

**Department of Electrical & Electronic Engineering
Imperial College London**

E4.20 Digital IC Design

Laboratory 1: Introduction to Electric & WinSPICE

Objectives

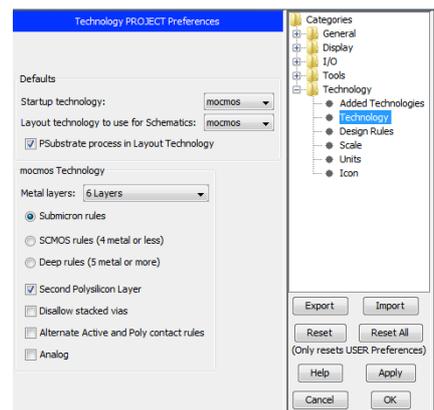
By the end of this laboratory session, you should be able to:

- *Capture the circuit schematic of a CMOS inverter using Electric.*
- *Connect multiple stages of the inverter for form a ring oscillator.*
- *Produce the SPICE netlist of your circuit.*
- *Simulate the circuit using WinSpice.*
- *Plot the result waveform in Electric and to make basic measurements.*

Exercise 1: Getting Started

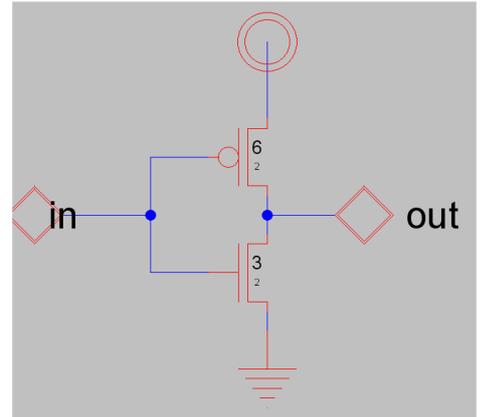
If you have not installed Electric on your computer, you must first:

1. Download and install Java Run-Time Environment from <http://www.java.com/>
 2. Download and install the latest version of Electric from <http://www.staticfreesoft.com/>.
- Create a directory to store all your lab1 files. Make sure that the directory name and all file names do not contain any spaces. All software tools used on this course were originally designed to run on Unix, where file names do not contain the space character.
 - Start Electric and create a new library called `lab1` with the command: **File > New Library ...**
 - Select the correct technology for the cells in this library with: **File > Preferences** then select **Technology**
 - Before you start, you must select the technology that you are designing for. We are going to use MOSIS Scalable CMOS technology. Since we are not making the chips, we can choose which ever is most convenient. For this course, we will be using the TSMC 0.18 μ sub-micron rules (SCN6M_SUBM). Therefore Select `mocmos`, 6 layers metal and Submicron rules as shown here.
 - Click on Scale and enter at the bottom 100 nanometers at the bottom of the form. This represents the dimension of lambda (λ) is 0.1 μ . (This is the lambda value corresponding to the 0.18 μ process.)
 - Use command: **Cell > New Cell...** and select schematic naming the new cell "`inv`". You should now see a palette menu appearing on the left with schematic symbols. The important symbols for this exercise are:



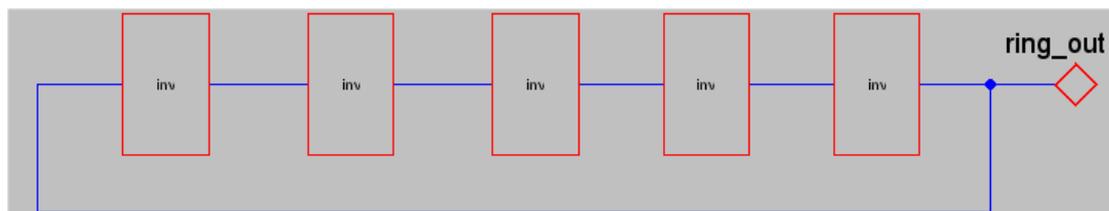
Exercise 2: Create an inverter schematic

