Tutorial of PSPICE Power Simulation for CMOS Circuit

--Xingguo Xiong

In this tutorial, we will use PSPICE to simulate the average power consumption of a CMOS circuit (nand gate). Because the power consumption of CMOS circuit is related to the charging and discharging of capacitances, it's important that we design the layout and extract the parasitic capacitances from the layout for the power simulation. In this way, we can achieve a more precise result. We will perform the power analysis in two steps:

1). Layout design and netlist extraction in Mentor Graphics IC Station.

2). Construct the spice circuit file from extracted netlist and include the auxiliary circuit for power analysis. Then use OrCAD PSPICE A/D tool to simulate the PSICE codes to achieve the pseudo power plot V(Pav) of the circuit.

Section 1. CMOS Layout Design and Netlist Extraction in Mentor Graphics IC Station

For the detailed procedures about the schematic and layout design of CMOS nand gate in Mentor Graphics IC Station, please refer to my previous "Mentor Graphic Tools Tutorial". The schematic of the CMOS nand gate designed in my previous Mentor Graphics Tools Tutorial is shown as below.

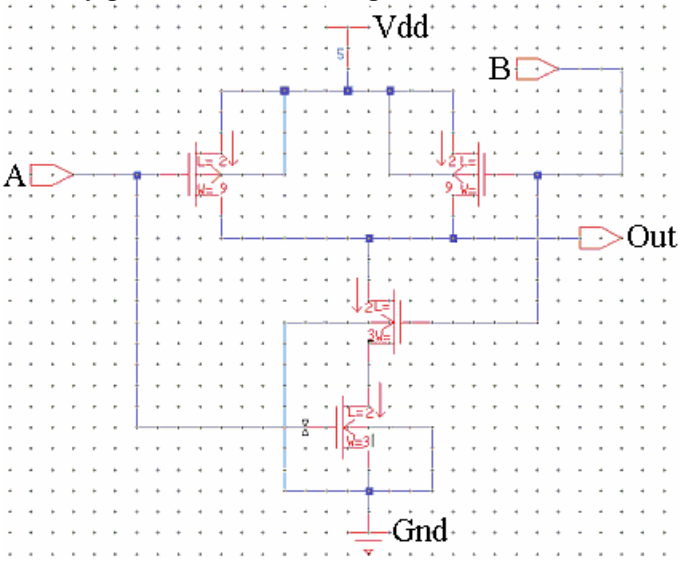


Figure 1. Schematic of a CMOS nand gate

The rough sketch of the CMOS nand gate layout in my previous "Mentor Graphics Tools Tutorial" is shown in Figure 2.

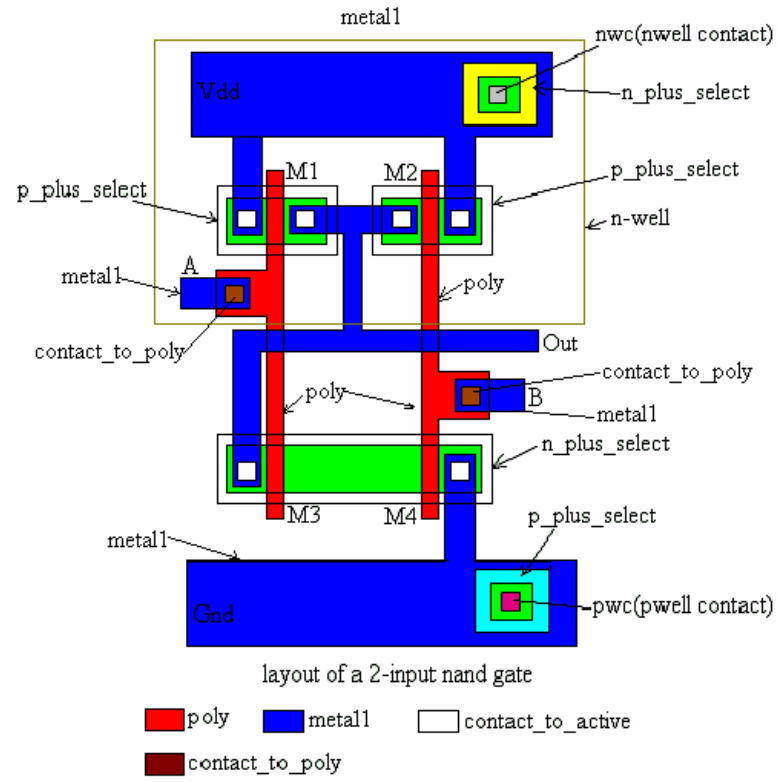


Figure 2. Rough sketch of the layout of CMOS nand gate

The designed layout of the CMOS nand gate in my previous "Mentor Graphics Tools Tutorial" is shown in Figure 3.

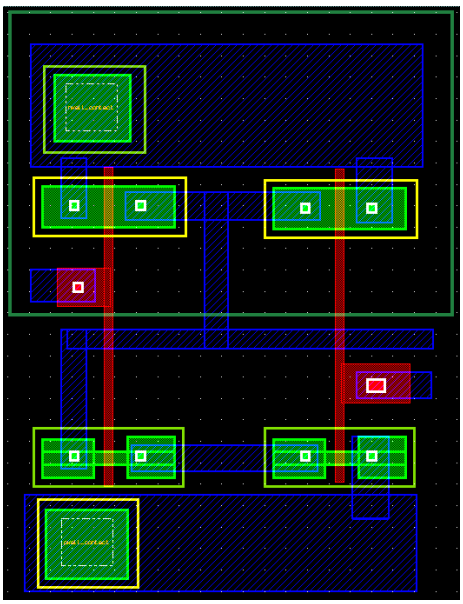


Figure 3. Layout of CMOS nand gate in IC station

Hint 1: Sizing of PMOS/NMOS transistors.

Since the mobility μ_p of holes in PMOS transistor is about 3 times as the mobility μ_n of electrons in NMOS transistor, in order to achieve equal rise and fall time for a CMOS circuit, generally we prefer to select the PMOS transistor size to be 3 times as that of NMOS transistors. That is:

$$\frac{(W_p / L_p)}{(W_n / L_n)} = 3:1$$

Generally we can select L_p=L_n, thus:

$$\frac{W_p}{W_n} = 3:1$$

We maintain this ratio in order to achieve symmetric behavior in digital design.

Hint 2: Sometimes you may have difficulty in meeting the design rules for the source and drain of a small size transistor. For example, if you set the length of NMOS transistor to be W_n=3λ(minimum size), you may find that according to design rules the size of source and drain of NMOS transistor need to be at least 6λ. In order to solve this problem, you may use a H-shape layout design for the transistor. That is, you can maintain a narrow channel while enlarge the source and drain regions in both sides of the channel, as shown in Figure 4.

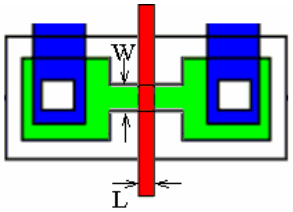


Figure 4. H-shape design for the layout of a minimum size transistor

Hint 3: Layout design for complex CMOS circuit

For the layout design of complex CMOS circuit, you need to use stick diagram and Euler path to figure out the arrangement and the connection of the MOS transistors. This will lead to a compact design with small circuit area. However, if you are still not familiar with these design techniques, here you can use a fast and convenient way for the layout design. It's easy and convenient, however, it's never a good design at all because the area will be very large. If you are not familiar with layout design techniques, you can temporarily use this way for your layout design. However, in the future when you get familiar with the layout design techniques, you should not use this method anymore. Let's use the following circuit as an example to demonstrate this "easy" layout design method.

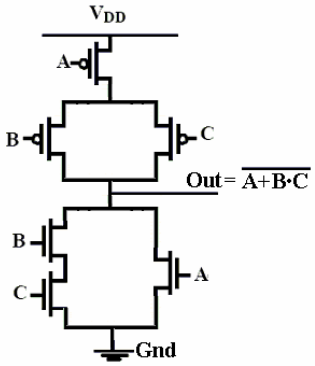


Figure 5. Schematic of a CMOS gate

For the circuit shown in Figure 5, totally there are 6 transistors and two power rails (Vdd and Gnd). First let's translate each transistor into layout and place them according to the relative locations in the schematic. We also put a Vdd power rail (metal1) in the top, and a Gnd power rail (metal1) in the bottom, and make the corresponding bulk connections. Then we plot a n-well to enclose all the PMOS transistors, as shown in Figure 6.

