialization

The following command executes MICROWIND2 with a default mask file « **test.MSK** » and the rule file « **cmos018.RUL** ».

```
C:\microwind2\Microwind2.exe test cmos018.rul
```

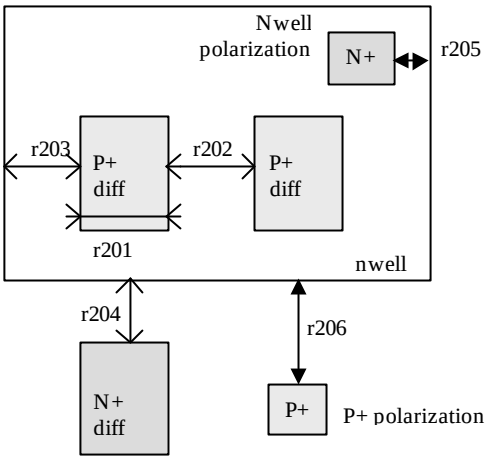
11.3 Nwell Design Rules

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



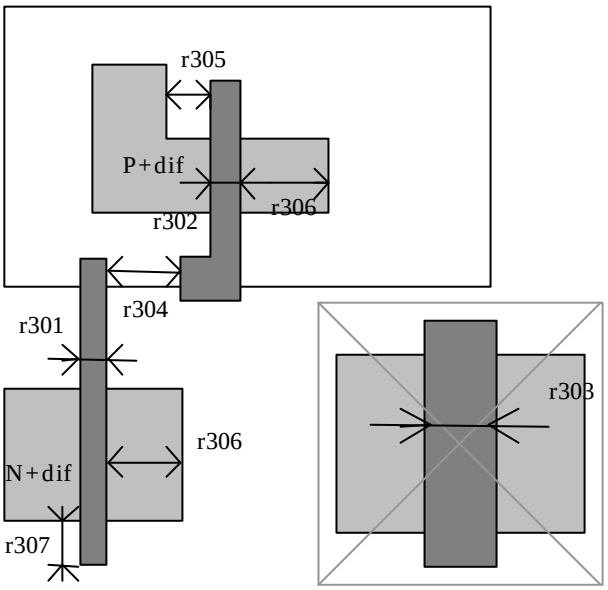
11.4 Diffusion Design Rules

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 Distance between Nwell and P+ polarization 6 λ
r210 Minimum surface : 24 λ^2

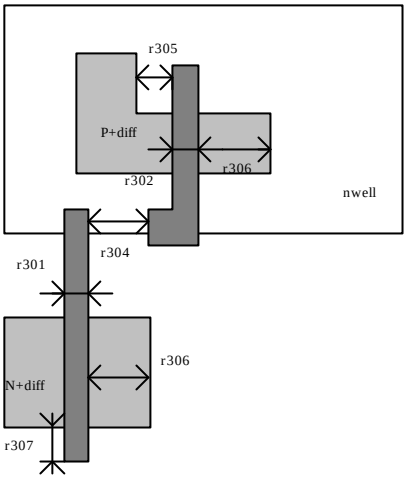


11.5 Polysilicon Design Rules

r301 Polysilicon width : 2 λ
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 polysilicium : 3 λ
r310 Minimum surface : 8 λ^2

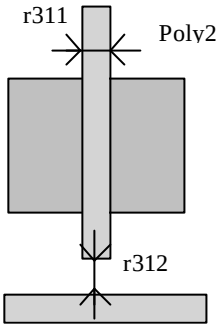


High voltage MOS



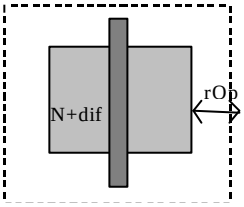
11.6 2nd Polysilicon Design Rules

r311 Polysilicon2 width : 2 λ
r312 Polysilicon2 gate on diffusion: 2 λ



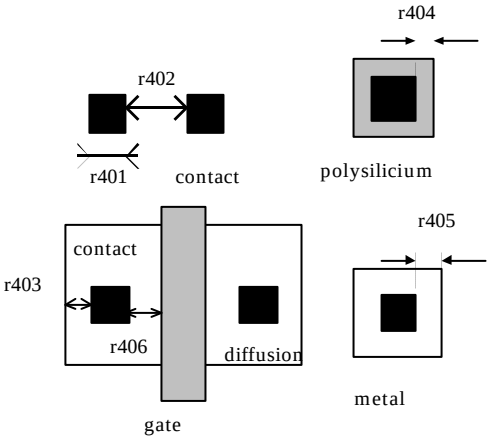
11.7 Option Design Rules

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



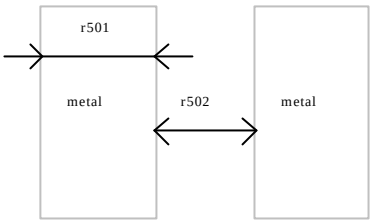
11.8 Contact Design Rules

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



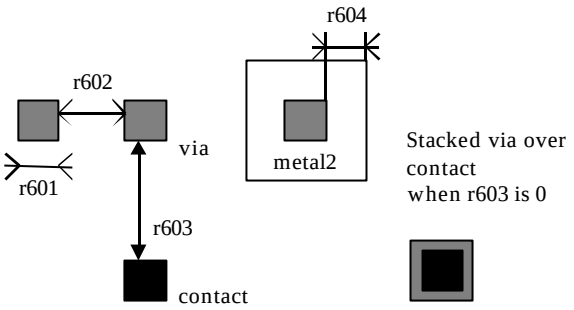
11.9 Metal & Via Design Rules

r501 Metal width : 4 λ
r502 Between two metals : 4 λ
r510 Minimum surface : 32 λ^2