- Our goal is to create an inverter schematic similar to this:
- Select the n-channel transistor on the palette, and click on the drawing area. Similarly, pick and place all components (called **nodes** in Electric) in the diagram.
- Hovering over nodes with the mouse highlights what will be selected when left clicked.
- Select the global signal symbol and rotate it with **Ctrl-J** until the small tail on the symbol (the terminal) is correctly oriented, also add the power and ground nodes.
- Wire up the components (wires are known as **arcs** in Electric). This is done by first left-clicking a component to select it. You will see that not only is the component (such as the n-transistor) highlighted, there is also a small + on the terminal closest to the mouse click location. The will be the source of the wire. You can now move the mouse cursor to the other end of the wire and right-click the mouse.
- Alternatively if you hold the right mouse button down and move the mouse, you will see a wire being draw and follow the cursor.
- Left clicking on nodes will highlight the paths which have been wired, use this to confirm that everything is connected as expected.
- After drawing the circuit structure, you need to enter the various parameters and labels. Double-click on the transistor and enter the width and length parameters (in λ) as shown. (Note that n-channel $W=3$, $L=2$; p-channel $W=6$, $L=2$.)
- Select the input node of the inverter and use command: **Export > Create Export ...** to make the input node visible. Enter the name `in` and specify that it is an input. Do similarly to the output node.
- This inverter will be used as a cell to form other circuits. It is therefore useful to create an icon view of this cell. You may do this with the command: **View > Make Icon View**.
- Save everything you have done so far in the library.



Exercise 3: Create a 5 stage ring oscillator

- Create a new schematic cell "`osc5`". Click on the Cell symbol in the palette menu, and select the icon view of the inverter cell as shown here. Place the inverter in the new drawing area.
- Use **Edit > Duplicate** command (Ctrl-M) to make 4 further instances of this inverter cell. With alternating left & right mouse clicks, wire the cells together in a ring configuration as shown below.
- Create a global node "`ring_out`" and export it as an output node.
- Save your work so far.



Exercise 4: Generate SPICE netlist

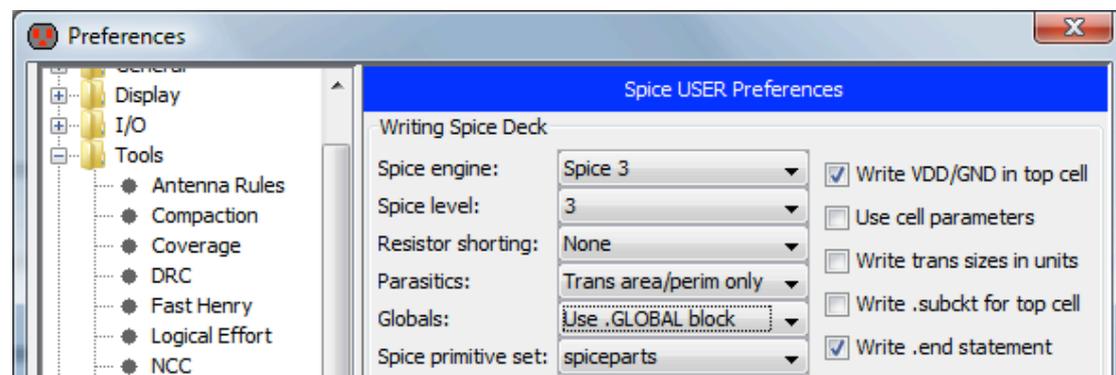
- You will now generate the SPICE netlist of the 5-stage ring oscillator. To do so, you must further specify: 1) the model of your transistors; 2) additional information to SPICE such as stimuli and supply sources; 3) the type of analysis to perform.
- The device model is already provided in the file "scmos18.mod" (available on the course web page: www.ee.ic.ac.uk/pcheung/teaching/ee4_asic). You must also specify the supply voltages and possibly the transient analysis you want to perform. This is best done by creating another file "user_append.spi" using a text editor containing the following lines:

```
* Specify power supply - 3.3v operation
vsupply vdd 0 3.3v
vgnd gnd 0 0v

* Specify transient analysis
.tran 0.01ns 5ns

* Dummy control block for SPICE 3
.control
.endc
```