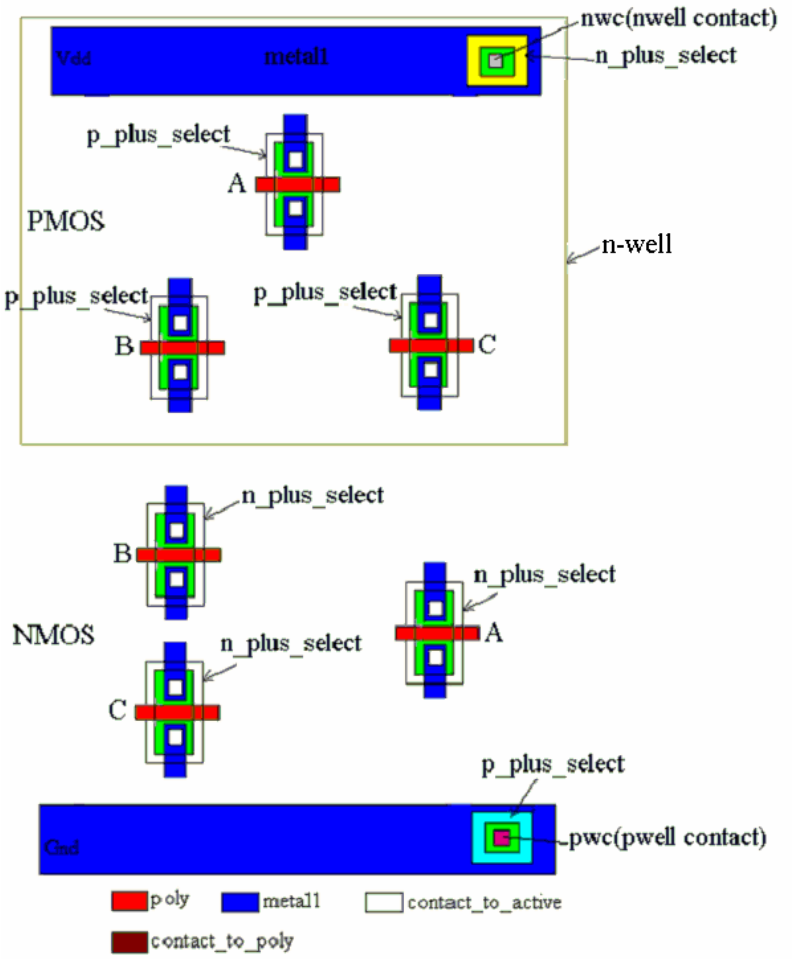


Figure 6. Place the transistors according to the arrangement in schematic

After that, you can make the poly and metal connections as below.

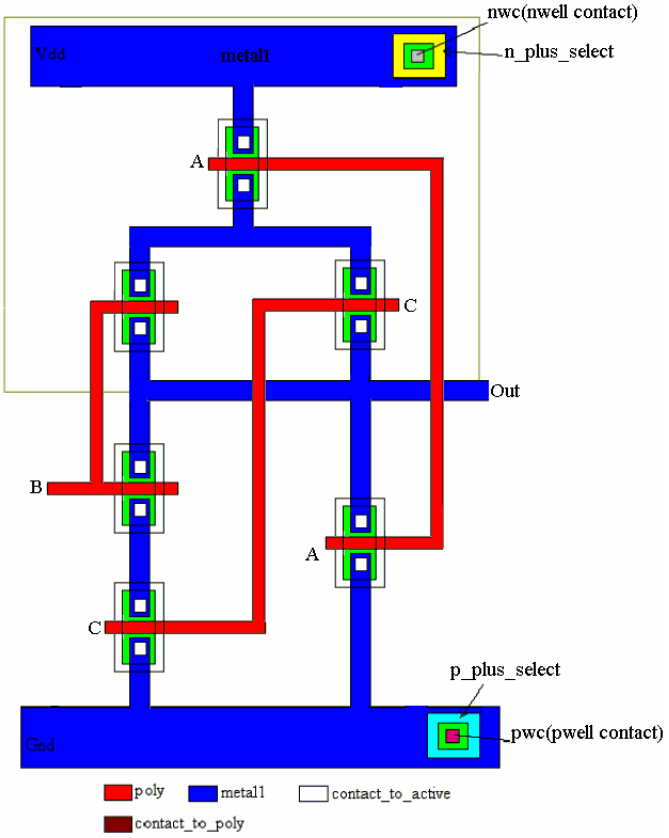


Figure 7. Layout design after making wire connections

Now based on Figure 7, you should be ready to design your layout in Mentor Graphics IC station. This is not a good compact layout design. It's only for your temporary solution when you are still a newbie in layout design. In the future when you get more familiar with layout design, you should use stick diagram and Euler path to work out a compact custom design.

Also please note that you should physically connect all the nodes with the same name together in your layout design. For example, in Figure 7, you have two inputs for node A, then you need to physically connect them together with poly wire (or metal1, etc.). However, sometimes your circuit may be complex and it's not easy to connect them while avoiding overlap problem. We do have certain ways to help reduce the wire overlapping problem. For example, you can make all the horizontal wires to be metal1, while all the vertical wires to be metal2, as shown in Figure 8. If you want a connection in a certain overlap, you can just place a via there.

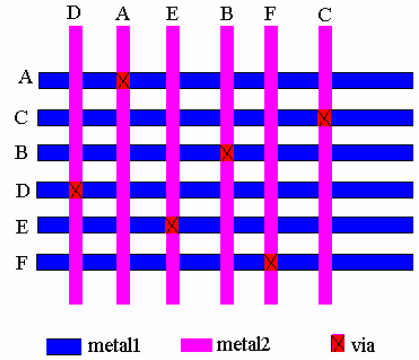


Figure 8. Complex wire connections to avoid overlap problem

But if you are still a newbie to layout design, you may just name the nodes you wish to connect with the same node name. When you extract the netlist, the nodes with same node name will be treated as short-circuit to each other. Thus you needn't make physical connection for them at this time. However, this is only a temporary solution. In the future when you are familiar with layout design, you should physically connect all the nodes with the same node name.

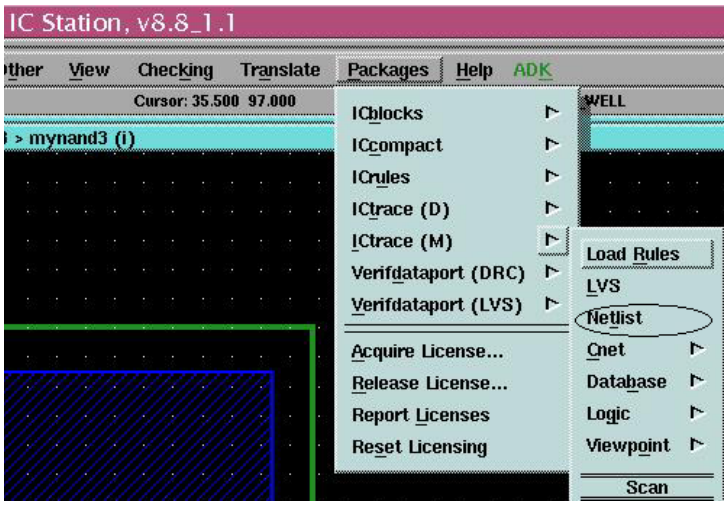
Please note that here we already named the ports (Vdd, Gnd, A, B, Out) in the layout although they are not shown. As for how to name the ports, please refer to my previous "Mentor Graphics Tools Tutorial". Now based on this, we are ready extract the netlist from the layout.