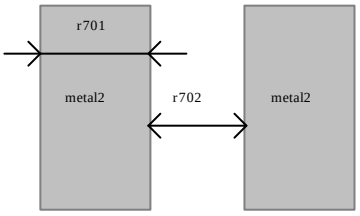
r601 Via width : 2 λ
r602 Between two Via: 5 λ
r603 Between Via and contact: 0 λ
r604 Extra metal over via: 2 λ
r605 Extra metal2 over via: 2 λ

When r603=0, stacked via over contact is allowed

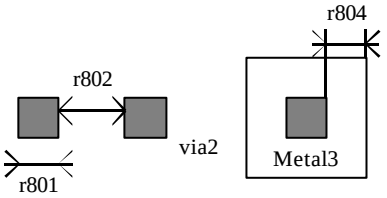


11.10 Metal2 & Via2 Design Rules

r701 Metal width: 4 λ
r702 Between two metal2 : 4 λ
r710 Minimum surface : 32 λ^2

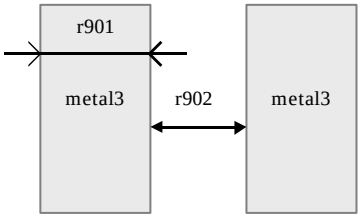


r801 Via2 width : 2 λ
r802 Between two Via2: 5 λ
r804 Extra metal2 over via2: 2 λ
r805 Extra metal3 over via2: 2 λ

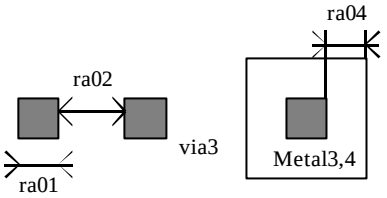


11.11 Metal3 & Via3 Design Rules

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

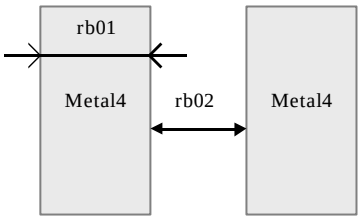


ra01	Via3 width : 2 λ
ra02	Between two Via3: 5 λ
ra04	Extra metal3 over via3: 2 λ
ra05	Extra metal4 over via3: 2 λ

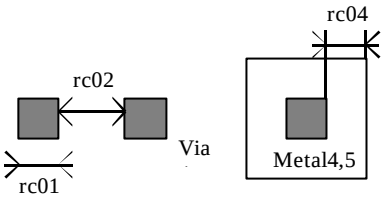


11.12 Metal4 & Via4 Design Rules

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

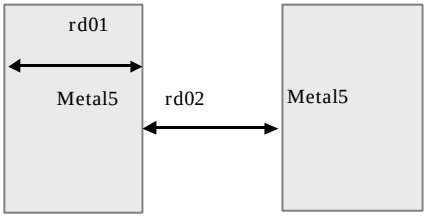


rc01	Via4 width : 2 λ
rc02	Between two Via4: 5 λ
rc04	Extra metal4 over via2: 3 λ
rc05	Extra metal5 over via2: 3 λ

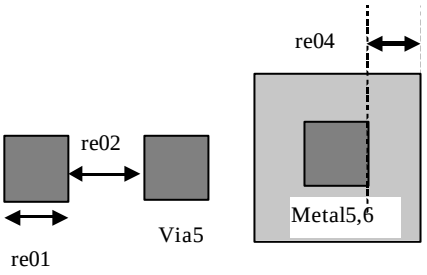


11.13 Metal5 & Via5 Design Rules

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

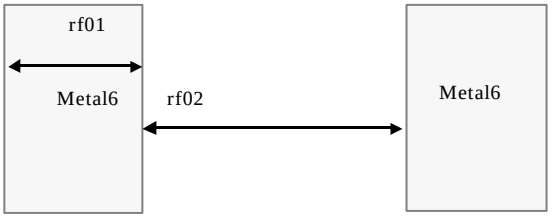


re01	Via5 width : 4 λ
re02	Between two Via5: 6 λ
re04	Extra metal5 over via5: 3 λ
re05	Extra metal6 over via5: 3 λ