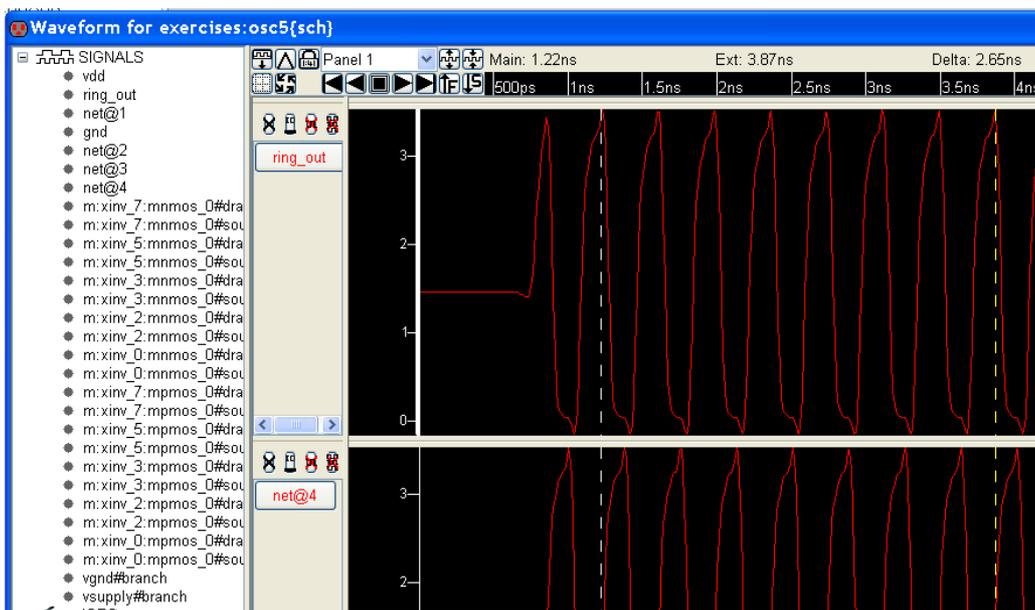
- These two files can be included automatically in the SPICE deck with the circuit netlist as Header and Trailer cards. This is done with command: **File > Preference > Tools > Spice** and then specify the file names in the form as appropriate. You must also specify the other entries as shown below: Spice3, level 3, and output format to Raw.



- Use command: **Tools > Simulation(Spice) > Write Spice Deck** to create a SPICE netlist "osc5.spi".
- Examine "osc5.spi" carefully with a text editor and make sure that you understand every statement here. Refer to the SPICE manual (links on course web page) if necessary.

Exercise 5: Simulate the ring oscillator using WinSPICE

- You will use WinSPICE to simulate the ring oscillator. If WinSPICE has not been installed on your computer, you may do so by first installing the evaluation version from the course website.
- Run WinSPICE (`wspice3.exe`) and change to the directory containing your files. Load the SPICE netlist file with command: `source osc5.spi`. Enter the command `run` to start the SPICE simulation.
- Use command: `plot ring_out` to see a plot of the ring oscillator output waveform.
- You can also generate a waveform file in “raw” format. To do so run the simulation again while creating a raw waveform file using command: `run osc5.raw`.
- You can now go back to Electric and import this waveform using command: **Tools>Simulation(Spice)>Plot Spice for this cell ...**
- You can now interactively plot the desired signals and measure the period of oscillation (See below).



Exercise 6: Extensions

- From the simulation waveform, it is clear that 5 stages of inverter do not provide sufficient delay round the loop to make a good ring oscillator. Modify your circuit to make an 11-stage (or longer) ring oscillator and repeat the simulation. Measure the effective delay of each inverter stage.
- Modify the width of the transistors in the inverter cell and see if it has any effect on the oscillation frequency of the ring oscillator.
- Create a two input NAND gate and replace the left-most inverter with this gate. Now you have a user controlled oscillator that you can switch on/off. Test and simulate this new 11-stage ring oscillator.

APPENDIX

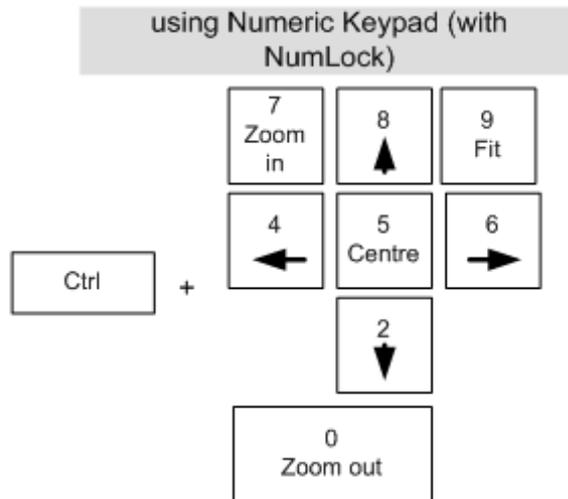
Electric Version 8.11 short-cuts

Use arrow keys to move objects left, right, up or down.