1. Extract Netlist from the Layout in IC Station

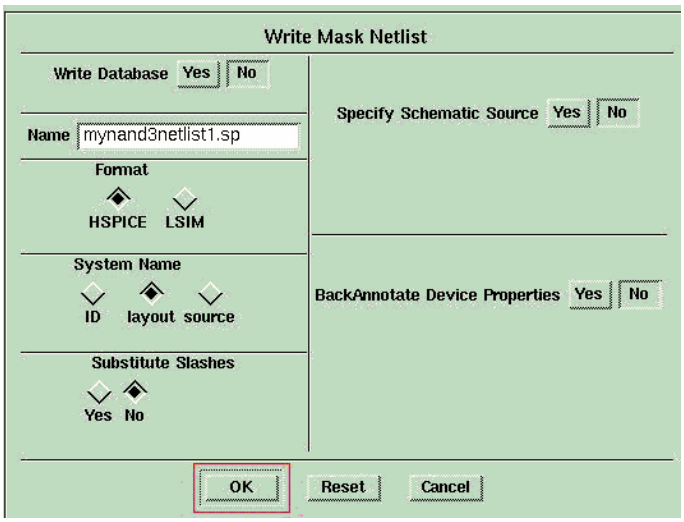
In order to simulate a circuit, we can design the schematic in Design Architect (DA), and then simulate its function in Accusim. However, the parasitic effects (parasitic capacitance, resistance, etc.) information is not available in schematic. In order to obtain the parasitic effects, we must design the layout and extract the netlist from the layout. With these parasitic effects, we can perform a post-layout simulation to have more precise evaluation about timing (delay) and power of the VLSI circuit.

There are some good Mentor Graphics tools to extract the post layout netlist of a circuit with parasitic effects, such as ICextract, Calibre Interactive-PEX. However, it seems these modules are not installed in our unix machines yet. Currently we will just extract the post layout netlist from ICtrace(M). It can extract the size (width W and length L) as well as the area of source and drain of the transistors. However, it cannot extract the parasitic capacitance and resistance. The detailed procedures are shown below.

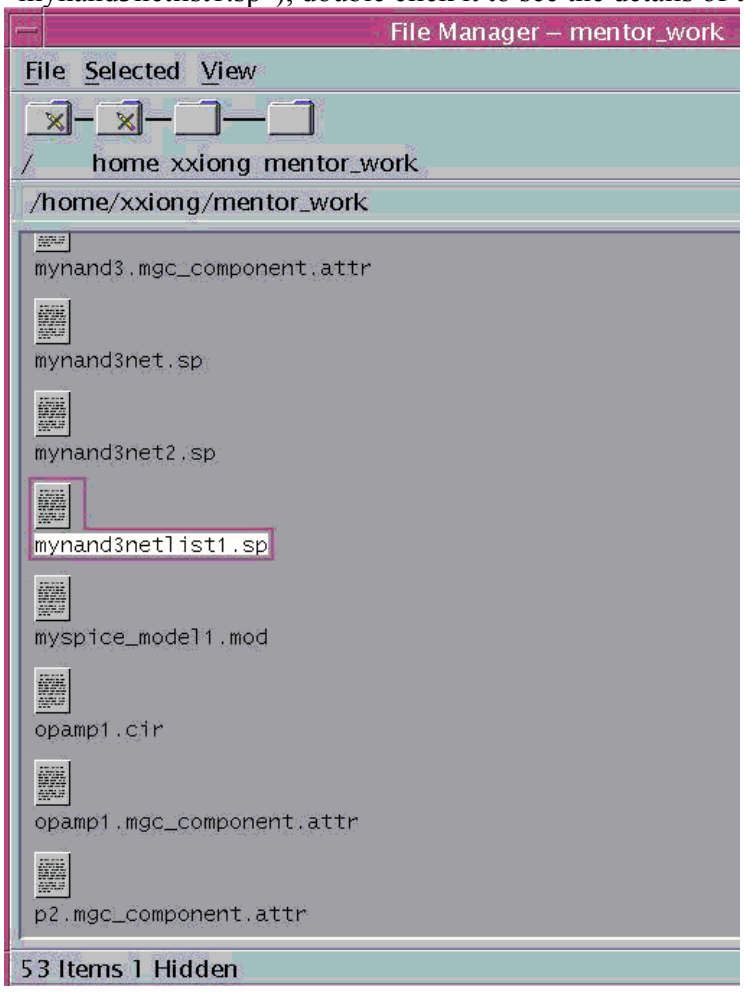
1). In IC station, open your layout, then click on the menu on the top: Packages—ICtrace(M)—Netlist, as shown below.



2). There will be a popup window, as shown below. Please input the name you wish for the output spice netlist file in the "Name" line. For example, here we name it as "mynand3netlist1.sp". However, you can choose any other name you like, but with the file extension name as ".sp". Please click to select "format" as "HSPICE", "System name" as "layout", "Substitute Slashes" as "No", "Specify Schematic Source" as "No", BackAnnotate Device Properties" as "No". Then click OK.



3). Now you already extracted the netlist for the circuit. Please save your layout design and exit the IC station. Please right click your mouse on the empty area of your desktop, in the popup menu, select "Folders—File Manager-Home", you will see the file manager window as below. It's just like Windows File Explorer. It's convenient to organize your files and folders. Your extracted netlist file should be located in the "\home\your_username\mentor_work" directory. Please find the file (here it's "mynand3netlist1.sp"), double click it to see the details of the netlist file.



For the mynand3netlist1.sp file, the contents are listed as below:

```
* File: /home/xxiong/mentor_work/mynand3netlist1.sp.   Creation time: Sun Oct 29
19:10:17 2006

.subckt mynand3 A B GND Out Vdd
* devices:
m0 Out A Vdd Vdd p l=0.6u w=2.7u ad=11.745p as=11.745p
m1 Vdd B Out Vdd p l=0.6u w=2.7u ad=11.745p as=11.745p
m2 5 A Out GND n l=0.6u w=0.9u ad=10.35p as=10.395p
m3 GND B 5 GND n l=0.6u w=0.9u ad=10.35p as=10.395p
.ends mynand3
```

The line started with a "*" sign indicates that this is a comment line. The syntax to describe a transistor is:

```
Mname ND NG NS NB ModName <L=VAL> <W=VAL> <AD=VAL> <AS=VAL> <PD=VAL>
  + <PS=VAL> <NRD=VAL> <OFF> <IC=VDS.VGS.VBS>
```

in which
Mname: The name of the transistor. The first letter "M" indicates that this is a MOS transistor, "name" can be any name you like. For example, M1, M20, Mab3c, etc.
ND, NG, NS and NB: the node names of the Drain, Gate, Source and Bulk terminals, respectively.
ModName: the name of the transistor model, defined as PMOS or NMOS in SPICE parameter.
L and W: the length and width of the transistor. For example, "3u" means 3μm, etc.
AD and AS: the area of source and drain. For example, "18p" means $18\times10^{-12}m^2$ etc.
PD and PS: the value of the perimeter of the source and drain.
+: it means this line is the continuation of the previous line.
NRD: the number of squares of the drain diffusion for resistance calculations.

For example, the line:

```
m0 Out A Vdd Vdd p l=0.6u w=2.7u ad=11.745p as=11.745p
```

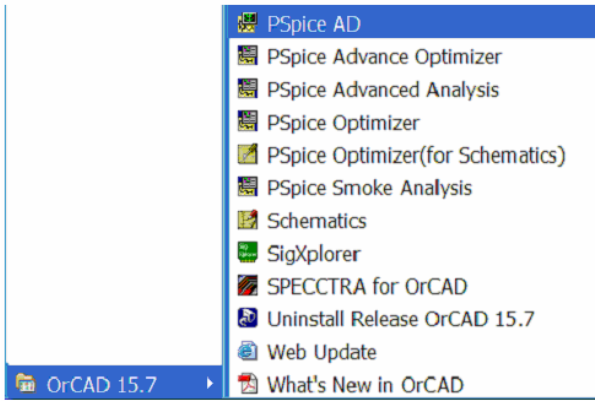
defines a transistor with name m0, the drain, gate, source and bulk terminals are connected to nodes Out, A, Vdd and Vdd separately. Its model name is p, which will be defined by Spice MOS parameter later. The area of the drain is $11.745\times10^{-12}m^2$, and the area of the source is $11.745\times10^{-12}m^2$. The parasitic capacitance of the source and drain will be considered based on the area of source and drain during SPICE simulation.