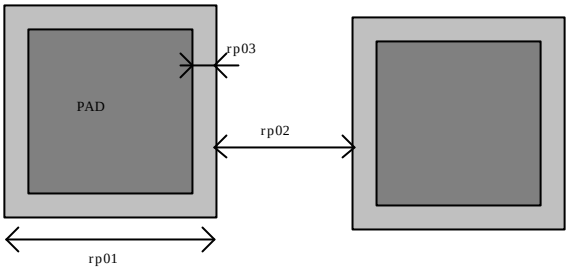
11.14 Metal6 Design Rules

rf01	Metal6 width: 8 λ
rf02	Between two metal6 : 15 λ
rf10	Minimum surface : 300 λ^2



11.15 Pad Design Rules

rp01	Pad width: 100 µm
rp02	Between two pads 100 µm
rp03	Opening in passivation v.s via : 5µm
rp04	Opening in passivation v.s metals:
rp05	5µm
	Between pad and unrelated active area : 20 µm



12 Electrical Rules

12.1 Electrical Circuit Extraction

MICROWIND2 includes a built-in extractor from layout to electrical circuit. Worth of interest are the MOS devices, capacitance and resistance. The flow is described in figure 12-1.

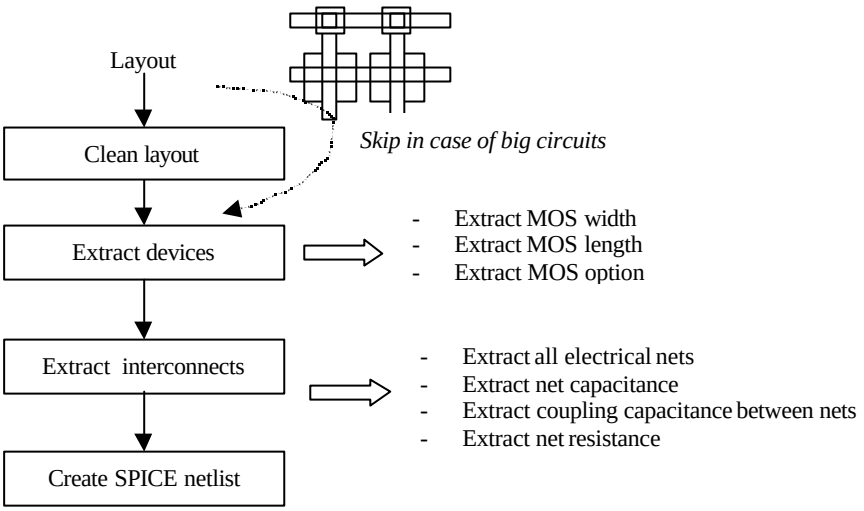


Figure 12-1: Extraction of the electrical circuit from layout

The first step consists in cleaning the layout. Mainly, redundant boxes are removed, overlapping boxes are transformed into non-overlapping boxes. In the case of complex circuits, MICROWIND2 may skip this cleaning step as it required a significant amount of computational time.

12.2 Capacitance

Each deposited layer is separated from the substrate by a SiO2 oxide and generated by a parasitic capacitor. The unit is the aF/µm^2 (atto = 10^{-18}). Basically all layers generate parasitic capacitors. Diffused layers generate junction capacitors (N+/P-, P+/N). The list of capacitance handled by MICROWIND2 is given below. The name corresponds to the code name used in CMOS025.RUL (CMOS 0.25µm)

Surface capacitance refers to the body. Vertical crosstalk capacitance refer to inter-layer coupling capacitance, while lateral crosstalk capacitance refer to adjacent interconnects.

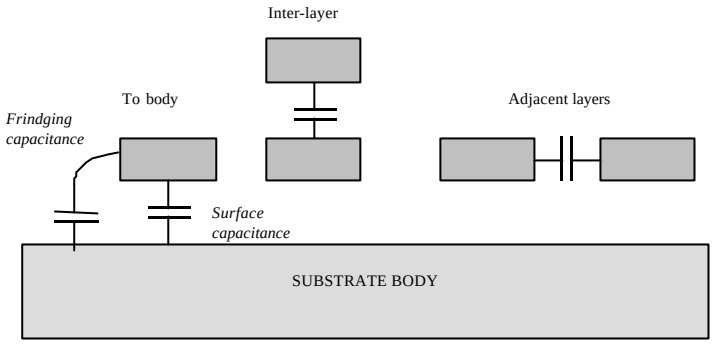


Figure 12-2: Capacitances

12.3 Surface Capacitance

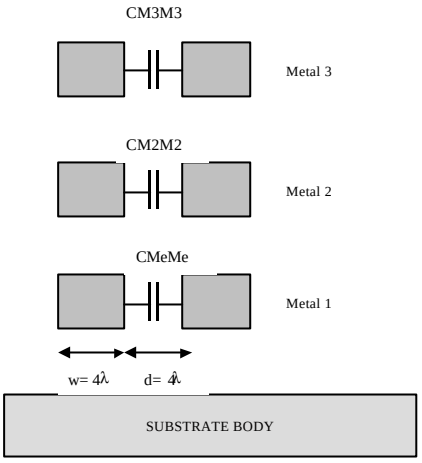
NAME	DESCRIPTION	LINEIC (aF/µm)	SURFACE (aF/µm^2)
CpoOxyde	Polysilicon/Thin oxide capacitance	n.c	4600
CpoBody	Polysilicon to substrate capacitance	n.c	80
CMEBody	Metal on thick oxide to substrate	42	28
CM2Body	Metal2 on body	36	13
CM3Body	Metal3 on body	33	10
CM4Body	Metal4 on body	30	6
CM5Body	Metal5 on body	30	5
CM6Body	Metal6 on body	30	4