Pan and Zoom

Pan	↓	Ctrl+2
Pan	←	Ctrl+4
Pan centre		Ctrl+5
Pan	→	Ctrl+6
Pan	↑	Ctrl+8

Zoom in	Ctrl+7
Zoom to fit	Ctrl+9
Zoom out	Ctrl+0



Basic Short-cuts

Command	Keystroke
Un-DO	Ctrl + Z
Re-DO	Ctrl + Y
Cut	Ctrl + X
Copy	Ctrl + C
Paste	Ctrl + V
Select ALL	Ctrl + A
Open library	Ctrl + O
New cell	Ctrl + N
Duplicate	Ctrl + M
Save all	Ctrl + S
Quit	Ctrl + Q

Other short-cuts

Command	Keystroke
Up hierarchy	Ctrl + U
Down hierarchy	Ctrl + D
Down h. In place	D
Export	Ctrl + E
Inform object prop.	Ctrl + I
Toggle Grid	Ctrl + G
Toggle Measure	M
Place cell instance	N

Design Rule Checks

Command	Keystroke
DRC Hierarchy	F5
Show next error	>
Show last error	<

WinSpice Interactive Commands(Frequently used commands shown in **bold**.)

ac [(DEC OCT LIN) N Fstart Fstop]	Do an ac analysis.
alias [[word] alias]	Define an alias.
alter devspecs : parmname value	Alter device parameters.
asciplot plotargs	Produce ascii plots.
bug	Report a WinSpice bug.
cd [directory]	Change working directory.
cdump	Dump the current control structures.
compose var parm=val ...	Compose a vector.
cross vecname n [vector ...]	Make a vector 'vecname' from the nth item in the list of vectors.
dc [args as for .dc]	Do a dc analysis.
define [[func (args)] stuff]	Define a user-definable function.
deftype spec name pat ...	Redefine vector and plot types.
delete [all] [break number ...]	Delete breakpoints and traces.
destroy [all][plotname] ...	Throw away all the data in the plot.
diff plotname plotname [vec ...]	'diff' two plots.
display	Display vector status.
disto [args as for .disto]	Do an distortion analysis.
dump	Print a dump of the current circuit.
echo [stuff ...]	Print stuff.
edit [filename]	Edit a spice deck and then load it in.
four fund_freq vector ...	Do a fourier analysis of some data. See 'fourier'.
fourier fund_freq vector ...	Do a fourier analysis of some data.
function [[func (args)] stuff]	Define a user-definable function. See 'define'.
hardcopy file plotargs	Produce hardcopy plots.
help [command name] ...	help.
history [-r] [number]	Print command history.
iplot [all] [node ...]	Incrementally plot a node.
let varname = expr	Assign vector variables.
linearize [vec ...]	Convert plot into one with linear scale.
listing [logical][physical][deck][expand]	Print the current circuit.
load file ...	Load in data.
noise [args as for .noise]	Do a noise analysis.
oldhelp [command name] ...	Print help.
op [args as for .op]	Determine the operating point of the circuit.
plot expr ... [vs expr] [xl xlo xhi] [yl ylo yhi]	Plot things.
print [col] expr ...	Print vector values.
pz [args as for .pz]	Do a pole/zero analysis.

quit	Quit WinSpice.
rawfile [rawfile][OFF]	Send subsequent simulation output to a rawfile.
reset	Terminate a simulation after a breakpoint (formerly 'end').
reshape vector ... [shape]	change the dimensions of a vector.
resume	Continue after a stop.
run [rawfile]	Run the simulation as specified in the input file.
rusage [resource ...]	Print current resource usage.
save [all] [node ...]	Save a spice output.
sens [args as for .sens]	Do a sensitivity analysis.
set [option] [option = value] ...	Set a variable.
setcirc [circuit name]	Change the current circuit.
setplot [plotname]	Change the current working plot.
setscale [vecname]	Change default scale of current working plot.
settype type vec ...	Change the type of a vector.
shell [args]	Fork a shell, or execute the command.
shift [var] [number]	Shift argv or the named list var to the left.
show devices ...	parameters ...
showmod models ...	parameters ...
source file	Source a WinSpice file.
spec start_freq stop_freq step_freq vector ...	Create a frequency domain plot.
status	Print the current breakpoints and traces.
step [number]	Iterate number times, or one.
stop [stop args]	Set a breakpoint.
strcmp varname s1 s2	Set \$varname to strcmp(s1, s2).
temp [temp] ...	Define a temperature or a list of temperatures.
tf [output_node input_source]	Do a transfer function analysis.
trace [all] [node ...]	Trace a node.
tran [args as for .tran]	Do a transient analysis.
transpose varname ...	Perform matrix transposition on multi-D vectors.
tutorial [subject] ...	Hierarchical documentation browser.
unalias word ...	Undefine an alias.
undefine [func ...]	Undefine a user-definable function.
unlet varname ...	Undefine vectors.
unset varname ...	Unset a variable.
version [number]	Print the version number.
view view ...	Dummy command for IsSpice compatibility.
where	Print last non-converging node or device
write [file [expr ...]]	Write data to a file.