

Name: Nguyễn Tuấn Dũng

ID: 21520746

Class: CE222.O23.MTCL

DIGITAL INTEGRATED CIRCUIT DESIGN LAB 2'S REPORT

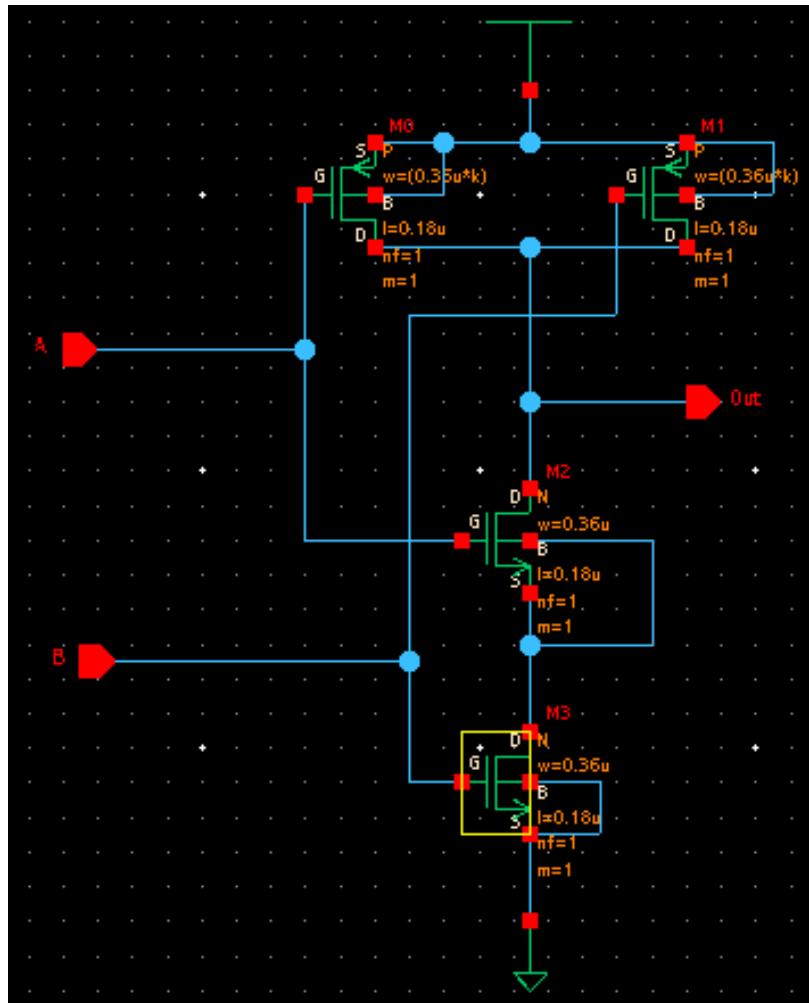
Section 1 : Schematic design and Nand2 simulation (Front-end):

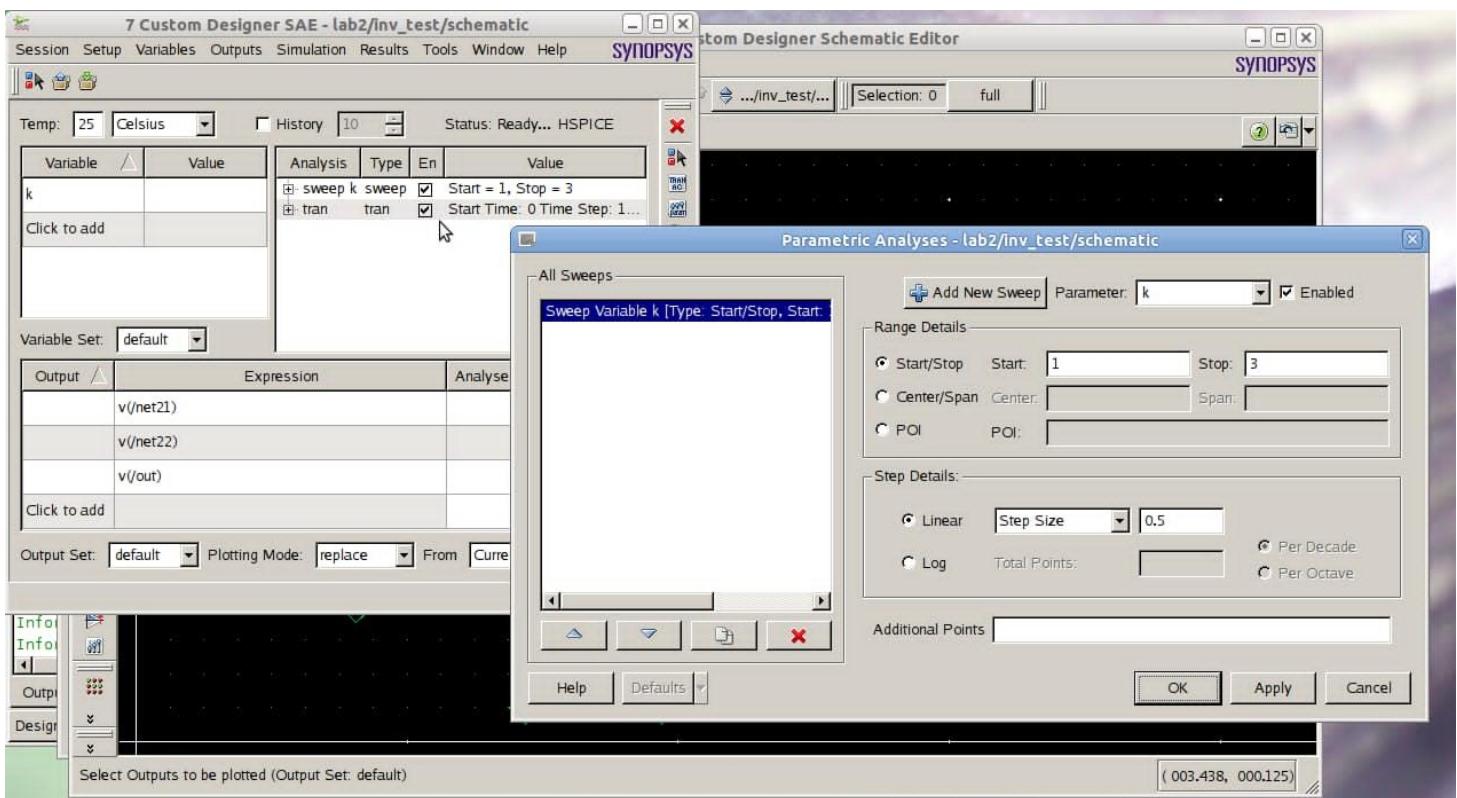
1. Width and Length measurements

W/L table of each NMOS and PMOS, explaining how to choose:

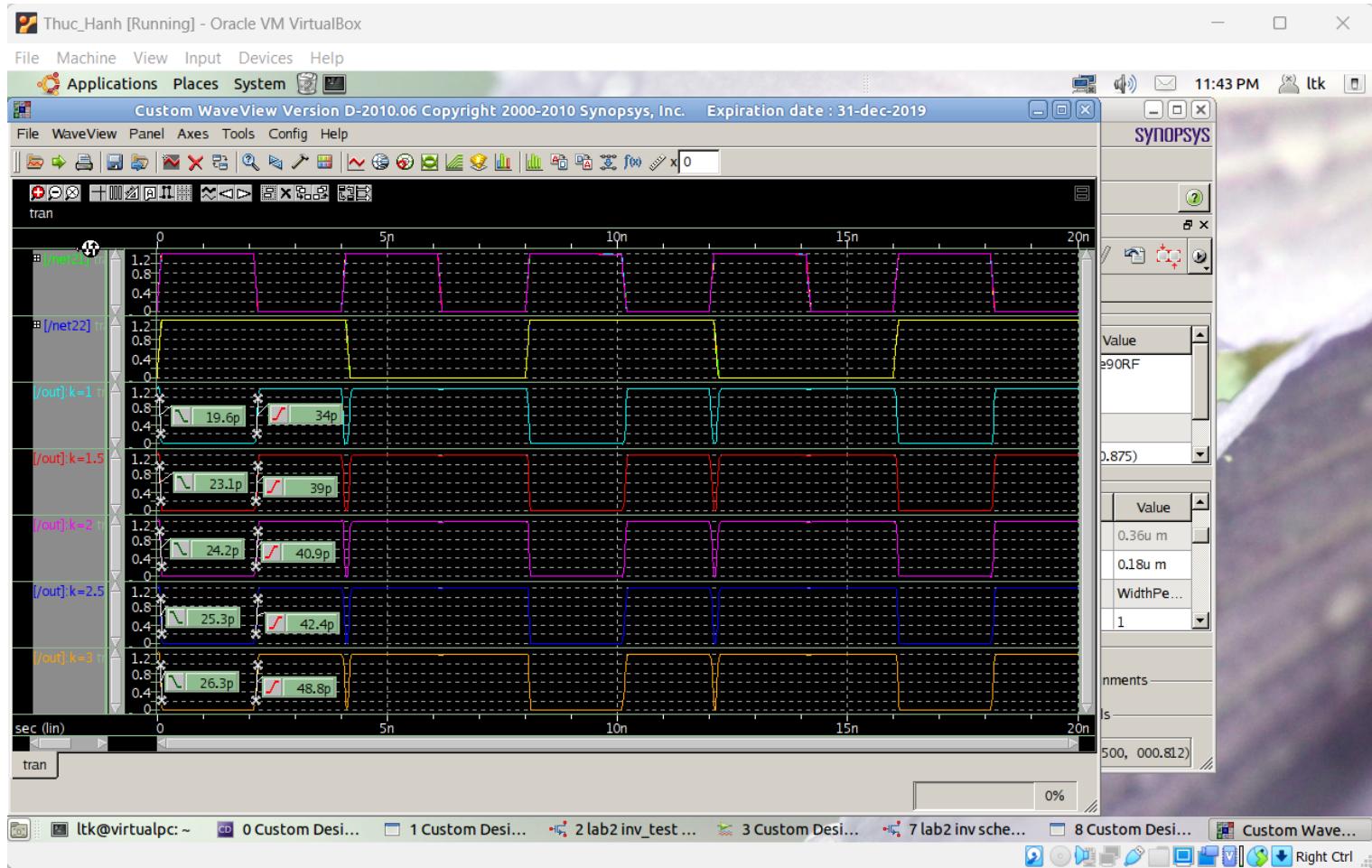
PMOS	NMOS
$W_p = 0.36\mu * k$	$W_n = 0.36\mu$
$L_p = 0.18\mu$	$L_n = 0.18\mu$

Schematic:

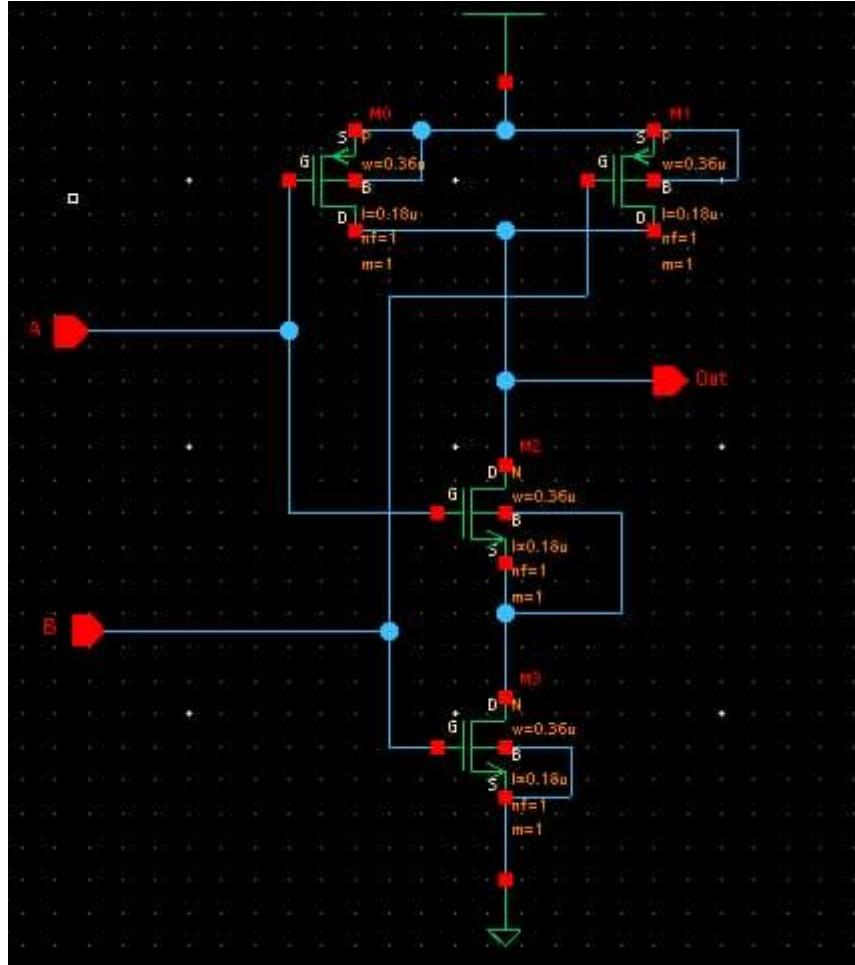




_ Using k as a variable to check the simulation results for the NAND2 gate, let k run from 1 to 3 with stepsize of 0.5, we have the following waveform:

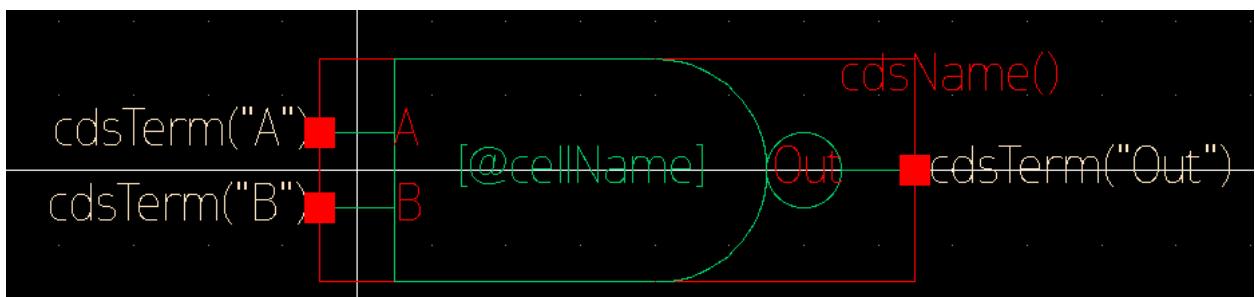


=> We see that when $k = 1$, T_{fall} and T_{rise} have the least different values, so we choose the value at $k = 1$. => $W_p = 0.36u$

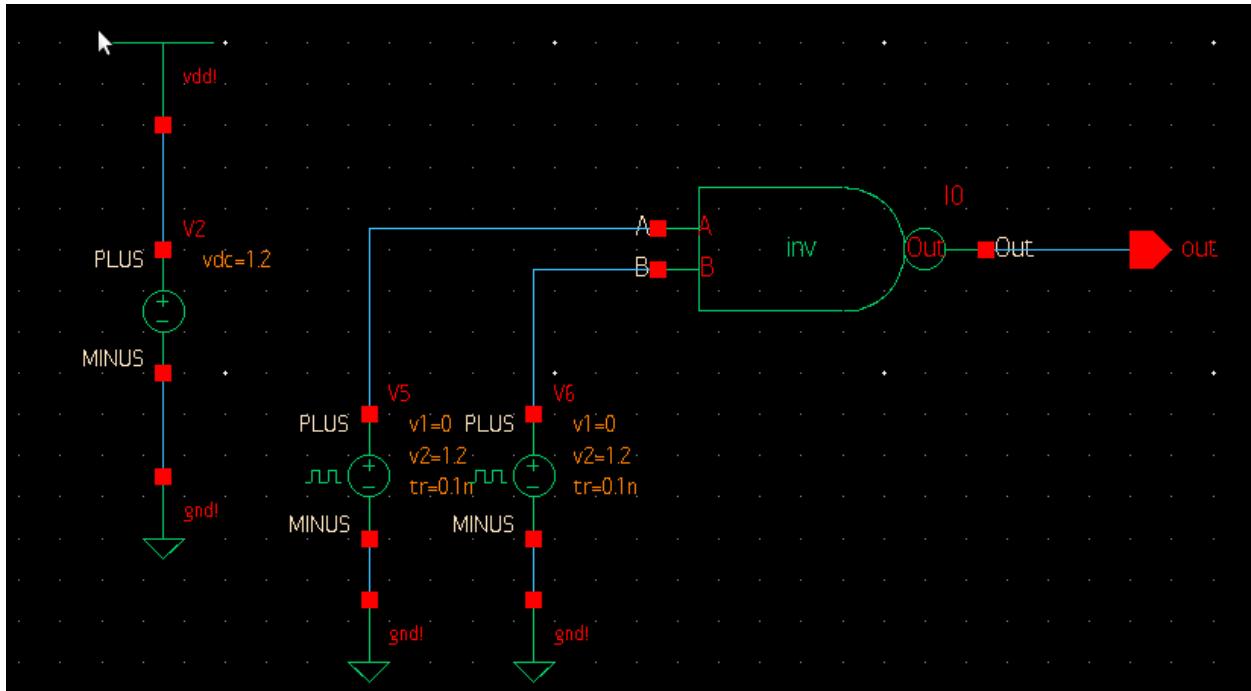


2 . Create Symbol and Testbench

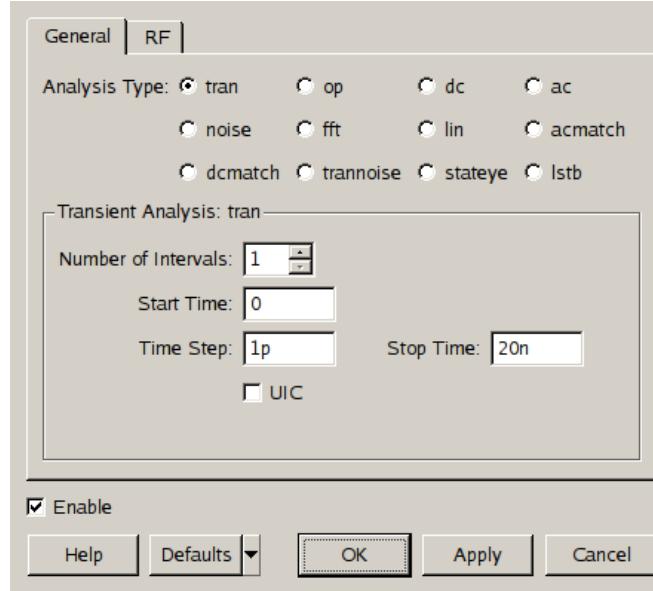
_ To facilitate simulation or reuse to design another circuit, people will create a symbol for the circuit. This symbol will represent the circuit in the design.