12.4 Interlayer Capacitance

NAME	DESCRIPTION	VALUE (aF/µm^2)
CM2Me	Metal2 on metal 1	50
CM3M2	Metal3 on metal 2	50
CM4M3	Metal4 on metal 3	50
CM5M4	Metal5 on metal 4	50
CM6M5	Metal6 on metal 5	50

12.5 Crosstalk Capacitance

NAME	DESCRIPTION	VALUE (aF/µm)
CMeMe	Metal to metal (at 4λ distance, 4λ width)	10
CM2M2	Metal2 to metal 2	10
CM3M3	Metal3 to metal 3	10
CM4M4	Metal4 to metal 4	10
CM5M5	Metal5 to metal 5	10
CM6M6	Metal6 to metal6	10



The crosstalk capacitance value per unit length is given in the design rule file for a predefined interconnect width (w=4λ) and spacing (d=4λ).
In Microwind2,
The computed crosstalk capacitance is not dependant on the interconnect width *w*.
The computed crosstalk capacitance value is proportional to 1/*d* where *d* is the distance between interconnects.

12.6 Resistance

NAME	DESCRIPTION	VALUE (Ω)
RePo	Resistance per square for polysilicon	4
ReP2	Resistance per square for polysilicon2	4
ReMe	Resistance per square for metal	0.05
ReM2	Resistance per square for metal 2 (up to 6)	0.05
ReCo	Resistance for one contact	20
ReVi	Resistance for one via (up to via5)	2

12.7 Vertical Aspect of the Technology

The vertical aspect of the layers for a given technology is described in the RUL file after the design rules, using coed HE (height) and TH (thickness) for all layers. The figure 12-3 below illustrates the altitude 0, which corresponds to the channel of the MOS. The height of diffused layers can be negative, for P++ EPI layer for example.

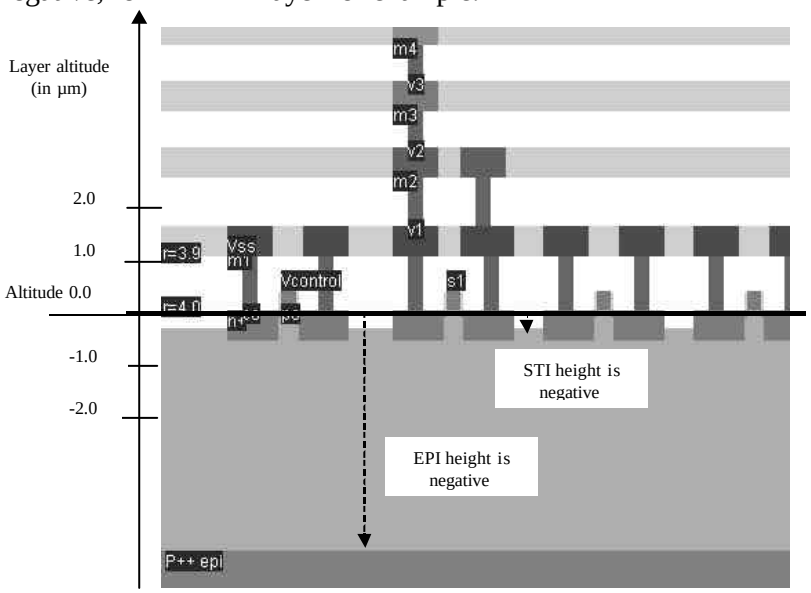


Figure 12-3: Description of the 2D aspect of the CMOS technology

LAYER	DESCRIPTION	PARAMETERS
EPI	Buried layer made of P++ used to create a good ground reference underneath the active area.	HEEPI for height (negative in respect to the origin) THEPI for thickness
STI	Shallow trench isolation used to separate the active areas.	HESTI for height THSTI for thickness
PASSIVATION	Upper SiO2 oxide on the top of the last metal layer	HEPASS for height THPASS for thickness
NITRIDE	Final oxide on the top of the passivation, usually Si3N4.	HENIT for height THNIT for thickness
NISO	Buried N- layer to isolate the Pwell underneath the nMOS devices, to enable forward bias and back bias	HENBURRIED for height THNBURRIED for thickness

12.8 Dielectrics

Some options are built in Microwind to enable specific features of ultra deep submicron technology. Details are provided in the table below.