The netlist file offers an entire description about the whole circuit structure and its connection. It has all the information about the circuit which you will need for SPICE simulation. Here we can see that ICtrace(M) only extracts the information of transistors. In the future we can try to find some other extraction tools for better extraction of parasitic capacitance and resistance. You may send this netlist file as an email attachment, and download it in a windows based computer. You will need it for the OrCAD PSPICE simulation, which is installed in the Windows based computers.

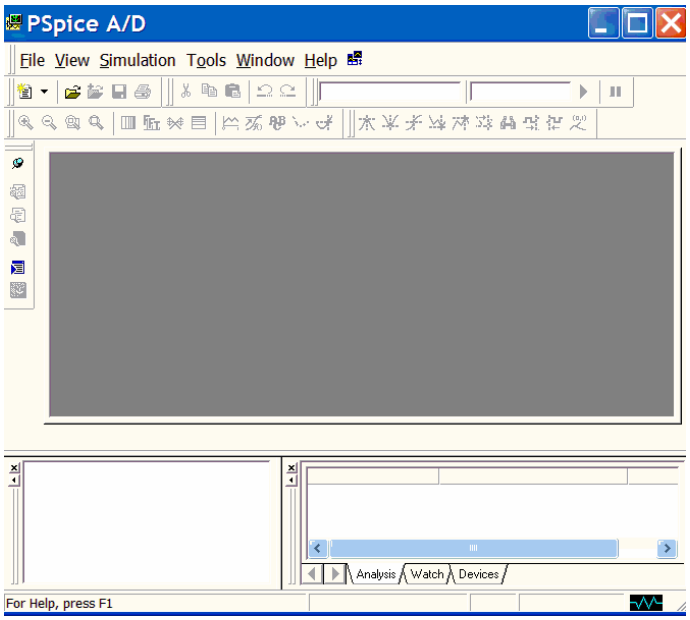
Section 2. OrCAD PSPICE Simulation for Power Analysis

OrCAD 15.7 (including PSPICE module) package is installed in Tech 113 computer lab machines. The detailed procedures for OrCAD PSPICE simulation for power analysis is shown as below.

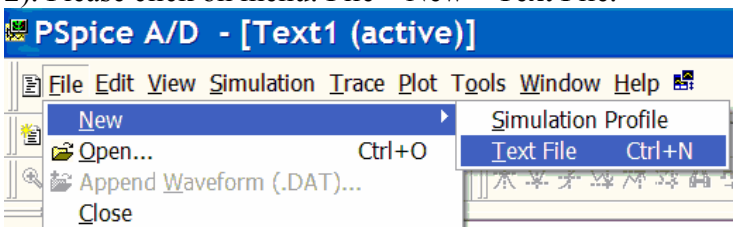
1). Please log into any PC there, and click on Windows menu: Start—All Programs—OrCAD 15.7—Pspice AD, as shown in below.



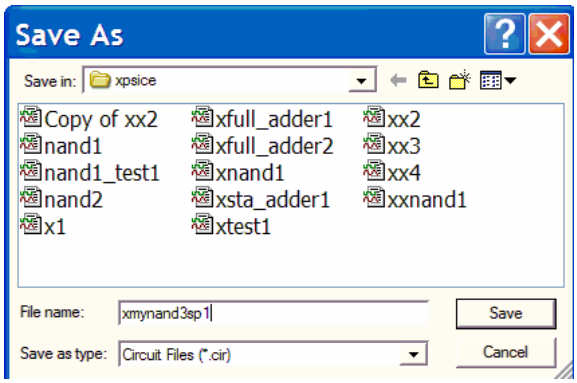
You will see the main interface as shown below.



2). Please click on menu: File—New—Text File.



3). You will see a blank text file is opened. First let's save the file. Please click on menu "File—Save As", you will see following interface.



In "Save in" line, please click to drag the bar and select the directory you want to save the file, and remember that directory. Then click on the drop menu of "Save as type", select the type as "Circuit Files (*.cir)". Type the File name you like in the "File name" line. Here we take the file name as xmynand3sp1, but you can use whatever name you like. Then click "Save". PSpice A/D takes ".cir" circuit file as input and can perform simulation on it.

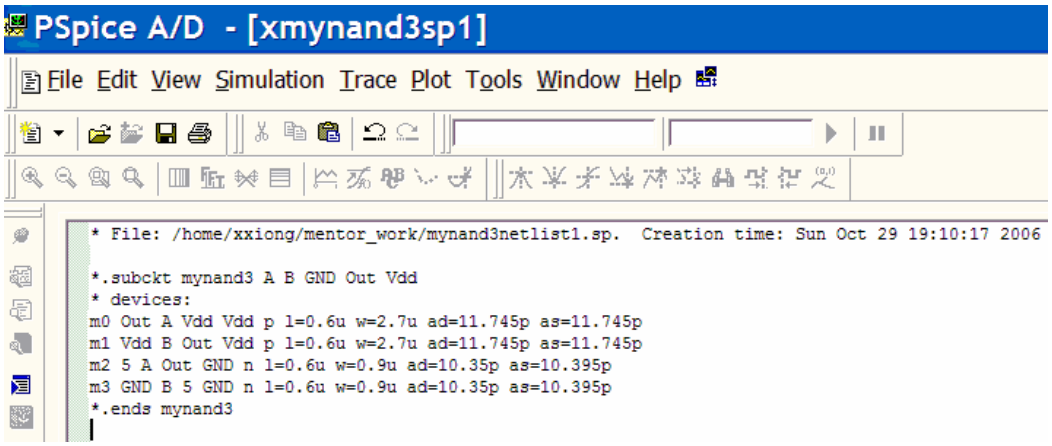
4). Now please copy and paste the contents from the extracted netlist file to here. The first line of a ".cir" spice circuit file must be a comment line and Spice simulation will automatically ignore the first line. Here our first line is a comment line (started with "*" sign). Thus it's ok. We only need the netlist of the circuit, thus please put a "*" sign in front of the line of

".subckt"

and the line of

".ends your_circuit_name".

In this way, they will not be considered in the Spice simulation. (You must comment out these two lines, otherwise you may have error in simulation.) Now your Spice .cir file should look like this.



5). The above is the netlist of the circuit. You still need to define the auxiliary circuit for power analysis, the SPICE parameters, input voltage signals, power rails (Vdd and Gnd), and tell Spice what simulations you need to perform. The circuit for power analysis simulation is shown in Figure 9, as we discussed in class.