_ To create a testbench circuit for the Nand2, we need to make another circuit to create an environment to test the operation of the circuit.



- Name the pin for the Nand2 as: output - In there:
 - + V2 is in the library analogLib => vdc. Choose vdc = 1.2v
 - + V5 is in the library analogLib => vpulse.
 - + V6 is in the library analogLib => vpulse.
- => We choose V5 to have a Rise Time of 2ns and a Fall Time of 4ns while V6 has a Rise Time of 4ns and a Fall Time of 6ns to be able to simulate all cases of Nand2.
- => Set the delay to 0.001ns so that the delay time is small and does not affect the simulation.
- _ Choose the simulation time:



_ A notification window appears, in the hspice.st0 window the following information displays the results simulation was successful (Result of simulation waveform by SAE):

```

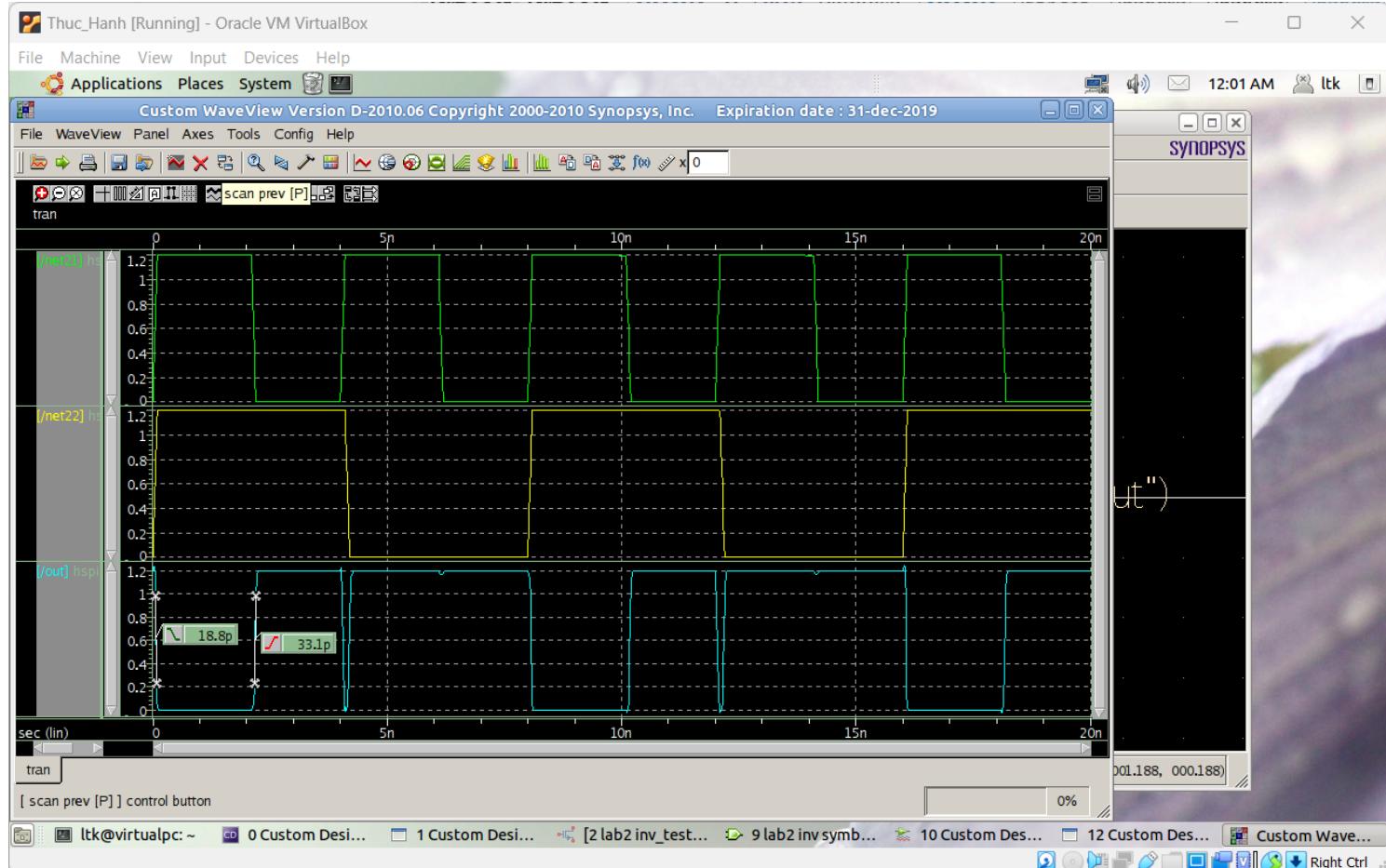
12 Custom Designer Text Viewer - /home/ltk/simulation/lab2/inv_test/schematic/HSPICE/nominal
File Edit View Window Help

hspice.lis run.log hspice.st0
option artist = 2.00
option psf = 2.00
option opfile = 1.00
option split_ = 1.00
init: end read circuit files, cpu clock= 0.00E+00 memory= 150 kb
init: begin check errors, cpu clock= 0.00E+00
init: end check errors, cpu clock= 0.00E+00 memory= 150 kb
init: begin setup matrix, pivot= 0 cpu clock= 0.00E+00
    establish matrix -- done, cpu clock= 0.00E+00 memory= 153 kb
    re-order matrix -- done, cpu clock= 0.00E+00 memory= 154 kb
init: end setup matrix, cpu clock= 0.00E+00 memory= 167 kb
dcop: begin dcop, cpu clock= 0.00E+00
dcop: end dcop, cpu clock= 0.00E+00 memory= 167 kb tot_iter= 7
sweep: tran tran0 begin, stop_t= 2.00E-08 #sweeps=**** cpu clock= 0.00E+00
tran: time= 2.1000E-09 tot_iter= 49 conv_iter= 19 cpu clock= 0.00E+00
tran: time= 4.0000E-09 tot_iter= 88 conv_iter= 33 cpu clock= 0.00E+00
tran: time= 6.1000E-09 tot_iter= 168 conv_iter= 60 cpu clock= 1.00E-02
tran: time= 8.0000E-09 tot_iter= 207 conv_iter= 77 cpu clock= 1.00E-02
tran: time= 1.0100E-08 tot_iter= 249 conv_iter= 91 cpu clock= 1.00E-02
tran: time= 1.2000E-08 tot_iter= 294 conv_iter= 108 cpu clock= 1.00E-02
tran: time= 1.4100E-08 tot_iter= 371 conv_iter= 133 cpu clock= 1.00E-02
tran: time= 1.6000E-08 tot_iter= 412 conv_iter= 150 cpu clock= 1.00E-02
tran: time= 1.8100E-08 tot_iter= 475 conv_iter= 172 cpu clock= 1.00E-02
tran: time= 2.0000E-08 tot_iter= 520 conv_iter= 189 cpu clock= 1.00E-02
sweep: tran tran0 end, cpu clock= 1.00E-02 memory= 168 kb
>info:      **** hspice job concluded
lic: Release hspice token(s)

Find: Next Previous Match Case Regexp

```

_ To see the waveform of inverter circuit we plot the input and output of circuit, a program called WaveView displays simulation results as shown below:



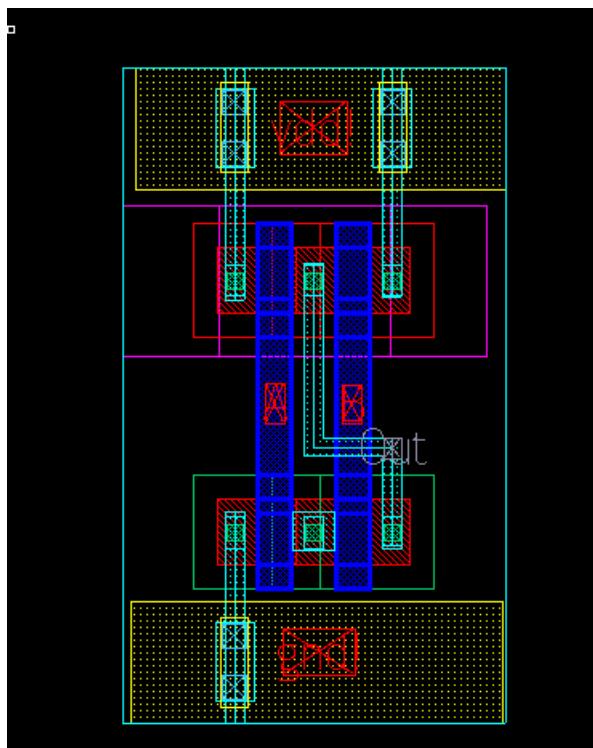
_ Explain: The green signal is A and the yellow signal is B. The bottom signal is the value after performing the Nand operation of A and B. 1 signal period of B is equal to 2 signal periods of A. Looking at the waveform, we draw the following table:

A	B	Nand2
1	1	0
0	1	1
1	0	1
0	0	1

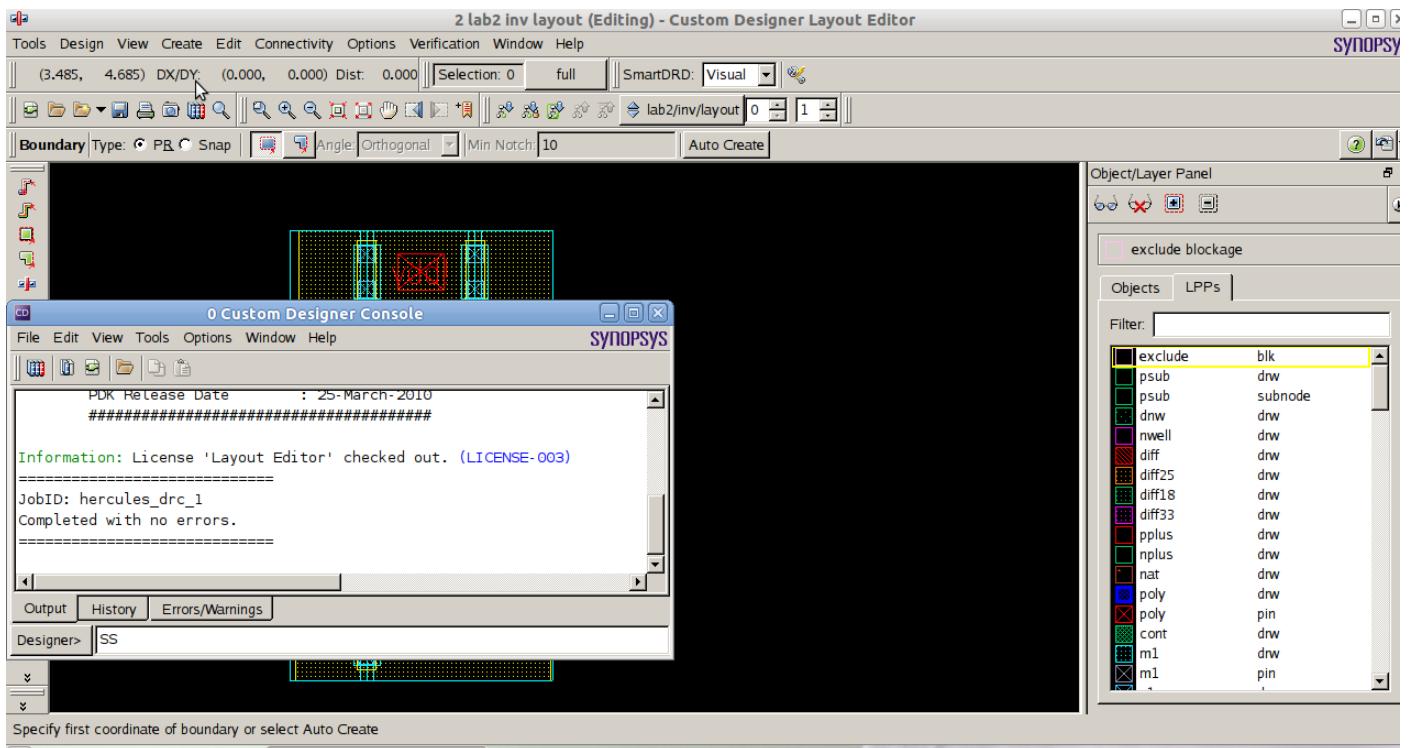
=> The Nand2 we designed is running correctly

Section 2 : Layout Nand2 (Back-end)

1. Layout Nand2



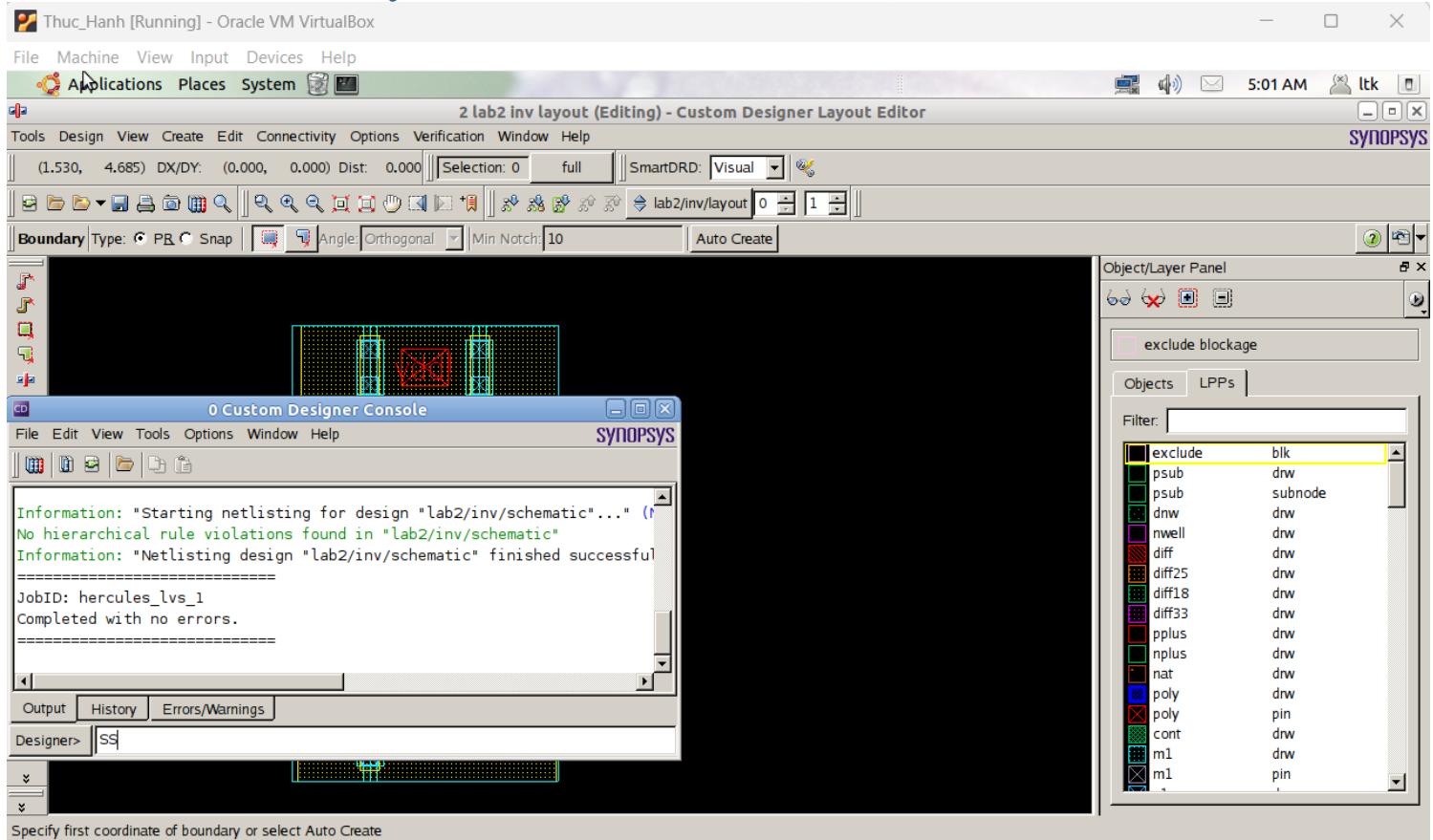
2. Check Design Rules Check (DRC)



_ This step uses the DRC function to check whether the design complies with the design rules or not.