CODE	DESCRIPTION	EXAMPLE VALUE
HIGHK	Oxide for interconnects (SiO2)	4.1
GATEK	Gate oxide	4.1
LOWK	Inter-metal oxide	3.0
LK11	Inter-metal1 oxide	3.0
LK22	Inter-metal2 oxide (up to LK66)	3.0
LK21	Metal2-Metal1 oxide	3.0
LK32	Metal3-Metal2 oxide (up to LK65)	3.0
TOX	Normal MOS gate oxide thickness	0.004 µm (40 Å)
HVTOX	High voltage gate oxide thickness	0.007 µm (70 Å)

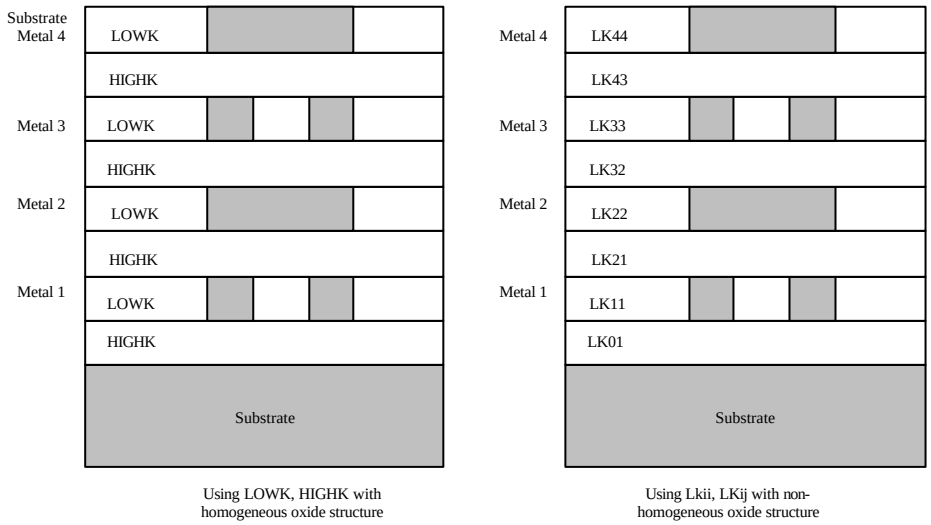


Fig. 12-4: Illustration of the use of LOWK, HIGHK dielectric constants (left figure) or detailed permittivity for each layer (right figure)

12.9 Simulation Parameters

The following list of parameters is used in Microwind2 to configure the simulation.

CODE	DESCRIPTION	TYPICAL VALUE
VDD	Supply voltage of the chip	2.0 V
HVDD	High voltage supply	3.3V
DELTAT	Simulator minimum time step to ensure convergence. You may increase this value to speed up the simulation but instability problems may rise.	0.5e-12 s
TEMPERATURE	Operating temperature of the chip	25 °C
RISETIME	Typical rise and fall time of clocks	0.05ns

12.10 Models Level1 and Level3 for analog simulation

Four types of MOS devices may be described as detailed in figure 12-4 (Data from SIA, 0.12µm CMOS technology). In the rule file, the keyword "MOS1", "MOS2', "MOS3" and "MOS4" are used to declare the device names appearing in menus. In 0.12µm technology, three types of MOS devices are declared as follows. Also, NMOS & PMOS keywords are used to select n-channel Mos or p-channel Mos device parameters.

Parameter	MOS1	MOS2	MOS3
Default name	High Speed	Low Leakage	High voltage
Vt (nmos)	0.3	0.5	0.7
Vt (pmos)	-0.3	-0.5	-0.7
KP (nmos)	300	300	200
KP (pmos)	150	150	100

```
* MOS definition
*
MOS1 low leakage
MOS2 high speed
MOS3 high voltage
```

Figure 12-5: Description of MOS options in 0.12µm technology (cmos012.RUL)

The list of parameters for level 1 and level 3 is given below:

PARAMETER	KEYWORD	DEFINITION	TYPICAL VALUE 0.25µm NMOS	pMOS
VTO	l3vto	Threshold voltage	0.4V	-0.4V
U0	l3u0	Low field mobility	0.06 m^2/V.s	0.025 m^2/V.s
PHI	l3phi	Surface potential at strong inversion	0.3V	0.3V

LD	l3ld	Lateral diffusion into channel	0.01µm	0.01µm
GAMMA	l3gamma	Bulk threshold parameter	0.4 $V^{0.5}$	0.4 $V^{0.5}$
KAPPA	l3kappa	Saturation field factor	0.01 V^{-1}	0.01 V^{-1}
VMAX	l3vmax	Maximum drift velocity	150Km/s	100Km/s
THETA	l3theta	Mobility degradation factor	0.3 V^{-1}	0.3 V^{-1}
NSS	l3nss	Substhreshold factor	0.07 V^{-1}	0.07 V^{-1}
TOX	l3tox	Gate oxide thickness	3nm	3nm

For MOS2, MOS3 and MOS4, only the threshold voltage, mobility ant oxides thickness are user-accessible. All other parameters are identical to MOS1.