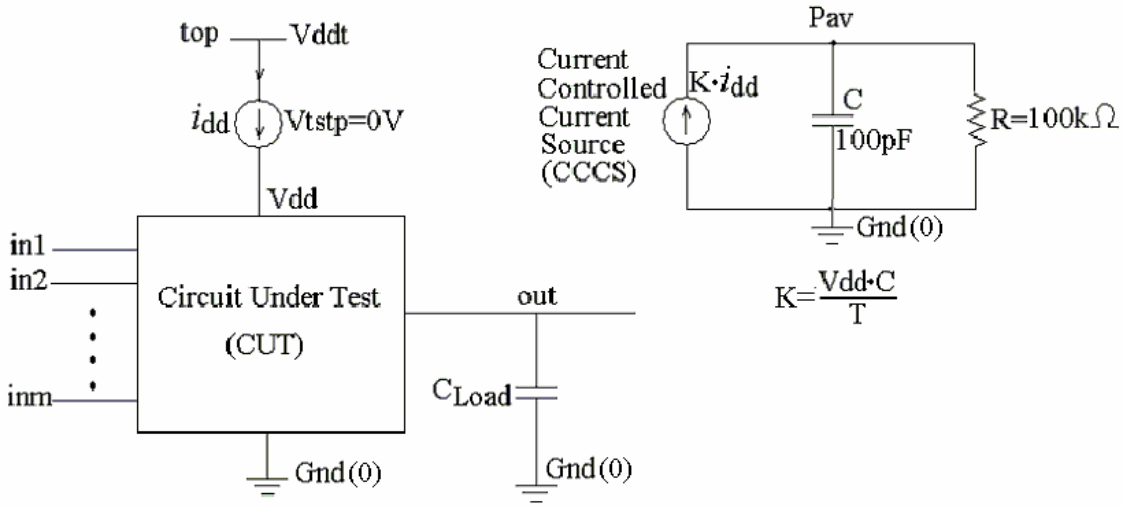


Figure 9. Circuit for power analysis of CMOS circuit

Now please copy and paste the following Spice codes into your .cir file. We will explain them one by one in detail.

```
* Auxiliary circuit for power analysis
Cp Pav 0 100p
Rp Pav 0 100k
Fp 0 Pav Vtstp 0.003125

*Output load capacitance if you have any
*Cload1 out GND 200fF

.MODEL n NMOS LEVEL  = 3
+ TOX    = 200E-10          NSUB   = 1E17            GAMMA  = 0.5
+ PHI    = 0.7              VTO    = 0.8             DELTA  = 3.0
+ UO     = 650              ETA    = 3.0E-6          THETA  = 0.1
+ KP     = 120E-6           VMAX   = 1E5             KAPPA  = 0.3
+ RSH    = 0                NFS    = 1E12            TPG    = 1
+ XJ     = 500E-9           LD     = 100E-9
+ CGDO   = 200E-12          CGSO   = 200E-12         CGBO   = 1E-10
+ CJ     = 400E-6           PB     = 1               MJ     = 0.5
+ CJSW   = 300E-12          MJSW   = 0.5

.MODEL p PMOS LEVEL  = 3
+ TOX    = 200E-10          NSUB   = 1E17            GAMMA  = 0.6
+ PHI    = 0.7              VTO    = -0.9            DELTA  = 0.1
+ UO     = 250              ETA    = 0               THETA  = 0.1
+ KP     = 40E-6            VMAX   = 5E4             KAPPA  = 1
+ RSH    = 0                NFS    = 1E12            TPG    = -1
+ XJ     = 500E-9           LD     = 100E-9
+ CGDO   = 200E-12          CGSO   = 200E-12         CGBO   = 1E-10
+ CJ     = 400E-6           PB     = 1               MJ     = 0.5
+ CJSW   = 300E-12          MJSW   = 0.5

*Define power rails
Vddt   top 0 5
```

```
Vtstp  top Vdd 0
Vss    GND 0 0

* Define input voltages of A and B
Va A 0 PWL(0 5 20N 5 20.1N 0  39.9N 0 40N 5 60N 5 60.1N 0 79.9N 0 80N 5
+ 100N 5 100.1N 0 119.9N 0 120N 5 140N 5 140.1N 0)
Vb B 0 PWL(0 0 20N 0 20.1N 5 60N 5 60.1N 0 100N 0 100.1N 5 140N 5 140.1N 0)

*Define transient simulation and probe voltage/current signals
.TRAN 5N 200N
.PROBE  V(*) I(*)

.end
```

Here we explain the above codes in details one by one.

i). `Rp Pav 0 100k`
`Cp Pav 0 100p`
`Fp 0 Pav Vtstp 0.003125`

The above codes define the auxiliary circuit for measuring the average power consumption during time period t=0~160ns.

The first line "Rp Pav 0 100k" defines a resistor named Rp, its positive and negative nodes are Pav and 0 (Gnd) separately. The resistance value is Rp=100kΩ.

The second line "Cp Pav 0 100p" defines a capacitance named Cp, its positive and negative nodes are Pav and 0 (Gnd) separately. The capacitance value is 100pF.

The third line "`Fp 0 Pav Vtstp 0.003125`" defines a current controlled current source named as Fp. It flows from positive end of node 0 (Gnd) to negative end of node Pav. The controlling current comes from the current flowing through the pseudo voltage source Vtstp, which will be introduced shortly. In this tutorial, we wish to simulate the average power consumption during time period of t=0→160ns. Thus the controlling coefficient K can be calculated from the following equation.

$$K = \frac{V_{dd} \cdot C}{T} = \frac{5 \times 100 \times 10^{-12}}{160 \times 10^{-9}} = 0.003125$$

ii). `*Cload1 out GND 200fF`

This line defines the output load capacitance Cload1=200fF connected between nodes "Out" and "GND". However, since we do not have output load capacitance in our case, we use "*" to comment out this line and it will not be executed in Spice simulation. If you have any output load capacitance, you can remove the comment sign "*" and input your load capacitance with this line.

iii). SPICE level-3 parameters for NMOS (model name as "n") and PMOS (model name as "p").

iv). `*Define power rails Vdd`

```
Vddt   top 0 5
Vtstp  top Vdd 0
Vss    GND 0 0
```

This section defines the power rails Vddt, Vss and a pseudo voltage source Vtstp in order to construct the current controlled current source $K \cdot i_{dd}$.

The first line "Vddt top 0 5" defines a 5V power rail named as Vddt, the voltage is between positive end node "top" and negative end node "Gnd".

The second line "Vtstp top Vdd 0" defines a 0V pseudo dc voltage source between the "top" node and the "Vdd" node of your circuit. The current i_{dd} flowing through the pseudo voltage source V_{tstp} controls the current controlled current source Fp defined previously.

The third line "Vss GND 0 0" defines the Gnd power rail named as "Vss", which is connected between the GND node of your circuit and node 0. The voltage level is 0V.

v). `* Define input voltages of A and B`

```
Va A 0 PWL(0 5 20N 5 20.1N 0  39.9N 0 40N 5 60N 5 60.1N 0 79.9N 0 80N 5
```

```
+ 100N 5 100.1N 0 119.9N 0 120N 5 140N 5 140.1N 0)
Vb B 0 PWL(0 0 20N 0 20.1N 5 60N 5 60.1N 0 100N 0 100.1N 5 140N 5 140.1N 0)
```

This section defines the input voltage stimuli for input ports A and B.
Va is defined as a PWL (Piecewise Linear) voltage between positive end (node A) and negative end (node 0). It's given by the time-voltage pairs in the bracket: (T1 V1 T2 V2 T3 V3 …). The piece-wise linear voltage is just the piecewise linear connection of these points.
The 4th line defines input voltage named as Vb between positive end (node B) and negative end (node 0). It's also piecewise linear voltage defined by the time-voltage pairs in the bracket.
Let's see an example about PWL voltage definition. If we wish to define the following PWL voltage Vf as shown in Figure 10.

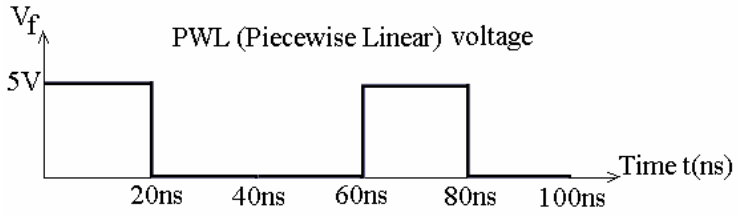


Figure 10. PWL signal you wish for voltage Vf

We need to introduce a small time interval (such as 0.1ns) to for each switching of the voltage. For example, we may define it as:

```
Vf F 0 PWL(0 5 20N 5 20.1N 0 60N 0 60.1N 5 80N 5 80.1N 0 100N 0)
```

Some students just directly give the voltage levels at each switching point without introducing this switching slope, as shown below.

```
Vf F 0 PWL(0 5 20N 0 60N 5 80N 0 100N 0)
```

You will find that you are not getting the above PWL voltage for Vf. Instead, you will get the following waveform for voltage Vf, as shown in Figure 11. Please notice the difference and be sure to introduce the short interval for describing the switching of the PWL voltage signal.

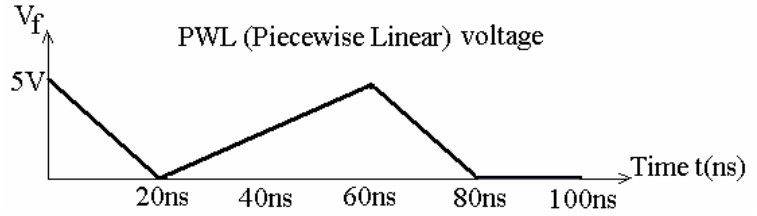


Figure 11. PWL waveform of voltage Vf due to improper definition

vi). `.TRAN 5N 200N`

```
.PROBE  V(*) I(*)
```

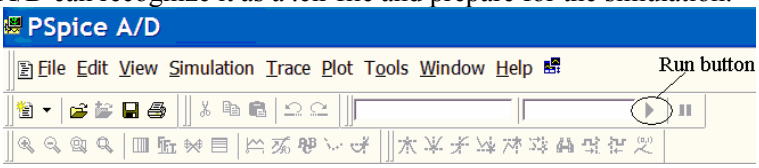
The first line asks PSPICE to perform a transient simulation with a step of 5ns and total simulation time of 200ns.
The second line ask PSPICE to probe all the voltage and current signals so that you can plot the curves for any voltage or current signal you are interested in.
vii). `.end`
This tells SPICE to finish the simulation.