=> **There are no DRC errors.**

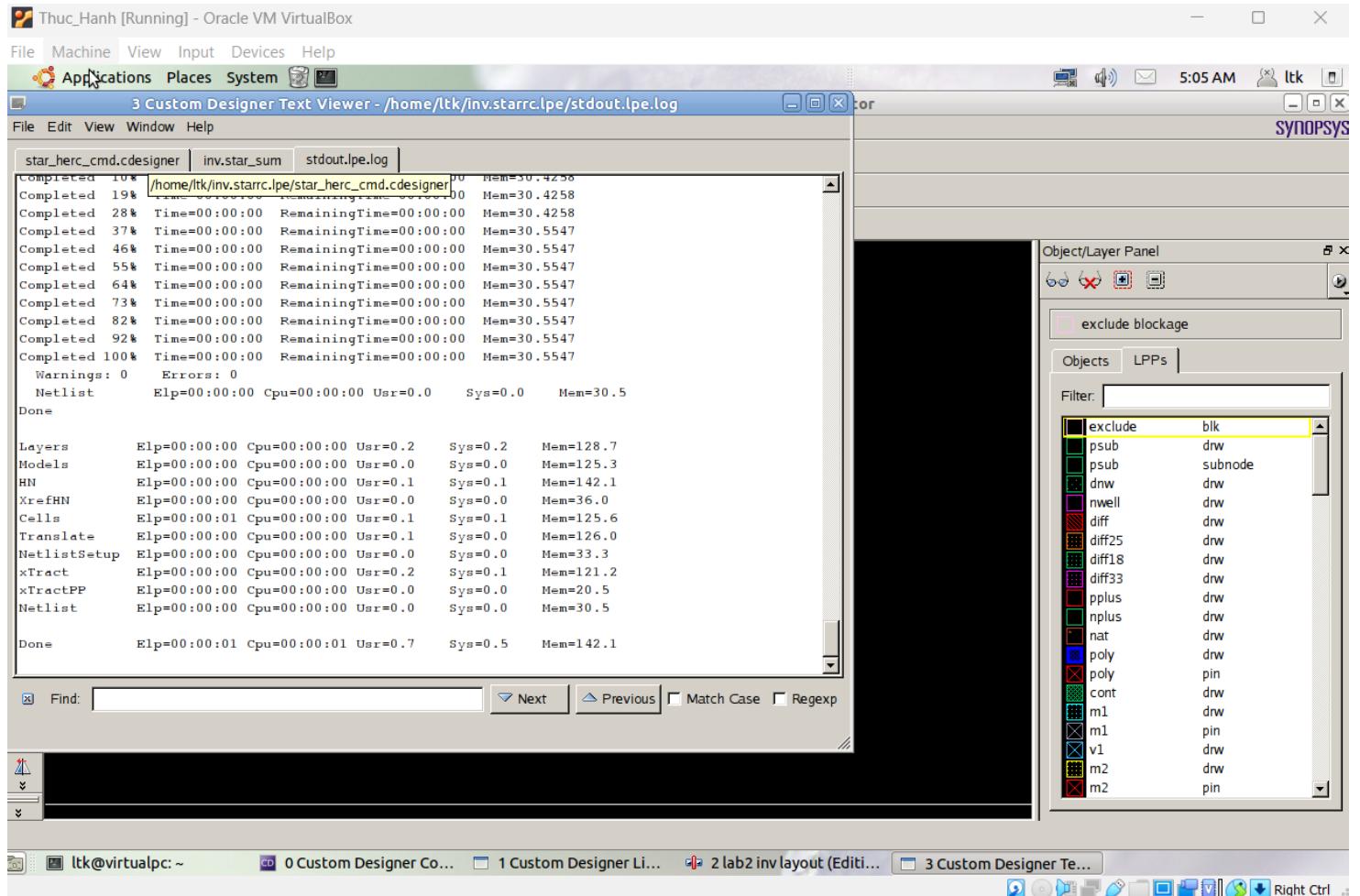
3. Check Layout Versus Schematic (LVS)



_ This step checks whether the layout is designed according to the design (schematic).

=> **The layout is designed according to the design (schematic) and does not appear errors.**

4. Layout Parasitic Extraction (LPE)



_ This step extracts the parasitic load on the circuit including capacitors and resistors.

=> **There are no LPE errors. When checked LPE had no errors then the capacitors and resistor parasitics in the layout will be displayed in the source.sp file.**

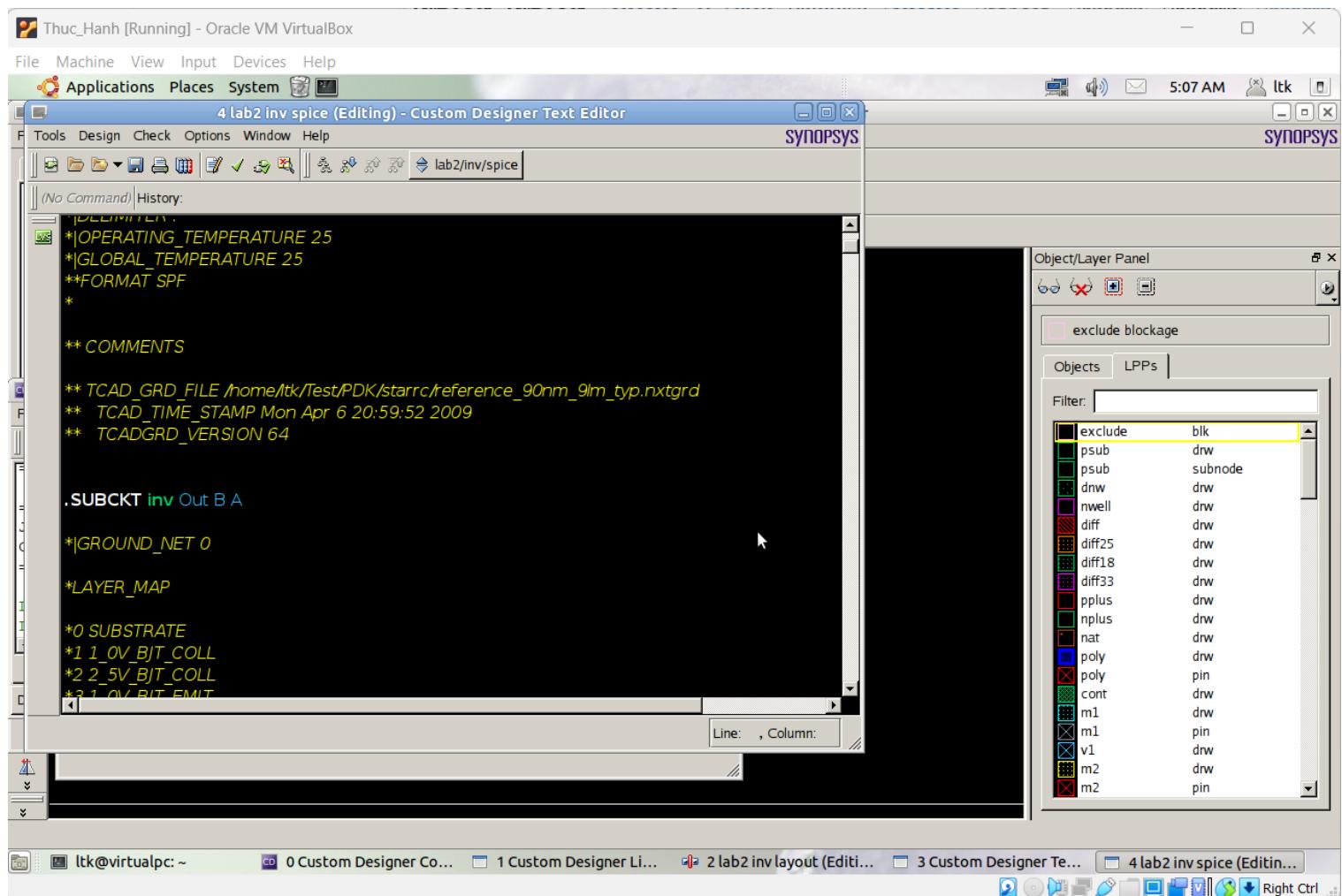
The screenshot shows a window titled "source (~/L2) - gedit" containing a text editor with a toolbar at the top. The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The toolbar contains icons for Open, Save, Undo, Redo, Cut, Copy, Paste, Find, and Replace. The main text area is titled "source" and contains the following SPICE code:

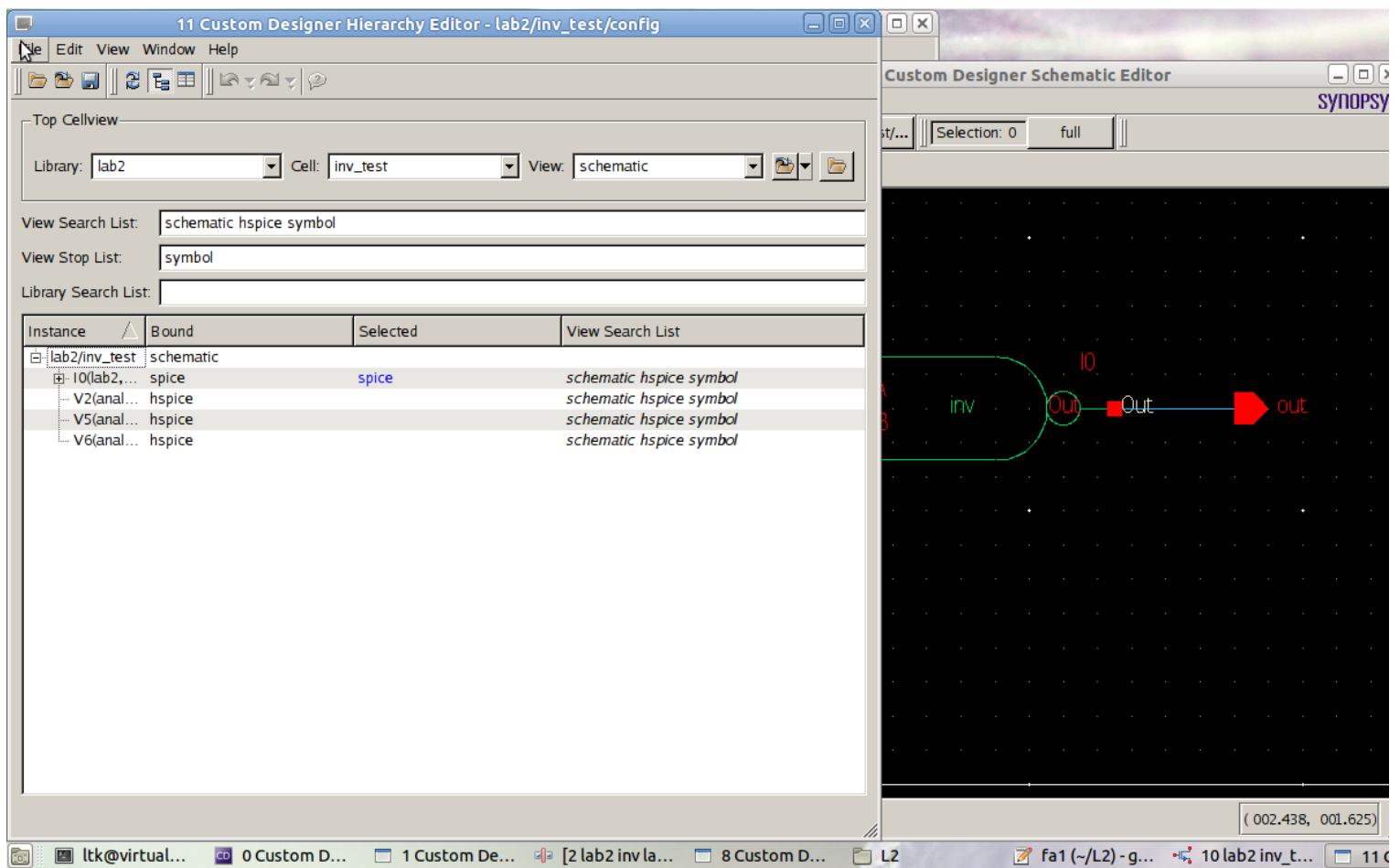
```
/*
*|DSPF 1.3
*|DESIGN inv
*|DATE "Mon May 22 05:05:20 2017"
*|VENDOR "Synopsys"
*|PROGRAM "StarRC"
*|VERSION "D-2010.06"
*|DIVIDER |
*|DELIMITER :
*|OPERATING TEMPERATURE 25
*|GLOBAL TEMPERATURE 25
**FORMAT SPF
*
** COMMENTS
**
** TCAD_GRD_FILE /home/ltk/Test/PDK/starrc/reference_90nm_9lm_typ.nxtgrd
**   TCAD_TIME_STAMP Mon Apr 6 20:59:52 2009
**   TCADGRD_VERSION 64
**
.SUBCKT inv vdd! gnd! Out B A
*
*|GROUND_NET 0
*
*LAYER_MAP
*
*0 SUBSTRATE
*1 1_0V_BJT_COLL
*2 2_5V_BJT_COLL
*3 1_OV_P1T_EMTR
```

At the bottom of the window, there are status indicators: Plain Text, Tab Width: 8, Ln 1, Col 1, and INS.

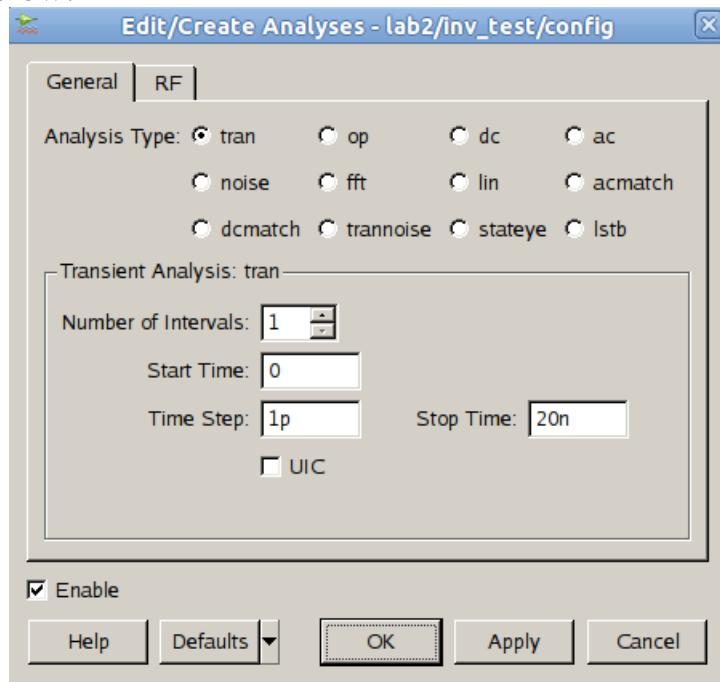
5. Simulate PostLayout and SEA simulation:

Following a successful extraction of parasitic in the layout, we obtain the output.sp file, which is a hspice file used for PostLayout simulation. Conducting PostLayout simulation enables us to achieve more precise results by accounting for the effects of resistors and capacitors. Then we will generate Spice files within Nand2 cells and substituting the contents of the spice.spc file with those of the output.sp file.