PARAMETER	**KEYWORD**	**DEFINITION**	**TYPICAL VALUE 0.25µm** NMOS	pMOS
VTO Mos2	l3v2to	Threshold voltage for MOS2	0.5V	-0.5V
VTO Mos3	l3v3to	Threshold voltage for MOS3	0.7V	-0.7V
U0 Mos2	l3u2	Mobility for MOS2	0.06	0.025
U0 Mos3	l3u3	Mobility for MOS3	0.06	0.025
TOX Mos 2	l3t2ox	Thin oxide thickness for MOS2	3nm	3nm
TOX Mos 3	l3t3ox	Thin oxide thickness for MOS3	7nm	7nm

```
* Nmos Model 3 parameters
*
NMOS
l3vto = 0.4
l3u0 = 0.06
l3tox = 3e-9
l3vmax = 170e3
l3gamma = 0.4
l3theta = 0.3
l3kappa = 0.06
l3phi = 0.2
l3ld = 8e-9
l3nss = 0.06
*
* high speed
l3v2to = 0.3
l3u2 = 0.06
l3t2ox = 3e-9
*
* high voltage
l3v3to = 0.7
l3u3 = 0.06
l3t3ox = 7e-9
```

12.11 BSIM4 Model for analog simulation

The list of parameters for BSIM4 is given below:

Parameter	Keyword	Description	NMOS value in 0.12µm	PMOS value in 0.12µm
VTHO	b4vtho	Long channel threshold voltage at Vbs = 0V	0.3V	0.3V
VFB	b4vfb	Flat-band voltage	-0.9	-0.9
K1	b4k1	First-order body bias coefficient	0.45 V1/2	0.45 V1/2
K2	b4k2	Second-order body bias coefficient	0.1	0.1
DVT0	b4d0vt	First coefficient of short-channel effect on threshold voltage	2.2	2.2
DVT1	b4d1vt	Second coefficient of short-channel effect on Vth	0.53	0.53
ETA0	b4et	Drain induced barrier lowering coefficient	0.08	0.08
NFACTOR	B4nf	Sub-threshold turn-on swing factor. Controls the exponential increase of current with Vgs.	1	1
U0	b4u0	Low-field mobility	0.060 m2/Vs	0.025 m2/Vs
UA	b4ua	Coefficient of first-order mobility degradation due to vertical field	11.0e-15 m/V	11.0e-15 m/V
UC	b4uc	Coefficient of mobility degradation due to body-bias effect	-0.04650e-15 V-1	-0.04650e-15 V-1
VSAT	b4vsat	Saturation velocity	8.0e4 m/s	8.0e4 m/s
WINT	b4wint	Channel-width offset parameter	0.01^{e}-6µm	0.01^{e}-6µm
LINT	b4lint	Channel-length offset parameter	0.01^{e}-6µm	0.01^{e}-6µm
PSCBE1	b4pscbe1	First substrate current induced body-effect mobility reduction	4.24e8 V/m	4.24e8 V/m
PSCBE2	b4pscbe2	Second substrate current induced body-effect mobility reduction	4.24e8 V/m	4.24e8 V/m
KT1	b4kt1	Temperature coefficient of the threshold voltage.	-0.1V	-0.1V
UTE	b4ute	Temperature coefficient for the zero-field mobility U0.	-1.5	-1.5
VOFF	b4voff	Offset voltage in subthreshold region.	-0.08V	-0.08V
PCLM	b4pclm	Parameter for channel length modulation	1.2	1.2
TOXE	b4toxe	Gate oxide thickness	3.5e-9m	3.5e-9m
NDEP	b4ndep		0.54	0.54
XJ	b4xj	Junction depth	1.5e-7	1.5e-7

For MOS2, MOS3 and MOS4, only the threshold voltage, mobility ant oxides thickness are user-accessible. All other parameters are identical to MOS1.

12.12 TEC file for DSCH2

The logic simulator includes a current evaluator. To run this evaluation, the following parameters are proposed in a TEC file (example: cmos012.TEC):

```
DSCH 2.0 - technology file
NAME "CMOS 0.12um"
VERSION 14.12.2001
* Time unit for simulation
TIMEUNIT = 0.01
* Supply voltage
VDD = 1.2
* Typical gate delay in ns
TDelay = 0.02
* Typical wire delay in ns
TWireDelay = 0.07
* Typical current in mA
TCurrent = 0.5
* Default MOS length and width
ML  = "0.12u"
MNW = "1.0u"
MPW = "2.0u"
```