Thus the total SPICE circuit file (mynand3netlist1.cir) should look like below. If you have a different circuit, you can still take Cp=100pF, Rp=100kΩ, but you need to decide your own time period T value and calculate your own K value for current controlled current source Fp. You also need to define your own input patterns for your circuit.

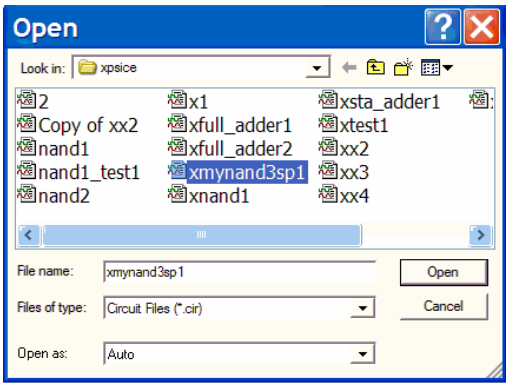
```
* File: /home/xxiong/mentor_work/mynand3netlist1.sp.  Creation time: Sun Oct
29 19:10:17 2006
```

```
*.subckt mynand3 A B GND Out Vdd
* devices:
m0 Out A Vdd Vdd p l=0.6u w=2.7u ad=11.745p as=11.745p
m1 Vdd B Out Vdd p l=0.6u w=2.7u ad=11.745p as=11.745p
m2 5 A Out GND n l=0.6u w=0.9u ad=10.35p as=10.395p
m3 GND B 5 GND n l=0.6u w=0.9u ad=10.35p as=10.395p
*.ends mynand3

* Auxiliary circuit for power analysis
Cp Pav 0 100p
Rp Pav 0 100k
Fp 0 Pav Vtstp 0.003125

*Output load capacitance if you have any
*Cload1 out GND 200fF

.MODEL n NMOS LEVEL  = 3
+ TOX    = 200E-10          NSUB   = 1E17            GAMMA  = 0.5
+ PHI    = 0.7              VTO    = 0.8             DELTA  = 3.0
+ UO     = 650              ETA    = 3.0E-6          THETA  = 0.1
+ KP     = 120E-6           VMAX   = 1E5             KAPPA  = 0.3
+ RSH    = 0                NFS    = 1E12            TPG    = 1
+ XJ     = 500E-9           LD     = 100E-9
+ CGDO   = 200E-12          CGSO   = 200E-12         CGBO   = 1E-10
+ CJ     = 400E-6           PB     = 1               MJ     = 0.5
+ CJSW   = 300E-12          MJSW   = 0.5

.MODEL p PMOS LEVEL  = 3
+ TOX    = 200E-10          NSUB   = 1E17            GAMMA  = 0.6
+ PHI    = 0.7              VTO    = -0.9            DELTA  = 0.1
+ UO     = 250              ETA    = 0               THETA  = 0.1
+ KP     = 40E-6            VMAX   = 5E4             KAPPA  = 1
+ RSH    = 0                NFS    = 1E12            TPG    = -1
+ XJ     = 500E-9           LD     = 100E-9
+ CGDO   = 200E-12          CGSO   = 200E-12         CGBO   = 1E-10
+ CJ     = 400E-6           PB     = 1               MJ     = 0.5
+ CJSW   = 300E-12          MJSW   = 0.5

*Define power rails
Vddt   top 0 5
Vtstp  top Vdd 0
Vss    GND 0 0

* Define input voltages of A and B
Va A 0 PWL(0 5 20N 5 20.1N 0  39.9N 0 40N 5 60N 5 60.1N 0 79.9N 0 80N 5
+ 100N 5 100.1N 0 119.9N 0 120N 5 140N 5 140.1N 0)
Vb B 0 PWL(0 0 20N 0 20.1N 5 60N 5 60.1N 0 100N 0 100.1N 5 140N 5 140.1N 0)

*Define transient simulation and probe voltage/current signals
.TRAN 5N 200N
.PROBE  V(*) I(*)

.end
```

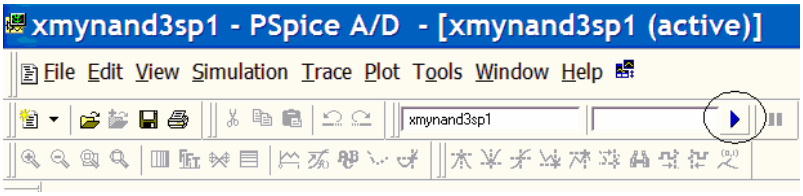
6). Now you are ready to simulate the file in PSPICE A/D. Please click on menu "File—Save" to save the file. You may see that the simulation button is grey and disabled at this time. This is because PSPICE AD still treats it as a text file and you cannot run the simulation now. You need to exit and reopen this .cir file, so that PSPICE A/D can recognize it as a .cir file and prepare for the simulation.



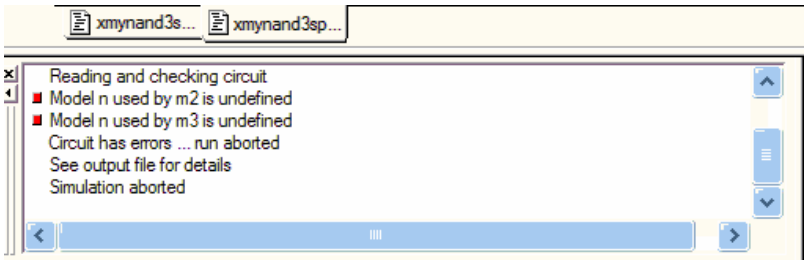
Now click menu "File—Close" to close the current file. Then click on menu "File—Open". In the pop-out window, click to and select the "Files of Types" as "Circuit Files(*.cir)". Find your .cir file name (here it's xmynand3sp1") and double click on it. You should see your .cir file is opened and now the simulation button is enabled.



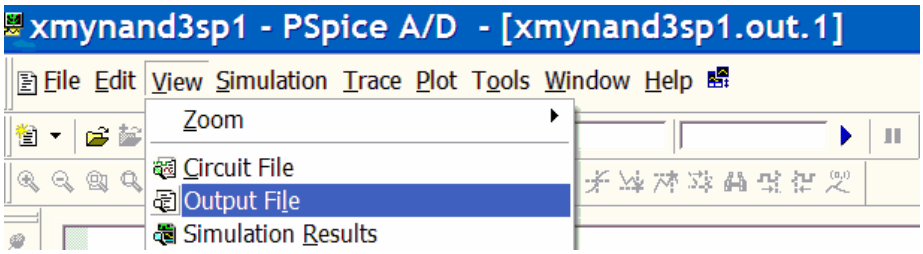
7). Now please click on the "Run" button in the top menu region, as circled below.



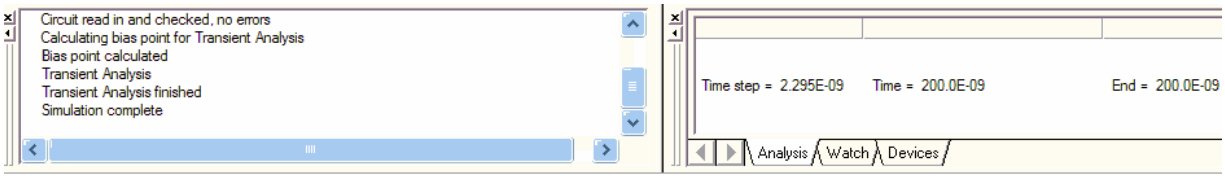
You can see that the simulation is running and will be done shortly. If there is some error with the simulation, you will see the information of "Simulation aborted". The errors will be marked with red dots in the bottom information window, as shown below.