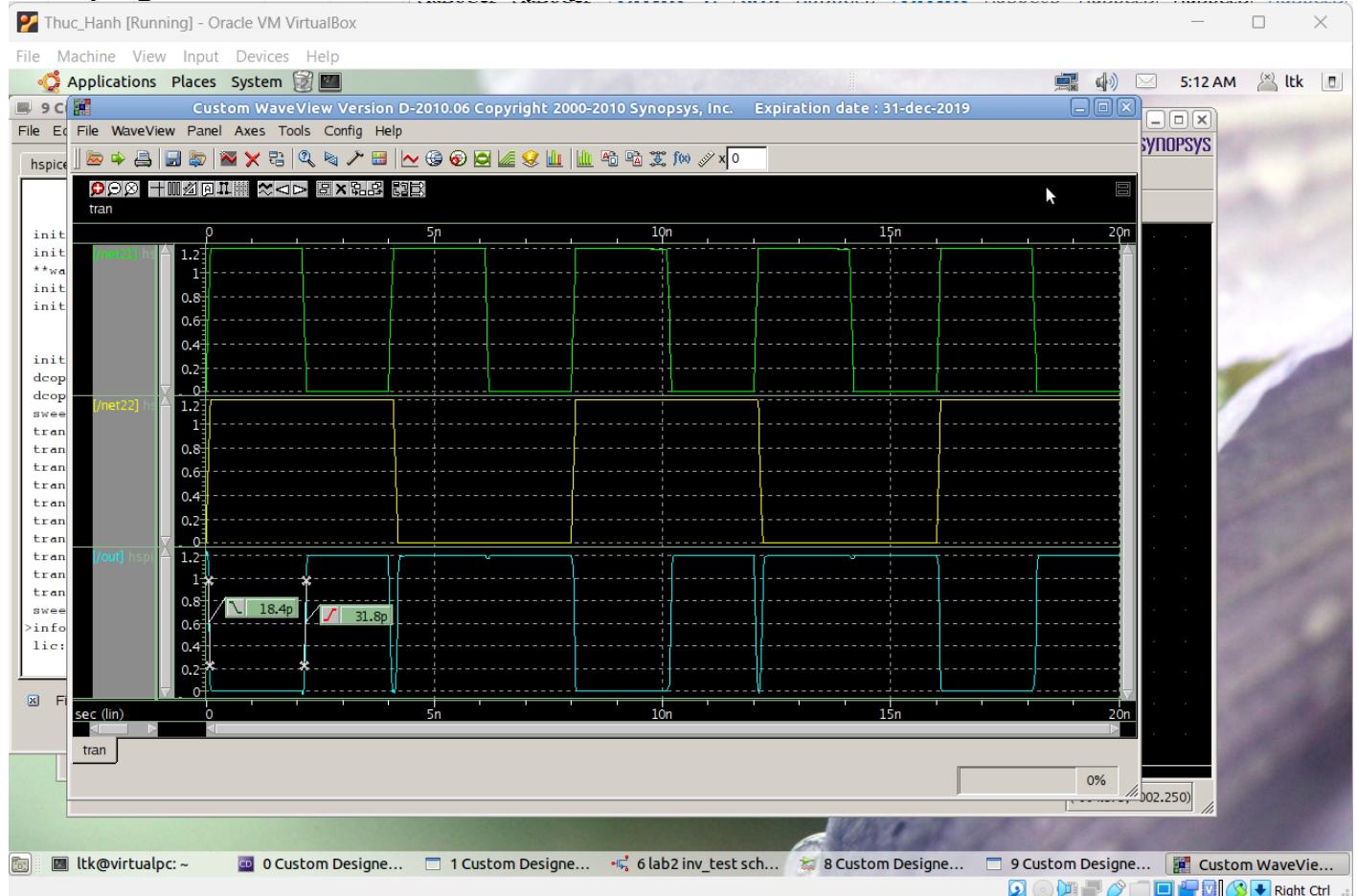




_ Then we will simulate PostLayout using SAE with parameters and setup model file are below:



- To see the waveform of inverter circuit we plot the input and output of circuit, a program called WaveView displays simulation results as shown below:



=> In this waveform, we can see the circuit simulation results have shown the correct properties of the Nand2 between the relationship of input and output

_ Explain: The green signal is A and the yellow signal is B. The bottom signal is the value after performing the Nand operation of A and B. 1 signal period of B is equal to 2 signal periods of A. Looking at the waveform, we draw the following table:

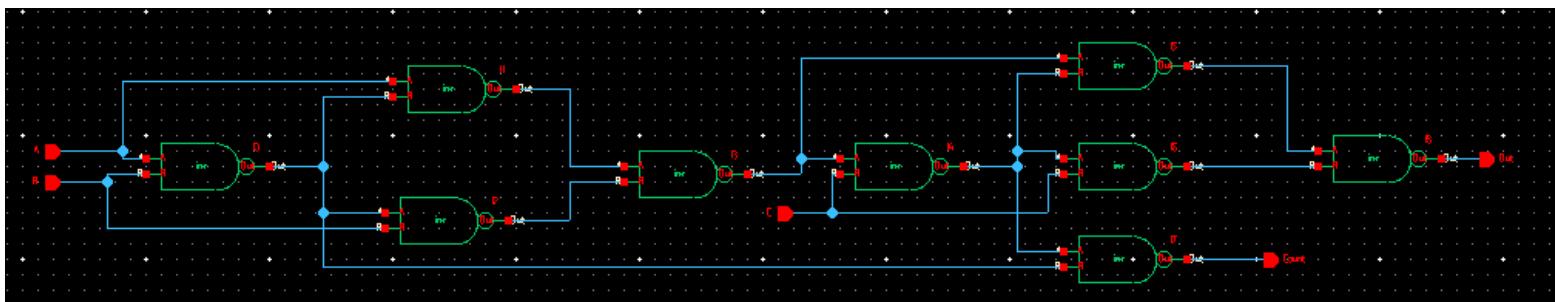
A	B	Nand2
1	1	0
0	1	1
1	0	1
0	0	1

=> The Nand2 we designed is running correctly

Section 3 : Schematic design and Full Adder 1bit simulation (Front-end):

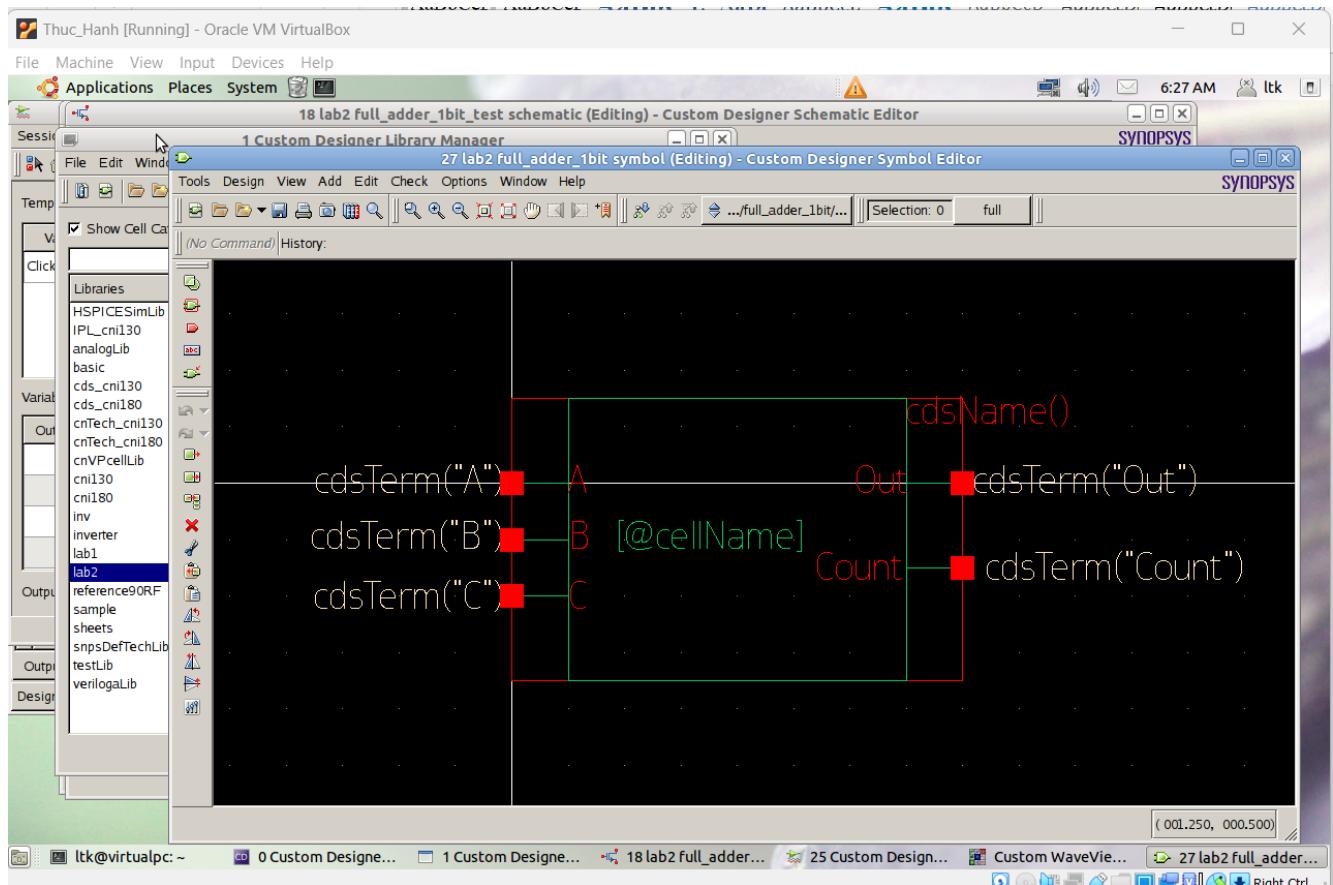
1. Schematic

_ Use nand2 we created above:

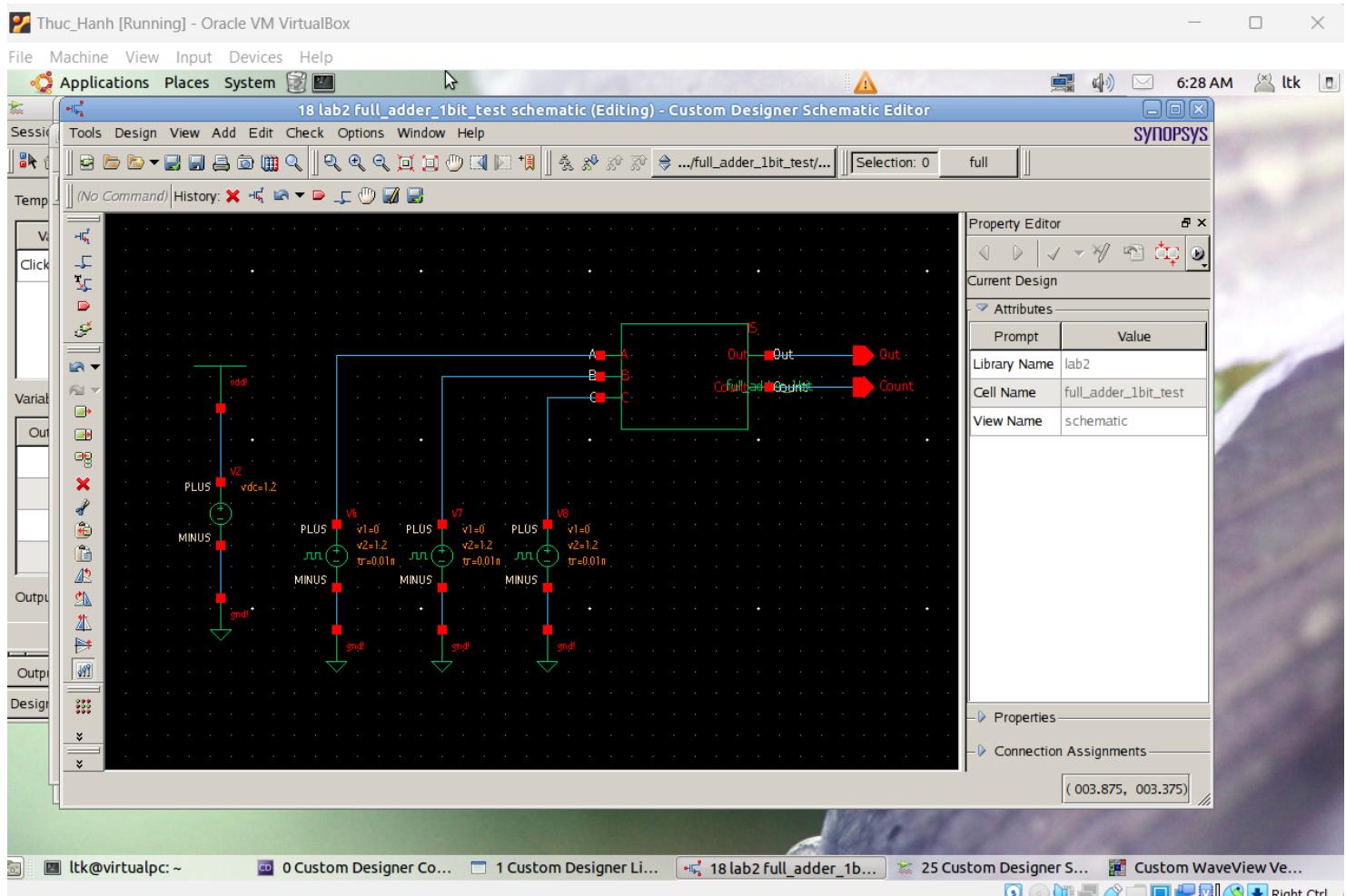


2 . Create Symbol and Testbench

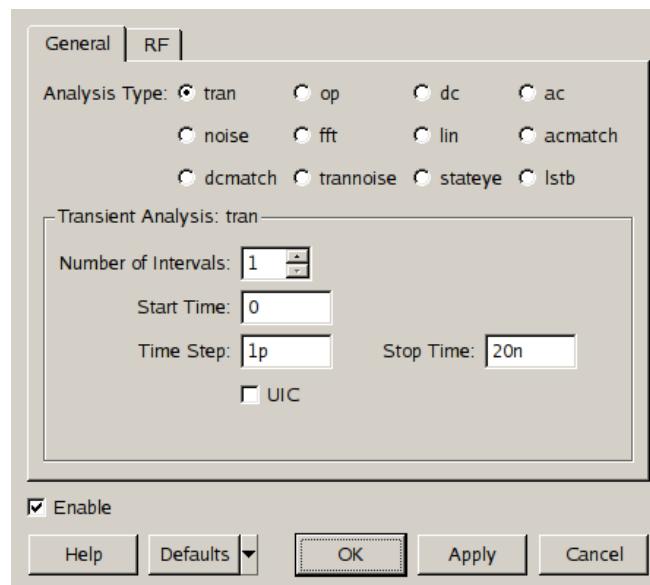
_ To facilitate simulation or reuse to design another circuit, people will create a symbol for the circuit. This symbol will represent the circuit in the design.