13 Design Rule File for 0.12µm

```
MICROWIND 2.0
*
* Rule File for CMOS 0.12µm
*
* Date : 27 Apr 99 created by Etienne
Sicard
*        04 Jan 00 smaller dt
*        03 Avr 01 2d cross-section
*        17 Apr 01 update params, add
high voltage, tox, level3
*        20 Apr 01 various lowK, 4
types of MOS
*        10 Dec 01 Bsim4 model, gatek
*
NAME CMOS 0.12µm - 6 Metal
*
lambda = 0.06    (Lambda is set to half
the gate size)
metalLayers = 6  (Number of metal
layers)
*
* Dielectrics
*
lowK = 3.2  (inter-metal oxide
permittivity)
gateK = 5.0  (HighK gate dielectric)
*
* Design rules associated to each layer
*
* Well
*
r101 = 10    (well width)
r102 = 11    (well spacing)
*
* Diffusion
*
r201 = 4     (diffusion width)
r202 = 4     (diffusion spacing)
r203 = 6     (border of nwell on diffp)
r204 = 6     (nwell to next diffn)
*
* Poly
*
r301 = 2     (poly width)
r302 = 2     (gate length)
r303 = 4     (high voltage gate length)
r304 = 3     (poly spacing)
r305 = 1     (spacing poly and
unrelated diff)
r306 = 4     (width of drain and source
diff)
r307 = 2     (extra gate poly)
*
* Poly 2
*
r311 = 2     (poly2 width)
r312 = 2     (poly2 spacing)
*
* Contact
r401 = 2     (contact width)
r402 = 3     (contact spacing)
r403 = 2     (metal border for contact)
r404 = 2     (poly border for contact)
r405 = 2     (diff border for contact)
r406 = 3     (contact to gate)
r407 = 2     (poly2 border for contact)
*
*  metal
r501 = 3    (metal width)
r502 = 4    (metal spacing)
*  via
r601 = 2    (Via width)
r602 = 4    (Spacing)
r604 = 2    (border of metal&metal2)
*  metal 2
r701 = 3    (Metal 2 width)
r702 = 4
*  via 2
r801 = 2    (Via width)
r802 = 4    (Spacing)
r804 = 2    (border of metal2&metal3)
*  metal 3
r901 = 3    (width)
r902 = 4    (spacing)
*  via 3
ra01 = 2    (Via width)
ra02 = 4    (Spacing)
ra04 = 2    (border of metal3&metal4)
*  metal 4
rb01 = 3    (width)
rb02 = 4    (spacing)
*  via 4
rc01 = 2    (Via width)
rc02 = 4    (Spacing)
rc04 = 2    (border of metal4&metal5)
*  metal 5
rd01 = 8    (width)
rd02 = 8    (spacing)
*  via 5
re01 = 5    (Via width)
re02 = 5    (Spacing)
re04 = 2    (border of metal5&metal6)
*  metal 6
rd01 = 8    (width)
rd02 = 15   (spacing)
*
* Pad rules
*
rp01 = 800  (Pad width)
rp02 = 800  (Pad spacing)
rp03 = 40   (Border of Vias)
rp04 = 40   (Border of metals)
rp05 = 200  (to unrelated active areas)
*
* Thickness of conductors for process
aspect
```

```
* All in µm
*
* P++ epitaxial
thepi = 1.0
heepi = -4.0
*
* Shallow tretch isolation
thsti = 0.8
hesti = -0.8
*
* Poly
thpoly = 0.20
hepoly = 0.01
*
* Poly2
thp2 = 0.2
hep2 = 0.22
*
* Diffusions
thdn = 0.4
thdp = 0.4
thnw = 1.0
*
* Metallisation
thme = 0.5
heme = 1.2
thm2 = 0.5
hem2 = 2.2
thm3 = 0.5
hem3 = 3.2
thm4 = 0.5
hem4 = 4.2
thm5 = 0.7
hem5 = 5.4
thm6 = 0.7
hem6 = 6.6
thpass = 0.5
hepass = 7.8
thnit = 0.6
henit = 8.4
*
* Resistances Copper
* Unit is ohm/square
*
repo = 4
rep2 = 4
reme = 0.06
rem2 = 0.06
rem3 = 0.06
rem4 = 0.06
rem5 = 0.05
rem6 = 0.05
*
* Resistances vias: unit is ohm/via
reco = 20
revi = 2
rev2 = 2
rev3 = 2
rev4 = 1
rev5 = 1
*
*
* Parasitic capacitances
*
cpoOxyde= 4600 (Surface capacitance
Poly/Thin oxyde aF/µm2)
cpobody = 400  (No lineic capa)
cp2body = 400
cmebody = 550  (Strong value due to
upper and lower capa)
cm2body = 550  (to metal grid i.e 2*Cg)
cm3body = 550
cm4body = 550
cm5body = 450
cm6body = 450
cgsn = 500          ( Gate/source capa
of nMOS)
cgsp = 500
cmelineic = 0
cm2lineic = 0
cm3lineic = 0
cm4lineic = 0
cm5lineic = 0