If this is the case, you need to debug your design, correct the errors, and rerun the simulation. Please click on menu "View—Output File", as shown below. The output file will be displayed with detailed information about your simulation. The information in output file will be very helpful for you to trace the errors in your simulation.

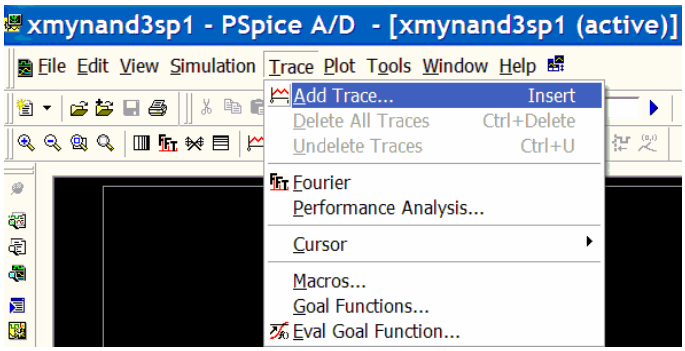


8). If your simulation is successful without error, you will see the information of "Simulation complete" in the bottom information window, as shown below.

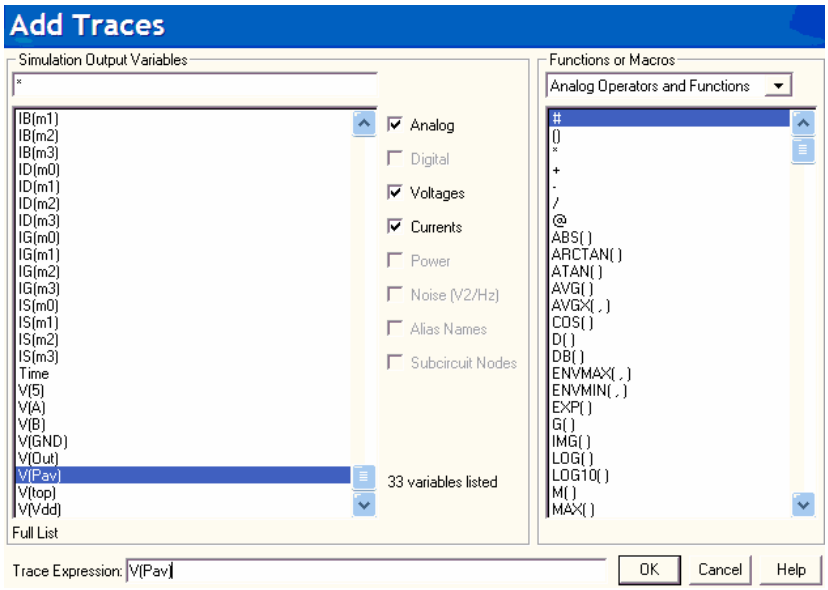


You can also see a black waveform window. You can click on the bottom to switch between your spice file and the waveform window.

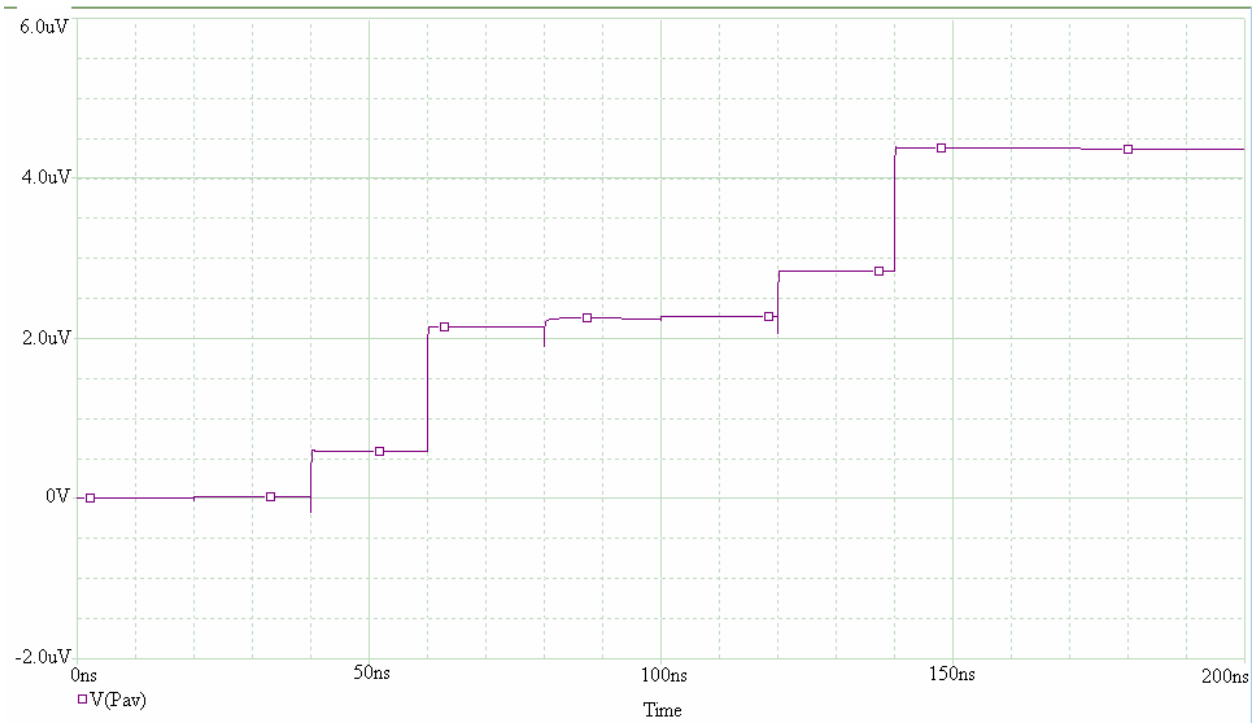
9). Now we are going to plot the voltage signals in the waveform window. In waveform window, please click on menu "Trace—Add Trace".



In the popup window as shown below, scroll down to find the voltage signal V(Pav), click on it and its name will appear on the "Trace Expression" line. Click OK.



10). Now you will see the waveform of the voltage signal V(Pav) at node Pav versus time t shown as below. Please note that here I captured the screen shot and intentionally inverted the color in Microsoft Paint, so that the major area is white instead of black. This can save your ink when you are printing it.



This is exactly the voltage $V(Pav)|_t$ at the node Pav, from which we can read the power consumption information of the circuit. Based on the curve, you can read and calculate the average power consumption as well as energy consumption of the circuit in any other time period t. The average power consumption Pavg(0→t) during any time period 0→t can be calculated from following equation (in value, not in unit):

$$P_{avg}(0 \to t) = \frac{V(Pav)|_t \times T}{t} \qquad (1)$$

If you use "V" as unit for voltage $V(Pav)|_t$, and use "sec" as unit for T and t, then the average power Pavg(0→t) is in unit of W. Please note that "t" and "T" are two different concepts in above equation. "t" is any time moment, while "T" is the time period you set to simulate the total average power when setting the coefficient for current controlled current source in equation:

$$K = \frac{V_{dd} \cdot C}{T} \qquad (2)$$

In this tutorial, the T is set as 160ns.

The total energy E(0→t) consumed during time period 0→t can be calculated as:

$$E(0 \to t) = V(Pav)|_t \times T \qquad (3)$$

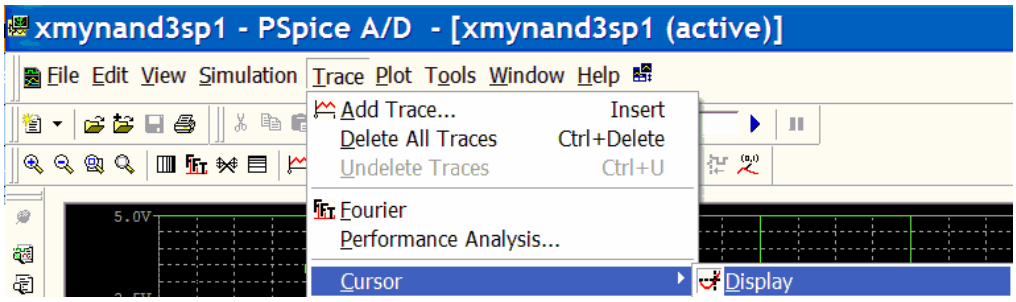
If you use "V" as unit for voltage $V(Pav)|_t$, and use "sec" as unit for T, then the energy consumption E(0→t) is in unit of J.