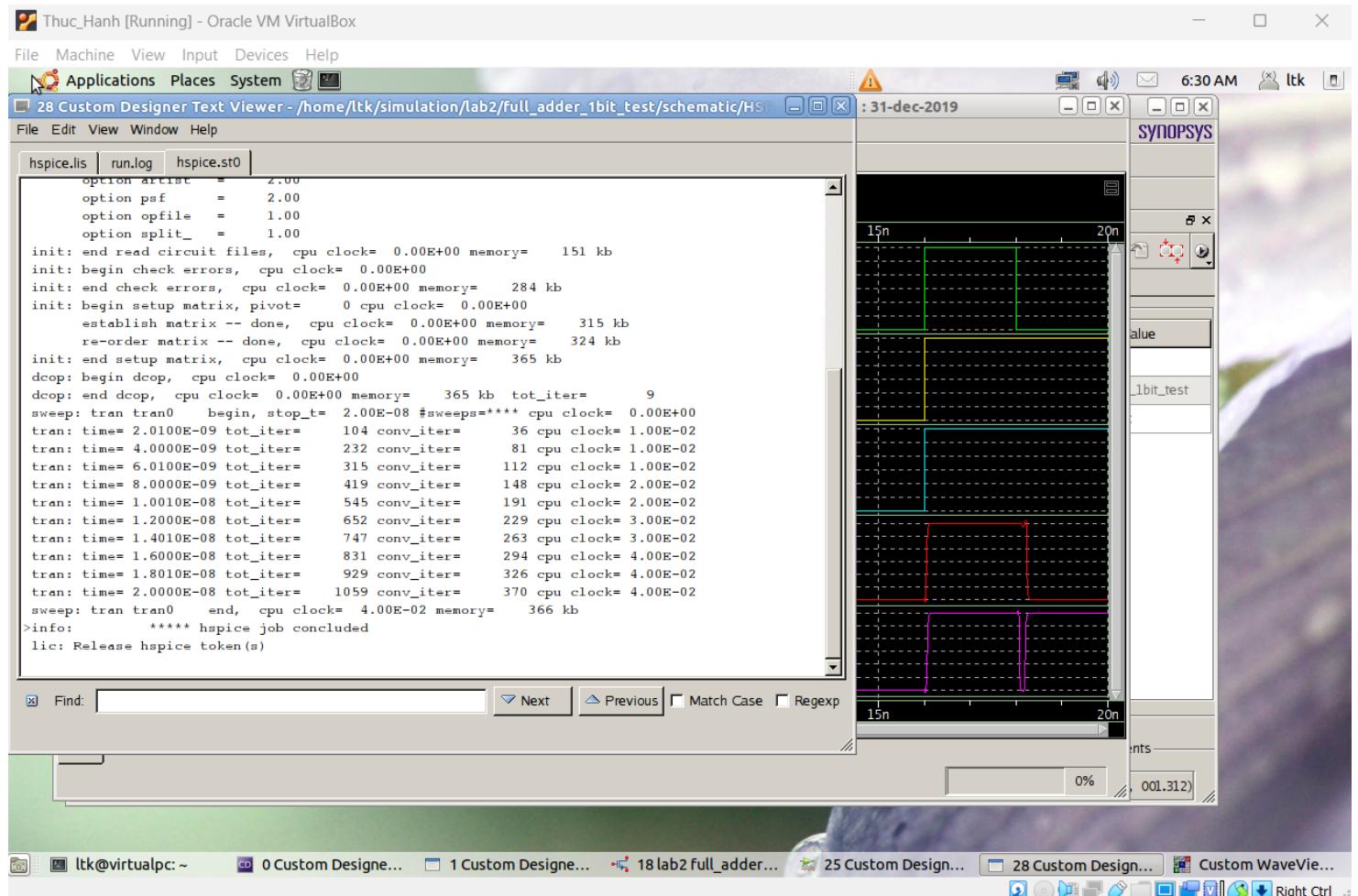
_ To create a testbench circuit for the Full Adder 1bit, we need to make another circuit to create an environment to test the operation of the circuit.

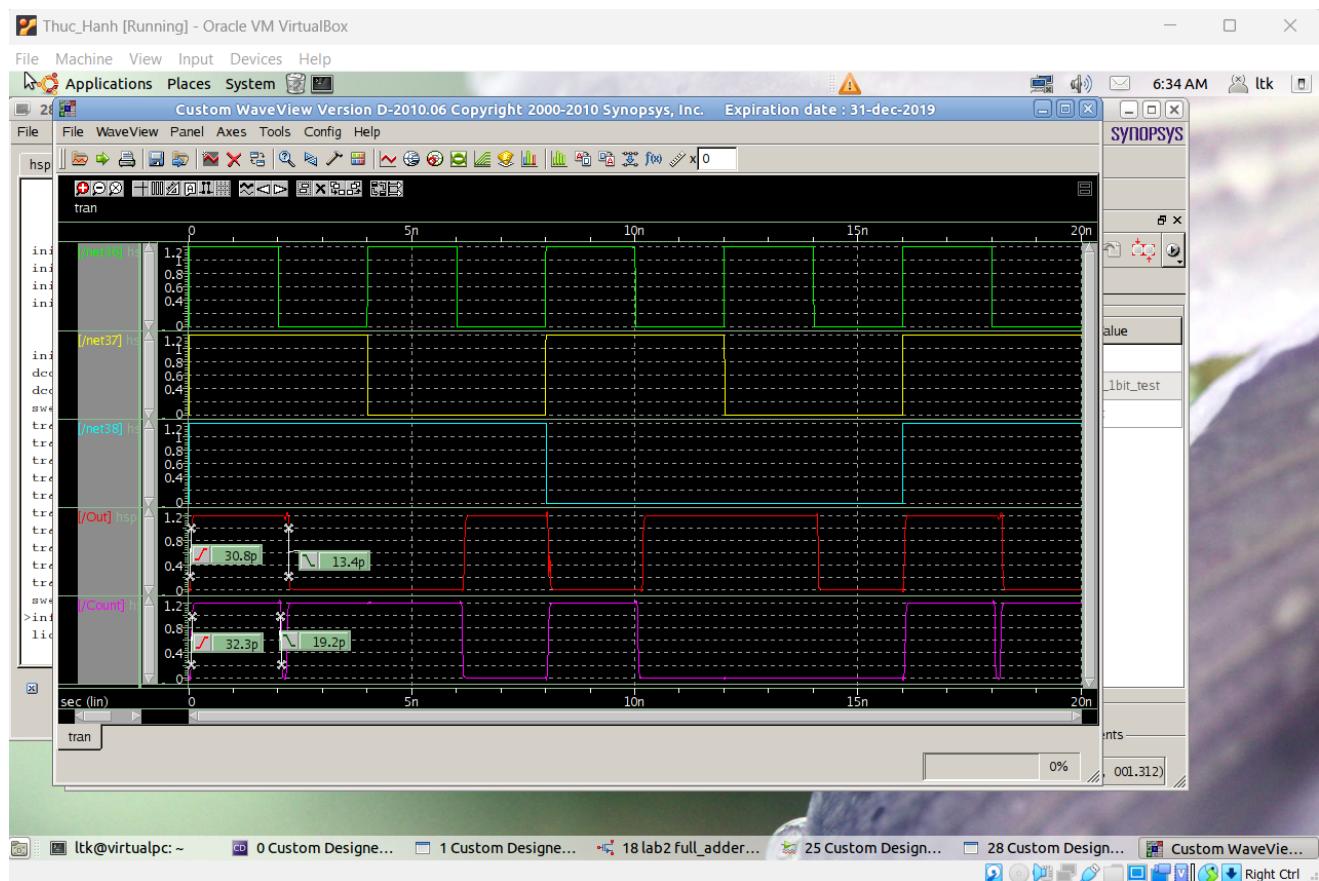


- Name the pin for the Full_adder_1bit as: output - In there:
 - + V2 is in the library analogLib => vdc. Choose vdc = 1.2v
 - + V6 is in the library analogLib => vpulse.
 - + V7 is in the library analogLib => vpulse.
 - + V8 is in the library analogLib => vpulse.
- => We choose V6 to have a Rise Time of 2ns and a Fall Time of 4ns, V7 to have a Rise Time of 4ns and a Fall Time of 6ns and finally V8 to have a Rise Time of 8ns and a Fall Time of 16ns to be able to simulate all cases of 1-bit full adder.
- => Set the delay to 0.001ns so that the delay time is small and does not affect the simulation
- _ Choose the simulation time:



_ A notification window appears, in the hspice.st0 window the following information displays the results simulation was successful (Result of simulation waveform by SAE):





=> In this waveform, we can see the circuit simulation results have shown the correct properties of the Full Adder 1bit between the relationship of input and output

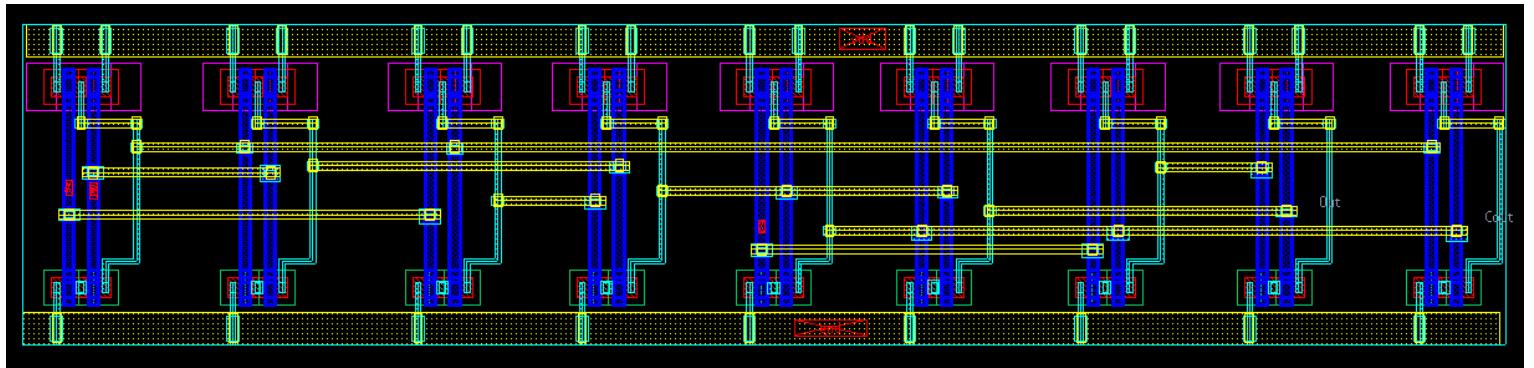
_ Explain: The green signal is A and the yellow signal is B and finally the light blue signal is C. The bottom 2 signals are the Sum and Cout values after performing the 1-bit Full Adder addition calculation between A, B and C. 1 signal cycle of B is equal to 2 signal cycles of A and the signal cycle of C is equal to 2 periods of B. Looking at the waveform, we can draw the following table:

A	B	C	Out	Cout
1	1	1	1	1
0	1	1	0	1
1	0	1	0	1
0	0	1	1	0
1	1	0	0	1
0	1	0	1	0
1	0	0	1	0
0	0	0	0	0

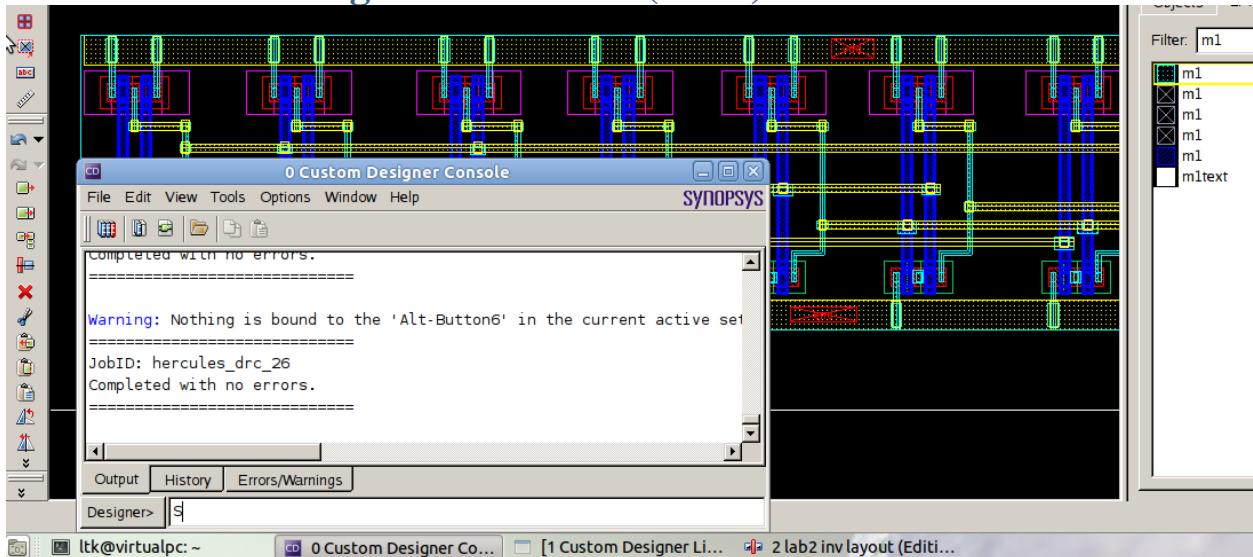
=> The Full Adder 1bit we designed is running correctly

Section 3 : Layout Full-Adder 1bit (Back-end)

1. Layout Full-Adder 1bit



2. Check Design Rules Check (DRC)



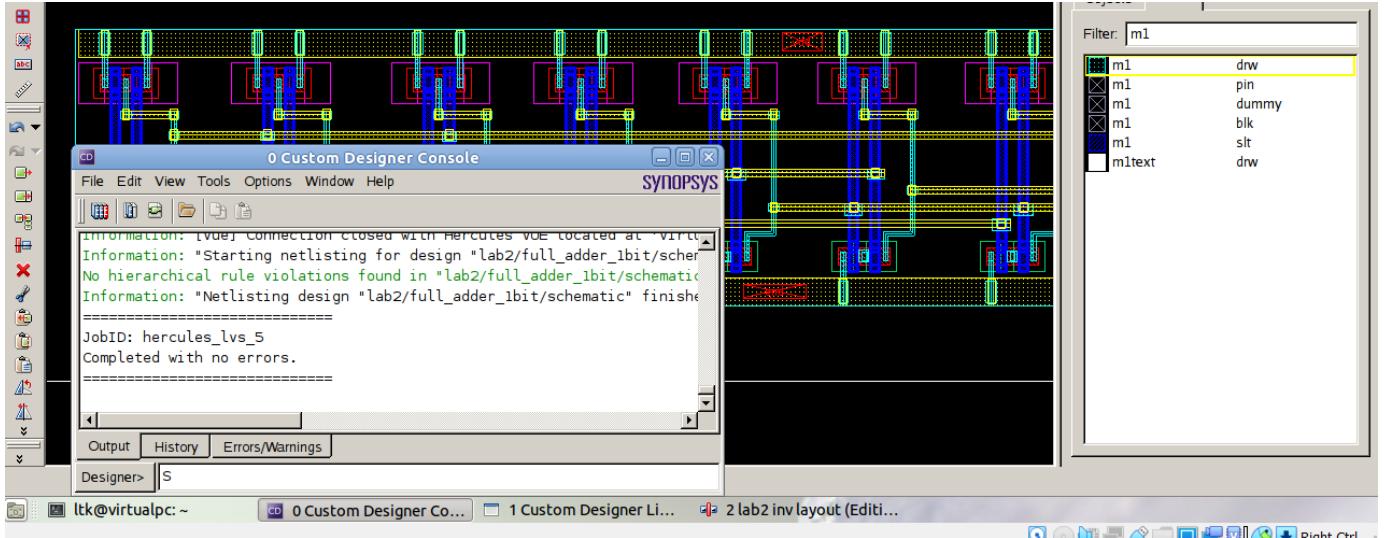
_ This step uses the DRC function to check whether the design complies with the design rules or not.