cm6lineic = 0
*
* Vertical crosstalk
*
cmepoly = 60
cm2me = 50
cm3m2 = 50
cm4m3 = 50
cm5m4 = 50
cm6m5 = 50
*
* Lateral Crosstalk
*
cmextk = 70      (Lineic capacitance
for crosstalk coupling in aF/µm)
cm2xtk = 100      (C is computed using
Cx=cmextk*l/spacing)
cm3xtk = 100
cm4xtk = 100
cm5xtk = 100
cm6xtk = 100
*
* Junction capacitances
*
cdnpwell = 350  (n+/psub)
cdpnwell = 300  (p+/nwell)
cnwell = 250    (nwell/psub)
cpwell = 100    (pwell/nsub)
cldn = 100      (Lineic capacitance
N+/P- aF/µm)
cldp = 100      (Idem for P+/N-)
*
* MOS definition
*
MOS1 low leakage
MOS2 high speed
MOS3 high voltage
*
* Nmos Model 3 parameters
*
NMOS
l3vto = 0.4
l3u0 = 0.06
l3tox = 3e-9
l3vmax = 170e3
l3gamma = 0.4
l3theta = 0.3
l3kappa = 0.06
l3phi = 0.2
```

```
l3ld = 8e-9
l3nss = 0.06
*
* high speed
l3v2to = 0.3
l3u2 = 0.06
l3t2ox = 3e-9
*
* high voltage
l3v3to = 0.7
l3u3 = 0.06
l3t3ox = 7e-9
*
* Pmos Model 3
*
PMOS
l3vto = -0.4
l3u0 = 0.02
l3tox = 3e-9
l3vmax = 120e3
l3gamma = 0.4
l3theta = 0.3
l3kappa = 0.06
l3phi = 0.2
l3ld = 8e-9
l3nss = 0.06
*
* high speed
l3v2to = -0.3
l3u2 = 0.02
l3t2ox = 3e-9
*
* high voltage
l3v3to = -0.7
l3u3 = 0.02
l3t3ox = 7e-9
*
* BSIM4 parameters
*
* Nmos
*
NMOS
b4vtho = 0.4
b4k1 = 0.45
b4k2 = 0.1
b4xj = 1.7e-7
b4toxe = 3.5e-9
b4ndep = 1.8e17
b4d0vt = 2.3
b4d1vt = 0.54
b4vfb = -0.9
b4u0 = 0.068
b4ua = 1e-15
b4uc = -0.047e-15
b4vsat = 100e3
b4pscbe1 =230e6
b4ute = -1.8
b4kt1 = -0.1
b4lint = 0.01e-6
b4wint = 0.02e-6
b4xj = 1.5e-7
b4ndep = 1.7e17
b4pclm = 1.1
*
* high speed
b4v2to = 0.3
b4t2ox = 3e-9
*
* high voltage
b4v3to = 0.7
b4t3ox = 7e-9
*
* Pmos BSIM4
*
PMOS
b4vtho = 0.4
b4k1 = 0.45
b4k2 = 0.1
b4xj = 1.7e-7
b4toxe = 3.5e-9
b4ndep = 1.8e17
b4d0vt = 2.3
b4d1vt = 0.54
b4vfb = -0.9
b4u0 = 0.028
b4ua = 1e-15
b4uc = -0.047e-15
b4vsat = 60e3
b4pscbe1 =230e6
b4ute = -1.8
b4kt1 = -0.1
b4lint = 0.01e-6
b4wint = 0.02e-6
b4xj = 1.5e-7
b4ndep = 1.7e17
b4pclm = 0.7
*
* high speed
b4v2to = 0.3
b4t2ox = 3e-9
*
* high voltage
b4v3to = 0.7
b4t3ox = 7e-9
*
* CIF Layers
* MicroWind layer, CIF layer, overetch
*
cif nwell 1 0.0
cif diffp 17 0.5
cif diffn 16 0.5
cif aarea 2 0.5
cif poly 13 0.0
cif contact 19 0.025
cif metal 23 0.0125
cif via 25 0.0125
cif metal2  27 0.0125
cif via2 32 0.0125
cif metal3  34 0.0125
cif via3 35 0.0125
cif metal4 36 0.0125
cif via4 52 0.0125
cif metal5 53 0.0
cif via5 54 0.0
cif metal6 55 0.0
cif passiv 31 0.0
cif text 56 0.0
*
*
* MicroWind simulation parameters
*
```

```
deltaT = 0.30e-12 (Minimum simulation
interval dT)
vdd = 1.2
hvdd = 3.3
temperature = 27
riseTime = 0.05
*
* End CMOS 0.12µm
*
```

14 References

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

[2] M. John, S. Smidth, "Application Specific Integrated Circuits", Addison Wesley, 1997, ISBN 0-201-50022-1, www.awl.com/cseng

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

[4] Y. P. Tsividis "Operating and Modeling of the MOS transistor", McGraw-Hill, 1987, ISBN 0-07-065381-X

[5] S. M Sze "Physics of Semiconductor devices", John-Wiley, 1981, ISBN 0-471-05661-8

[6] Y. Cheng, C. Hu "MOSFET Modeling & BSIM3 user's guide", Kluwer Academic Publishers, 1999

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

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

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

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

[11] C. Motchenbacher, J. A. Connely "Low Noise Electronic System Design", Wiley & sons, 1993, ISBN 0-471-57742-1

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