According to Equation (1) and (3), we can see that at time moment t=T,

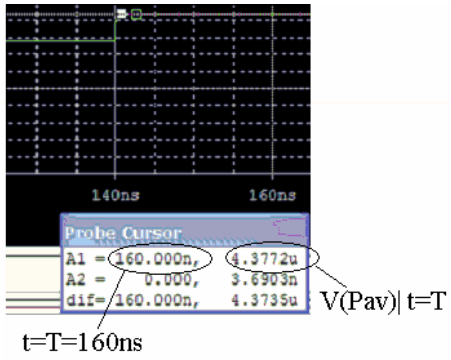
$$P_{avg}(0 \to T) = \frac{V(Pav)|_T \times T}{T} = V(Pav)|_T$$

We can see that at time moment t=T, the average power Pavg (in unit of W) of the circuit during time period 0→T is equal to the value of the voltage V(Pav) (in unit of volt) at node Pav. This is exactly how we measure the average power consumption of a CMOS circuit using PSPICE simulation.

11). Now let's read the average power at t=T=160ns. First click on menu "Trace—Cursor—Display".



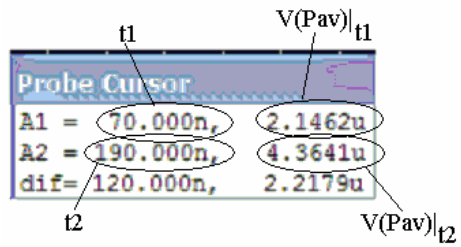
Click to move the cursor to be at time t=160ns. In the cursor window, you see the line of "A1=160.000n, 4.3772u". This indicates that at t=160ns, V(Pav)=4.3772μV. From this we can conclude that the average power consumption Pavg during t=0→T=160ns period is 4.3772μW.



The total energy consumption during t=0→T=160ns period is

$$E(0 \to T) = V(Pav)|_{t=T} \times T = 4.3772 \times 10^{-6} W \times 160 \times 10^{-9} s = 7.00 \times 10^{-13} J$$

12). We can also read the average power and energy consumption for any time moment other that t=T. For example, using the cursors, you can read the V(Pav) value at time t1=70ns and time t2=190ns, as shown below.



Thus we have

i). for the time period 0→t1=70ns, the average power is:

$$P_{avg}(0 \to t1) = \frac{V(Pav)|_{t1} \times T}{t1} = \frac{2.1462 \times 10^{-6} \times 160 \times 10^{-9}}{70 \times 10^{-9}} = 4.9056 \times 10^{-6} W$$

For the time period 0→t1=70ns, the total energy consumption is:

$$E(0 \to t1) = V(Pav)|_{t1} \times T = 2.1462 \times 10^{-6} \times 160 \times 10^{-9} = 3.43392 \times 10^{-13} J$$

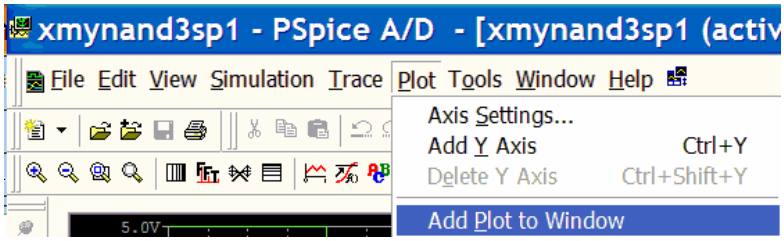
ii). For the time period 0→t2=190ns, the average power is:

$$P_{avg}(0 \to t2) = \frac{V(Pav)|_{t2} \times T}{t2} = \frac{4.3641 \times 10^{-6} \times 160 \times 10^{-9}}{190 \times 10^{-9}} = 3.675 \times 10^{-6} W$$

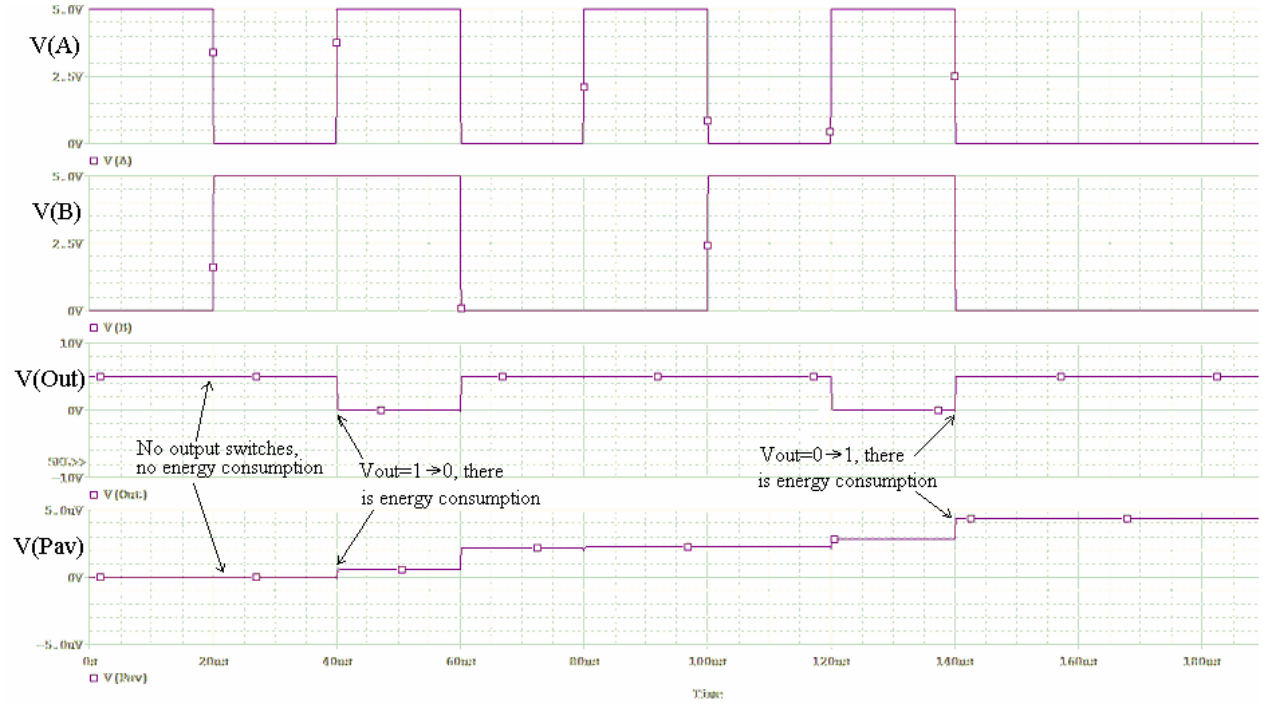
For the time period 0→t2=190ns, the total energy consumption is:

$$E(0 \to t2) = V(Pav)|_{t2} \times T = 4.3641 \times 10^{-6} \times 160 \times 10^{-9} = 6.95 \times 10^{-13} J$$

13). You may also want to see the input/output waveforms to verify the function of the circuit. Please click on menu "Plot—Add Plot to Window" to add 3 more plots to window.



Please click to select each plot to be active plot, and click on menu "Trace-Add Trace" to add voltage signals V(A), V(B) and V(Out) to each plot separately. Now you can see clearly the input-output waveforms of the circuit, and the relationship between the power consumption and the output state switches, as shown below.



From the waveforms of V(A), V(B) and V(Out) we can verify that the circuit has correct waveform as a nand gate. Further, from the waveform we can see clearly that the power consumption is related to the output state switching. During the stable period of an input pattern, or if the input switches but output does not change, there is no energy consumption. The circuit only consumes energy when there is an output state switching. This verifies our power model about VLSI circuits. That is, the majority of VLSI power consumption comes from switching power and short-circuit power, which occurs only when there is output state switching. Furthermore, we can see that when the output V(out) switches from 1 to 0, the energy consumption is less that when V(out) switches from 0 to 1. Why? Please try to explain this using our VLSI power models.

Reference

1. X. Xiong, "Mentor Graphics Tools Tutorial", University of Bridgeport.