=> **There are no DRC errors.**

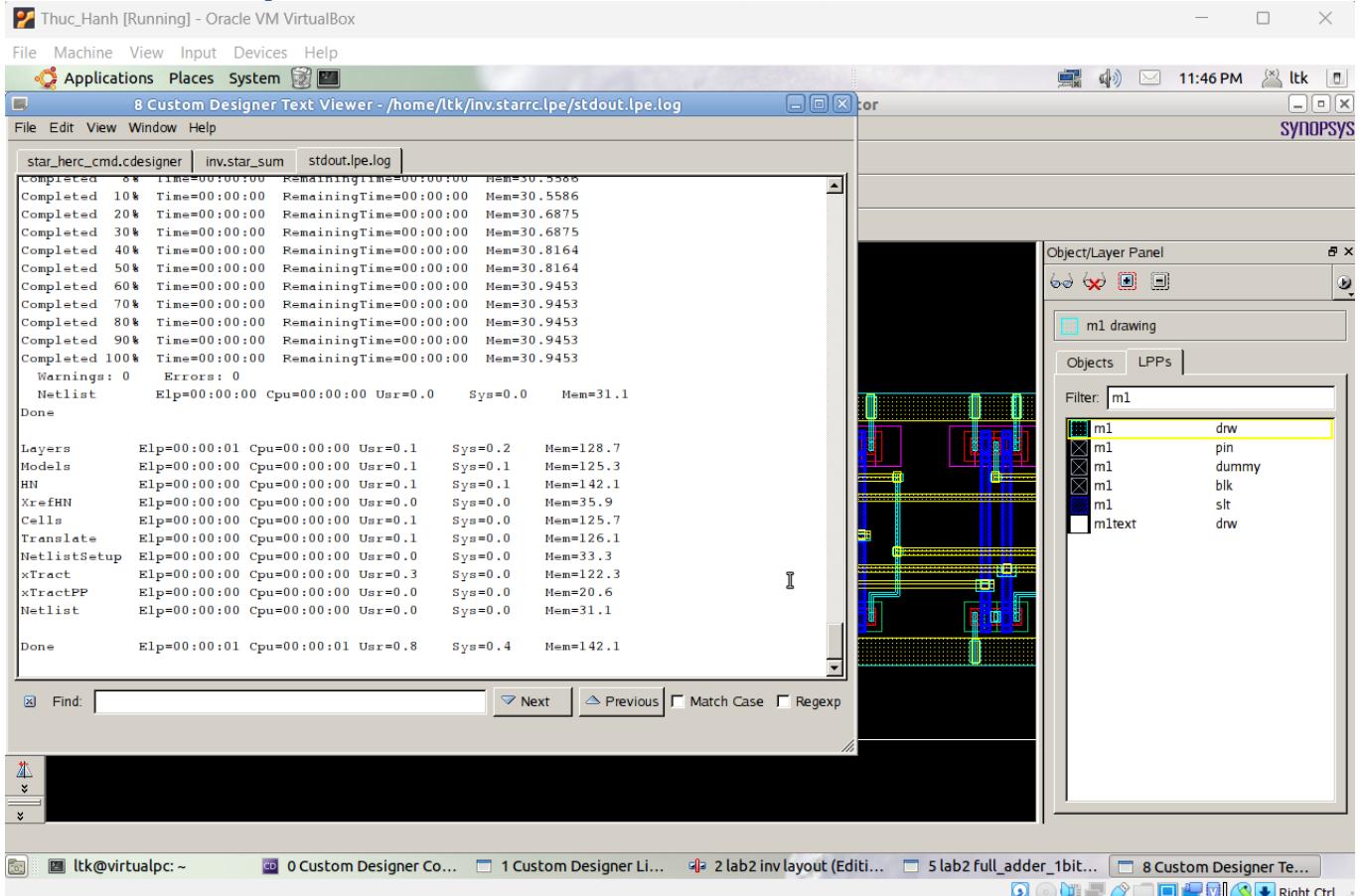
3. Check Layout Versus Schematic (LVS)

_ This step checks whether the layout is designed according to the design (schematic).

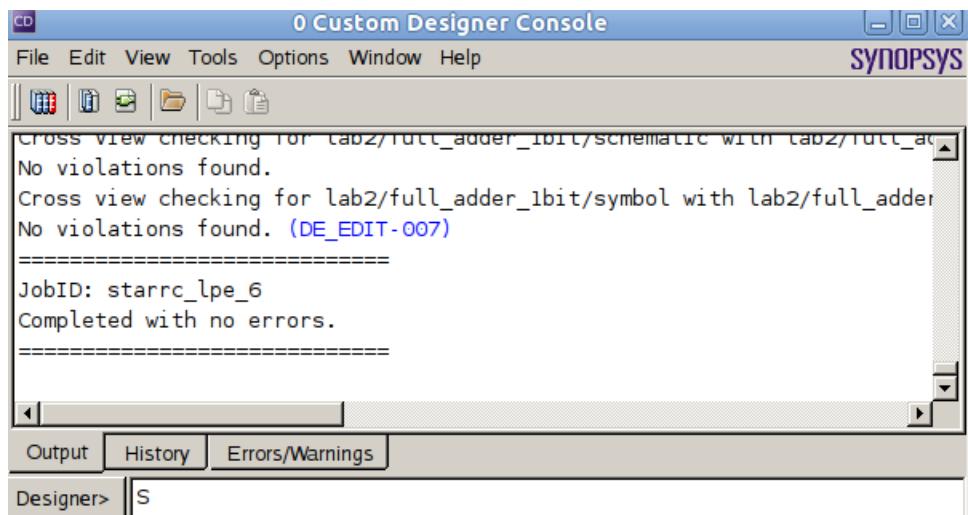
=> **The layout is designed according to the design (schematic) and does not appear errors.**



4. Layout Parasitic Extraction (LPE)



_ This step extracts the parasitic load on the circuit including capacitors and resistors.



=> There are no LPE errors. When checked LPE had no errors then the capacitors and resistor parasitics in the layout will be displayed in the fa1.sp file.

```

fa1 (~/L2) - gedit
File Edit View Search Tools Documents Help
Open Save Undo Redo Cut Copy Paste Find Replace
fa1 x
*|VENDOR "Synopsys"
*|PROGRAM "StarRC"
*|VERSION "D-2010.06"
*|DIVIDER |
*|DELIMITER :
*|OPERATING_TEMPERATURE 25
*|GLOBAL_TEMPERATURE 25
**FORMAT SPF
*
** COMMENTS
**
** TCAD_GRD_FILE /home/ltk/Test/PDK/starrc/reference_90nm_9lm_typ.nxtgrd
** TCAD_TIME_STAMP Mon Apr 6 20:59:52 2009
** TCADGRD_VERSION 64

.SUBCKT full_adder_1bit gnd! vdd! B A C Out Count
*|GROUND_NET 0
*LAYER_MAP
*0 SUBSTRATE
*1 1_0V_BJT_COLL
*2 2_5V_BJT_COLL
*3 1_0V_BJT_EMIT
*4 2_5V_BJT_EMIT
*5 1_0V_BJT_BASE
*6 2_5V_BJT_BASE

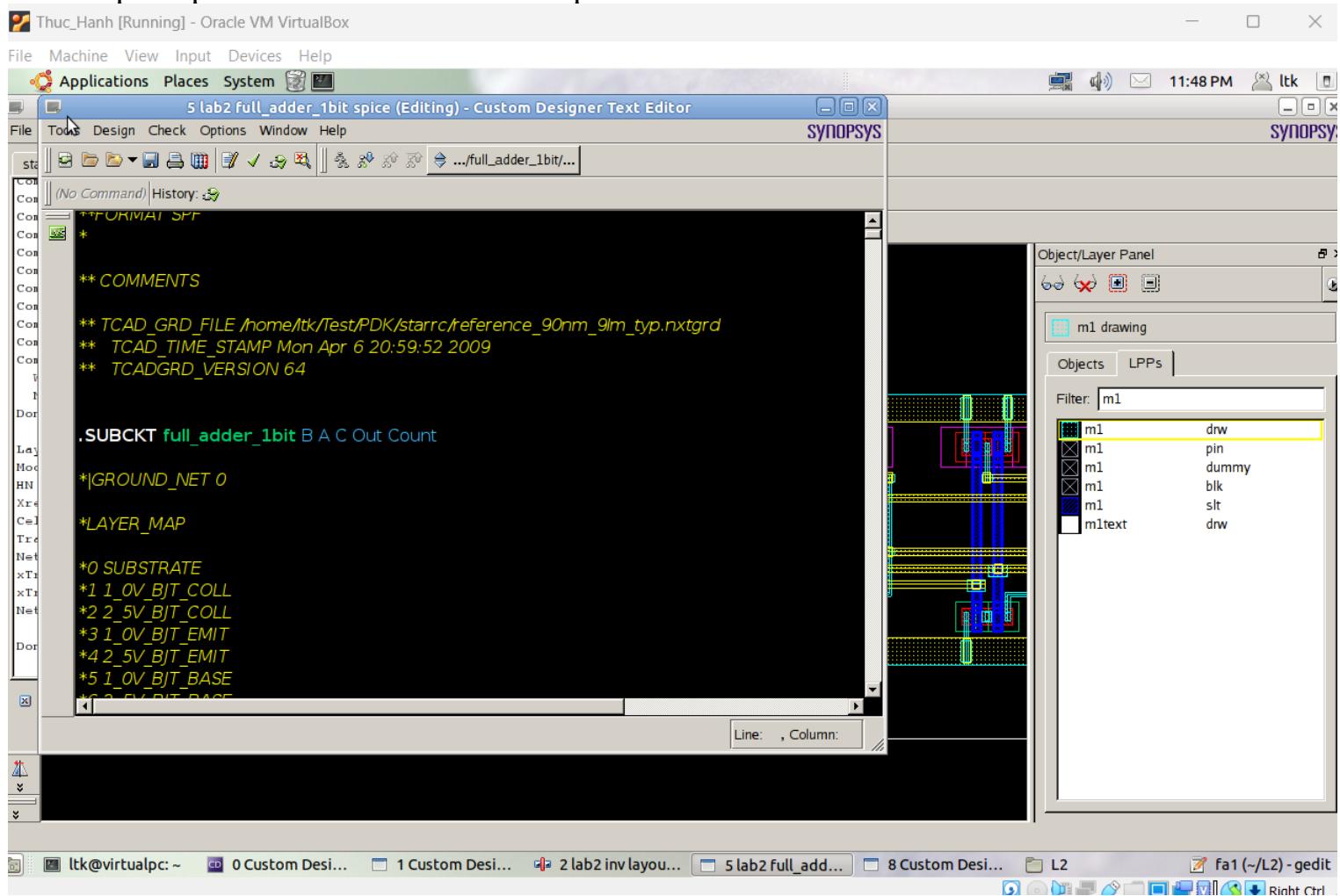
```

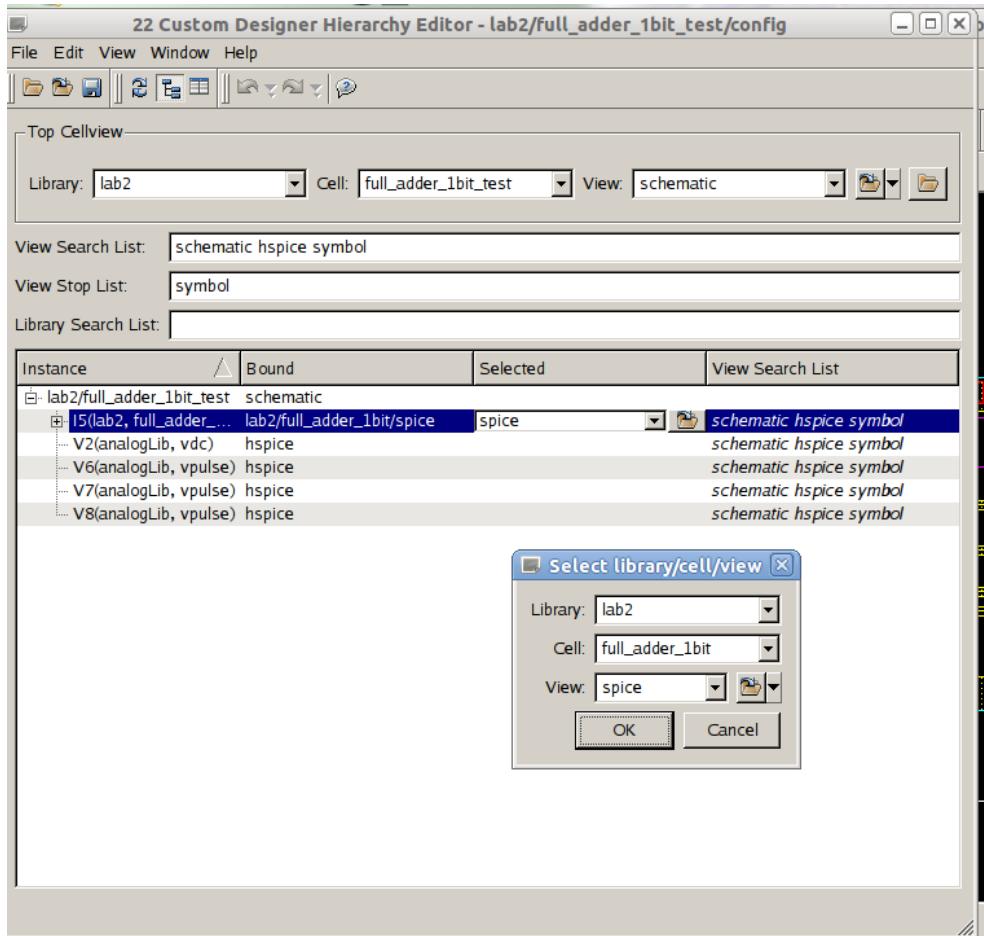
Plain Text Tab Width: 8 Ln 1, Col 1 INS

ltk@virtualpc: ~ 0 Custom Desi... 1 Custom Desi... 2 lab2 inv layout... 5 lab2 full_add... 8 Custom Desi...

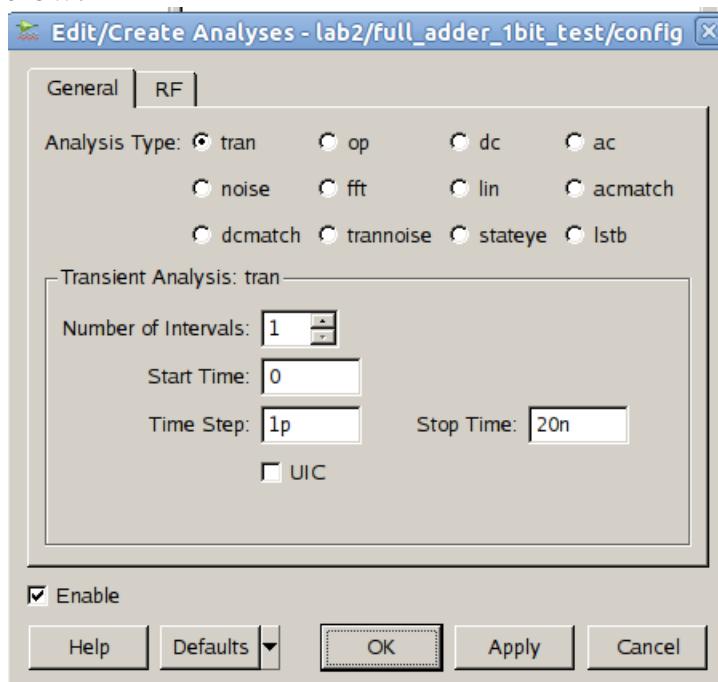
5. Simulate PostLayout and SEA simulation:

- Following a successful extraction of parasitic in the layout, we obtain the fa1.sp file, which is a hspice file used for PostLayout simulation.
- Conducting PostLayout simulation enables us to achieve more precise results by accounting for the effects of resistors and capacitors. Then we will generate Spice files within full_adder_1bit cells and substituting the contents of the spice.spc file with those of the fa1.sp file.

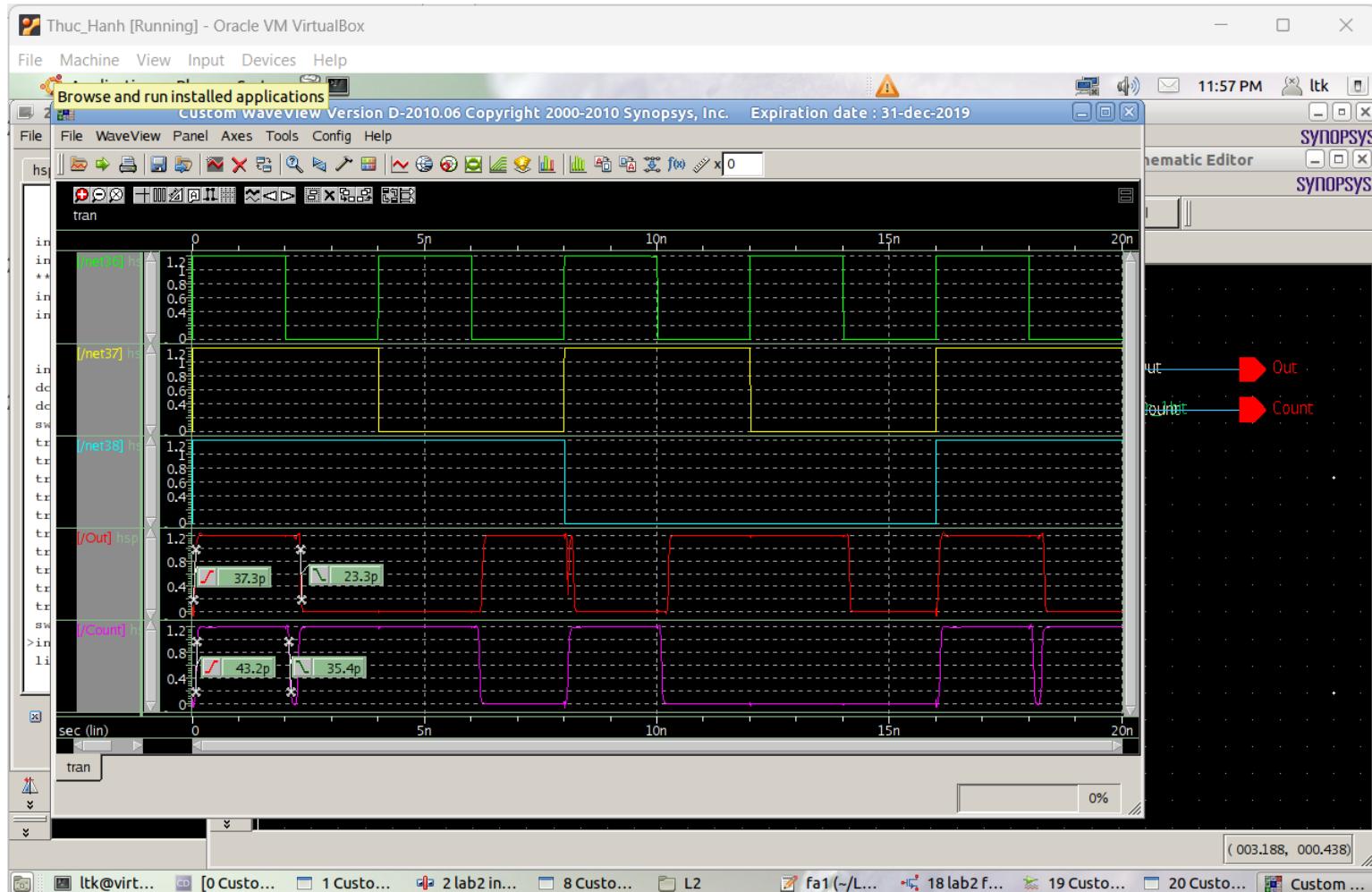




- Then we will simulate PostLayout using SAE with parameters and setup model file are below:



- To see the waveform of inverter circuit we plot the input and output of circuit, a program called WaveView displays simulation results as shown below:



=> In this waveform, we can see the circuit simulation results have shown the correct properties of the Full-Adder 1bit between the relationship of input and output

_ Explain: The green signal is A and the yellow signal is B and finally the light blue signal is C. The bottom 2 signals are the Sum and Cout values after performing the 1-bit Full Adder addition calculation between A, B and C. 1 signal cycle of B is equal to 2 signal cycles of A and the signal cycle of C is equal to 2 periods of A. Looking at the waveform, we can draw the following table:

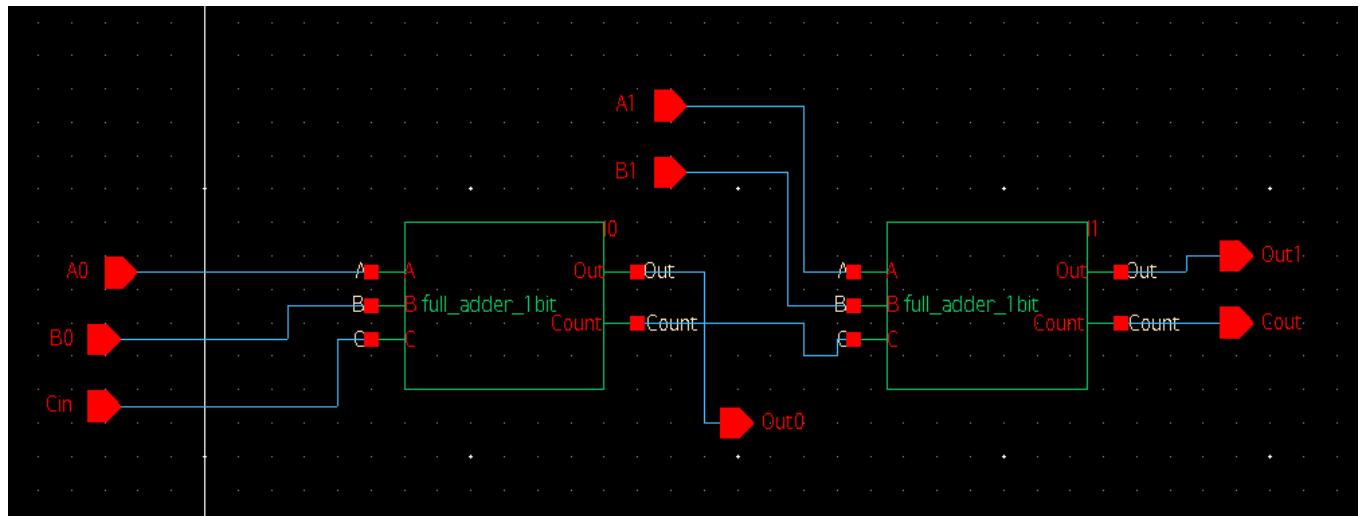
A	B	C	Out	Cout
1	1	1	1	1
0	1	1	0	1
1	0	1	0	1
0	0	1	1	0
1	1	0	0	1
0	1	0	1	0
1	0	0	1	0
0	0	0	0	0

=> The Full Adder 1bit we designed is running correctly

Section 4 : Schematic design and Full Adder 2bit simulation (Front-end):

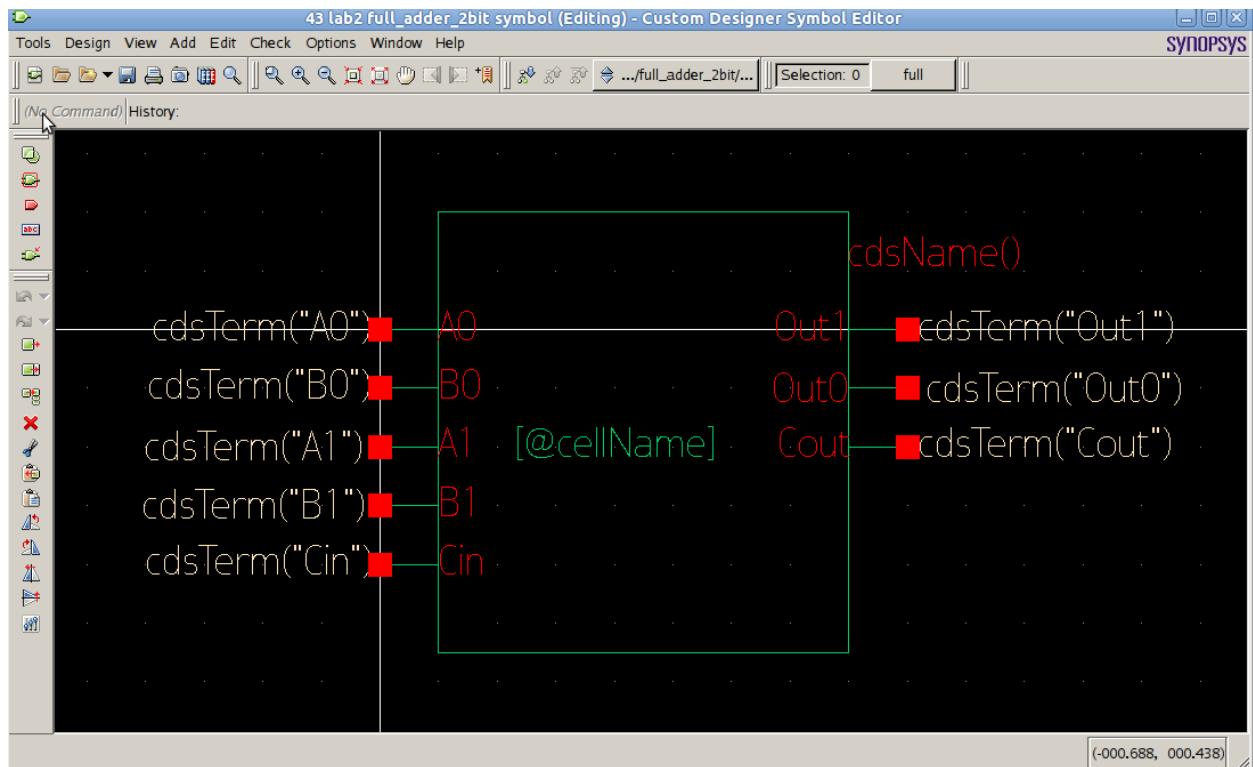
1. Schematic

Use Full Adder 1bit we created above:

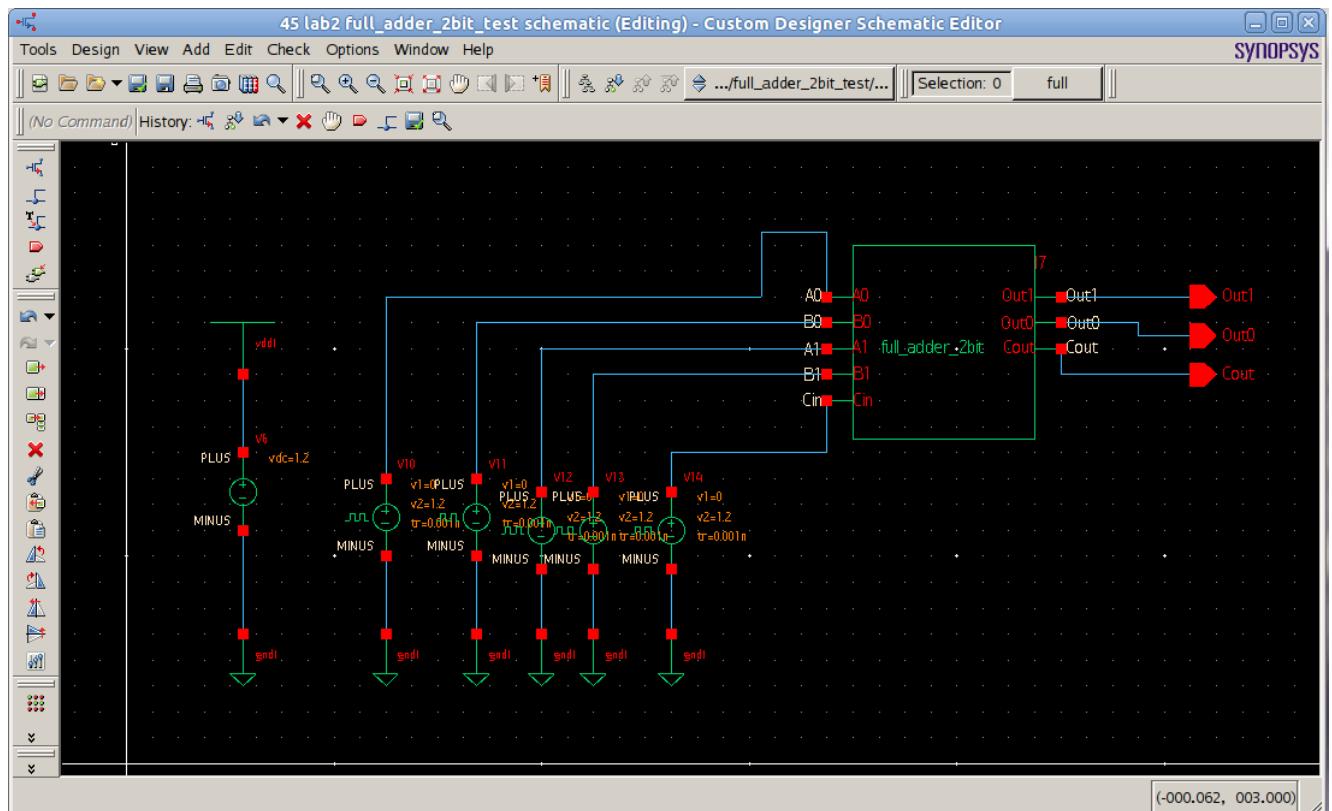


2 . Create Symbol and Testbench

_ To facilitate simulation or reuse to design another circuit, people will create a symbol for the circuit. This symbol will represent the circuit in the design.



To create a testbench circuit for the Full Adder 1bit, we need to make another circuit to create an environment to test the operation of the circuit.

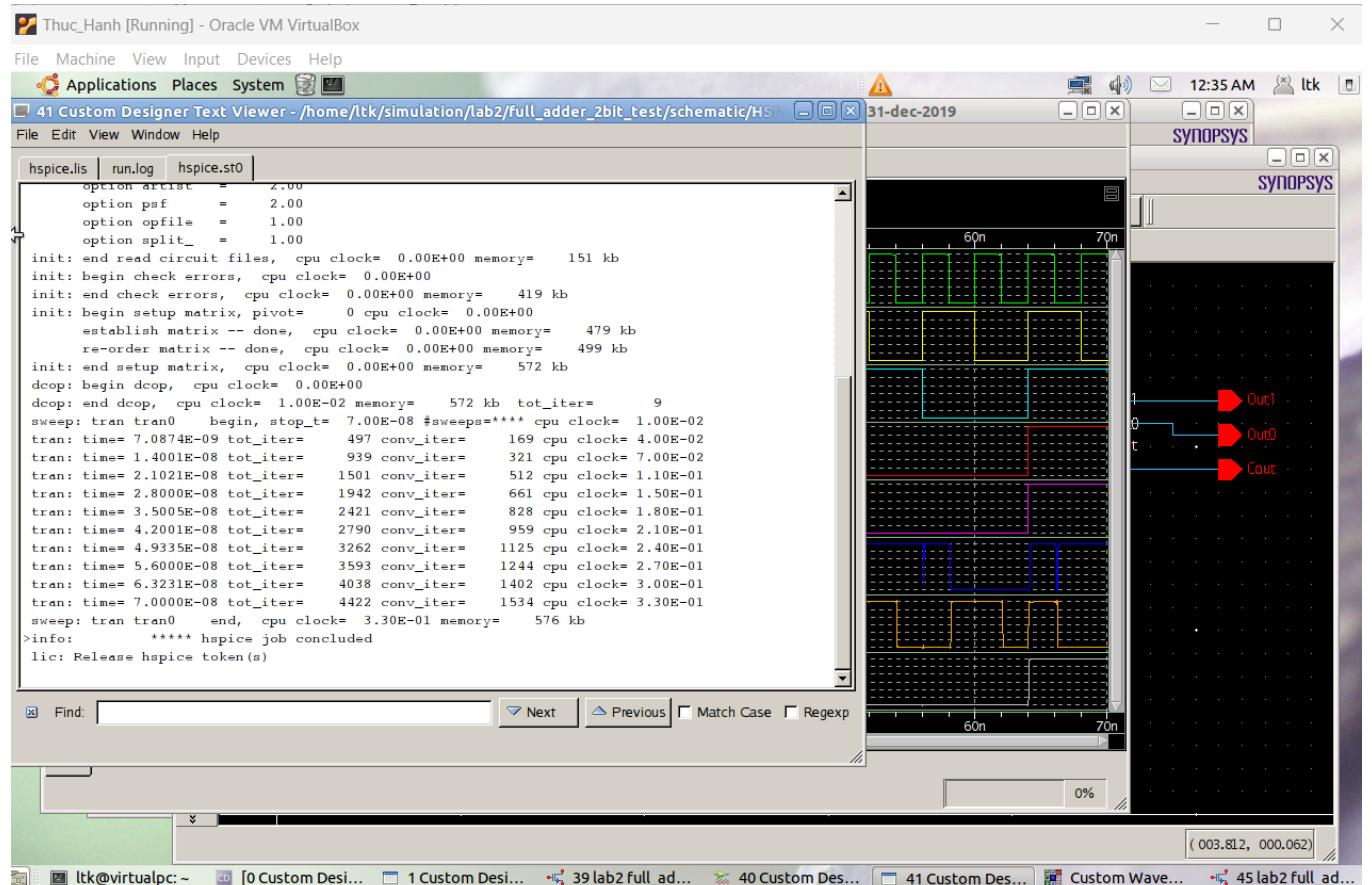


- Name the pin for the Full_adder_2bit as: output - In there:

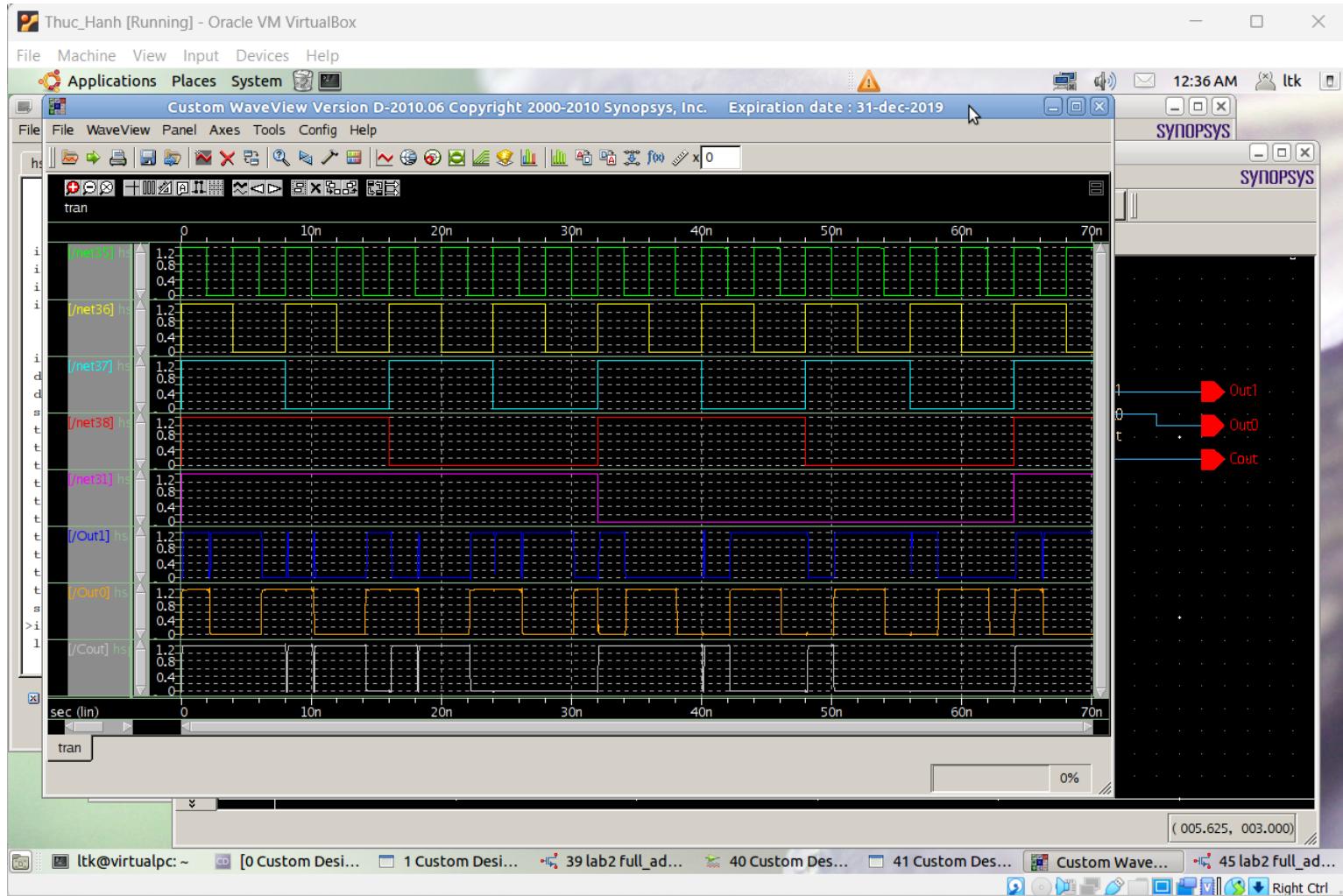
- + V6 is in the library analogLib => vdc. Choose vdc = 1.2v
- + V10 is in the library analogLib => vpulse.
- + V11 is in the library analogLib => vpulse.
- + V12 is in the library analogLib => vpulse.
- + V13 is in the library analogLib => vpulse.
- + V14 is in the library analogLib => vpulse.

=> We choose V10 with Rise Time of 2ns and Fall Time of 4ns, V11 with Rise Time of 4ns and Fall Time of 6ns, V12 with Rise Time of 8ns and Fall Time of 16ns, V13 with Rise Time of 16ns and Decrease Time of 32ns, V14 has a Rise Time of 32ns and Decrease Time of 64ns to be able to simulate all 2-bit full adder cases.

=> Set the delay to 0.001ns so that the delay time is small and does not affect the simulation



=> A notification window appears, in the hspice.st0 window the following information displays the results simulation was successful (Result of simulation waveform by SAE)



=> In this waveform, we can see the circuit simulation results have shown the correct properties of the Full-Adder 2bit between the relationship of input and output

_ Explain: The green signal is A0, the yellow signal is B0, the light blue signal is A1, the red signal is B1 and finally the pink signal is C. 1 signal cycle of B0 is equal to 2 signal cycles of A0, the cycle The signal cycle of A1 is equal to 2 cycles of B0, 1 cycle of B1 is equal to 2 cycles of A1 and 1 cycle of C is equal to 2 cycles of B1. Looking at the waveform, we can draw the following table:

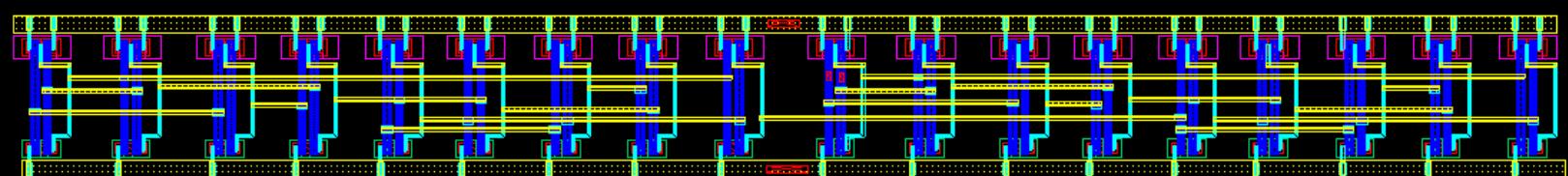
A0	B0	A1	B1	C	Out0	Out1	Cout
1	1	1	1	1	1	1	1
0	1	1	1	1	0	1	1
1	0	1	1	1	1	0	1
0	0	1	1	1	0	0	1
1	1	0	1	1	0	1	1
0	1	0	1	1	1	1	0
1	0	0	1	1	0	0	1

0	0	0	1	1	1	0	0
1	1	1	0	1	1	0	1
0	1	1	0	1	0	0	1
1	0	1	0	1	0	1	1
0	0	1	0	1	1	1	0
1	1	0	0	1	0	0	1
0	1	0	0	1	0	1	0
1	0	0	0	1	1	1	0
0	0	0	0	1	0	1	0
1	1	1	1	0	1	0	1
0	1	1	1	0	0	0	1
1	0	1	1	0	0	1	1
0	0	1	1	0	1	1	0
1	1	0	1	0	0	0	1
0	1	0	1	0	1	0	0
1	0	0	1	0	1	1	0
0	0	0	1	0	0	0	1
1	1	1	0	0	0	1	1
0	1	1	0	0	1	1	0
1	0	1	0	0	0	0	1
0	0	1	0	0	1	0	0
1	1	0	0	0	1	1	0
0	1	0	0	0	0	1	0
1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0

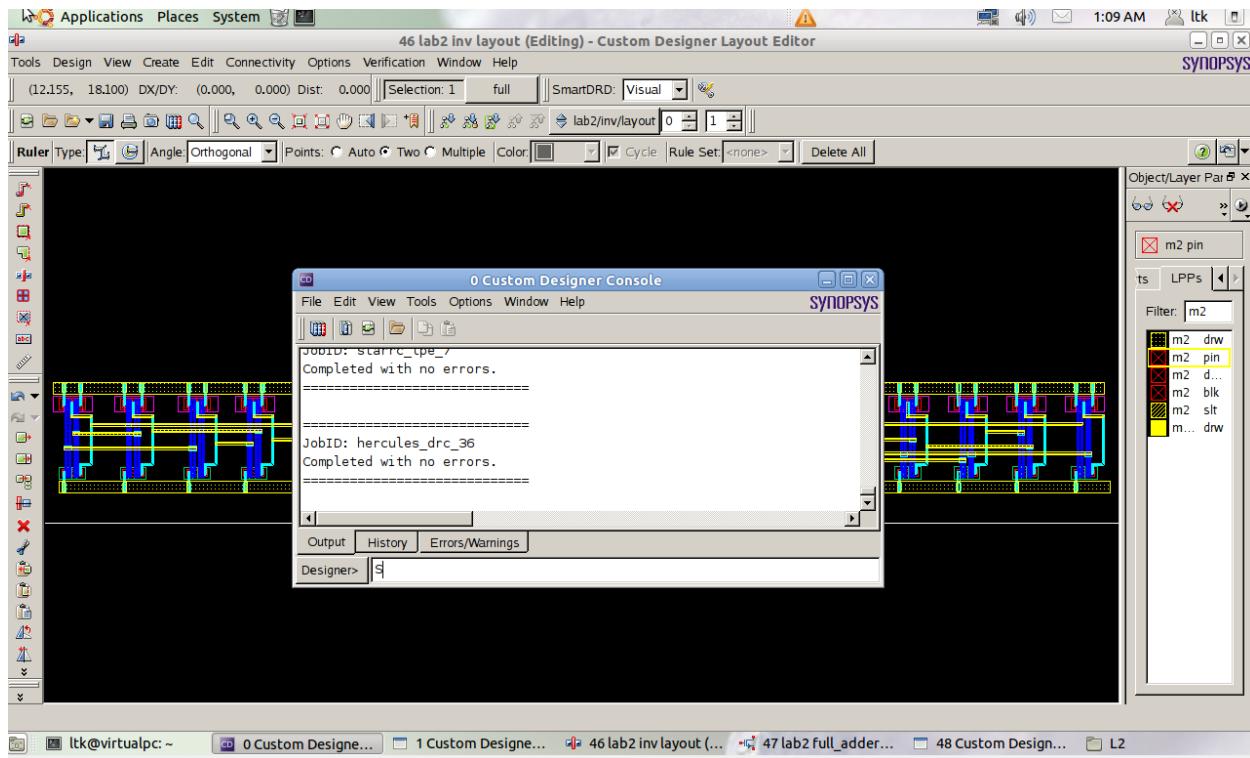
=> The Full Adder 2bit we designed is running correctly

Section 5 : Layout Full-Adder 2bit (Back-end)

1. Layout Full-Adder 2bit



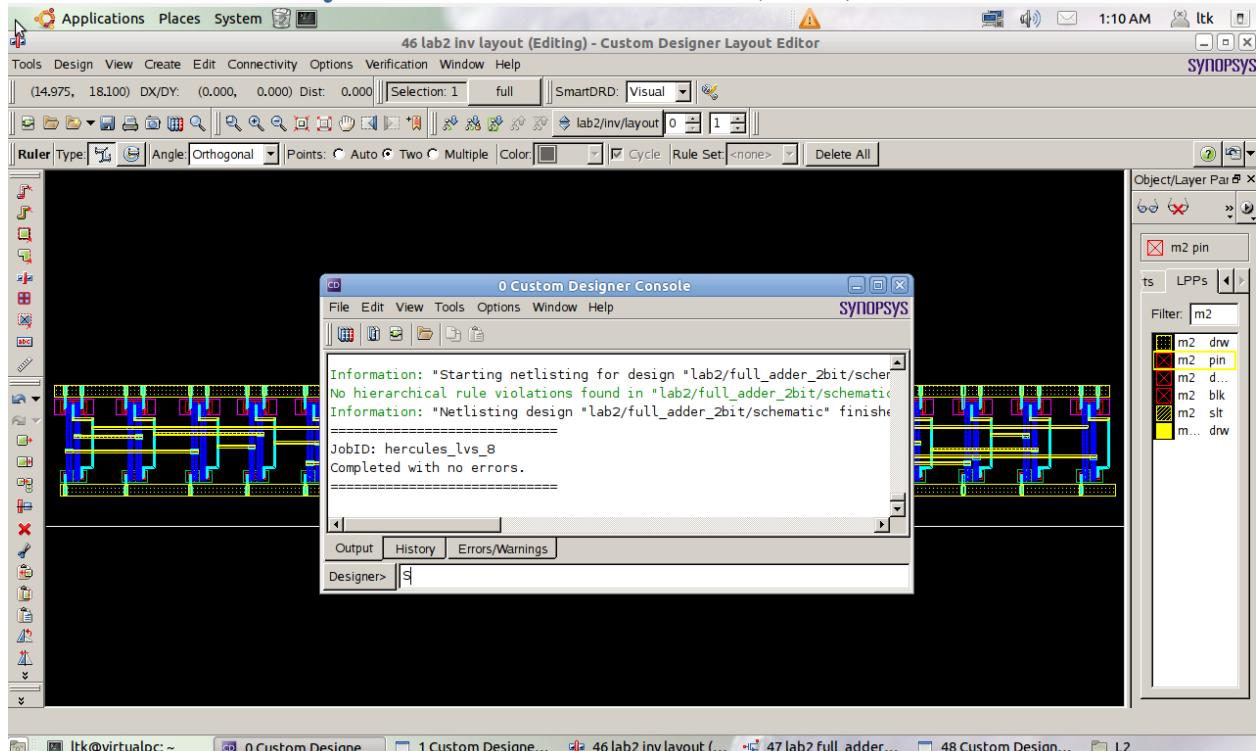
2. Check Design Rules Check (DRC)



This step uses the DRC function to check whether the design complies with the design rules or not.

=> There are no DRC errors.

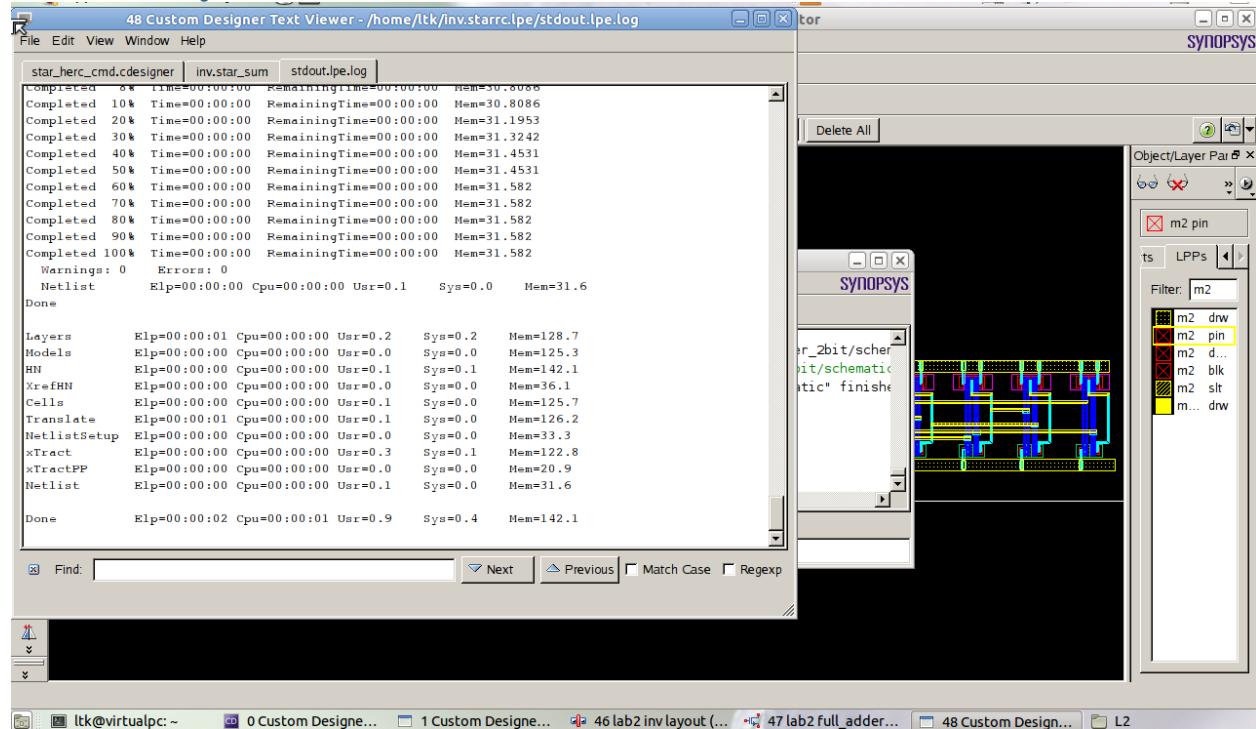
3. Check Layout Versus Schematic (LVS)



_ This step checks whether the layout is designed according to the design (schematic).

=> The layout is designed according to the design (schematic) and does not appear errors.

4. Layout Parasitic Extraction (LPE)



=> There are no LPE errors. When checked LPE had no errors then the capacitors and resistor parasitics in the layout will be displayed in the fa2.sp file.

fa2 (~/L2) - gedit

File Edit View Search Tools Documents Help

Open Save Undo Redo Cut Copy Paste Find Replace

fa2

```
*|VENDOR "Synopsys"
*|PROGRAM "StarRC"
*|VERSION "D-2010.06"
*|DIVIDER |
*|DELIMITER :
*|OPERATING_TEMPERATURE 25
*|GLOBAL_TEMPERATURE 25
**FORMAT_SPF
*
** COMMENTS

** TCAD_GRD_FILE /home/ltk/Test/PDK/starrc/reference_90nm_9lm_typ.nxtgrd
**   TCAD_TIME_STAMP Mon Apr 6 20:59:52 2009
**   TCADGRD_VERSION 64

.SUBCKT full_adder_2bit gnd! vdd! B0 A0 Cin Out0 B1 A1 Out1 Cout
*|GROUND_NET 0
*LAYER_MAP

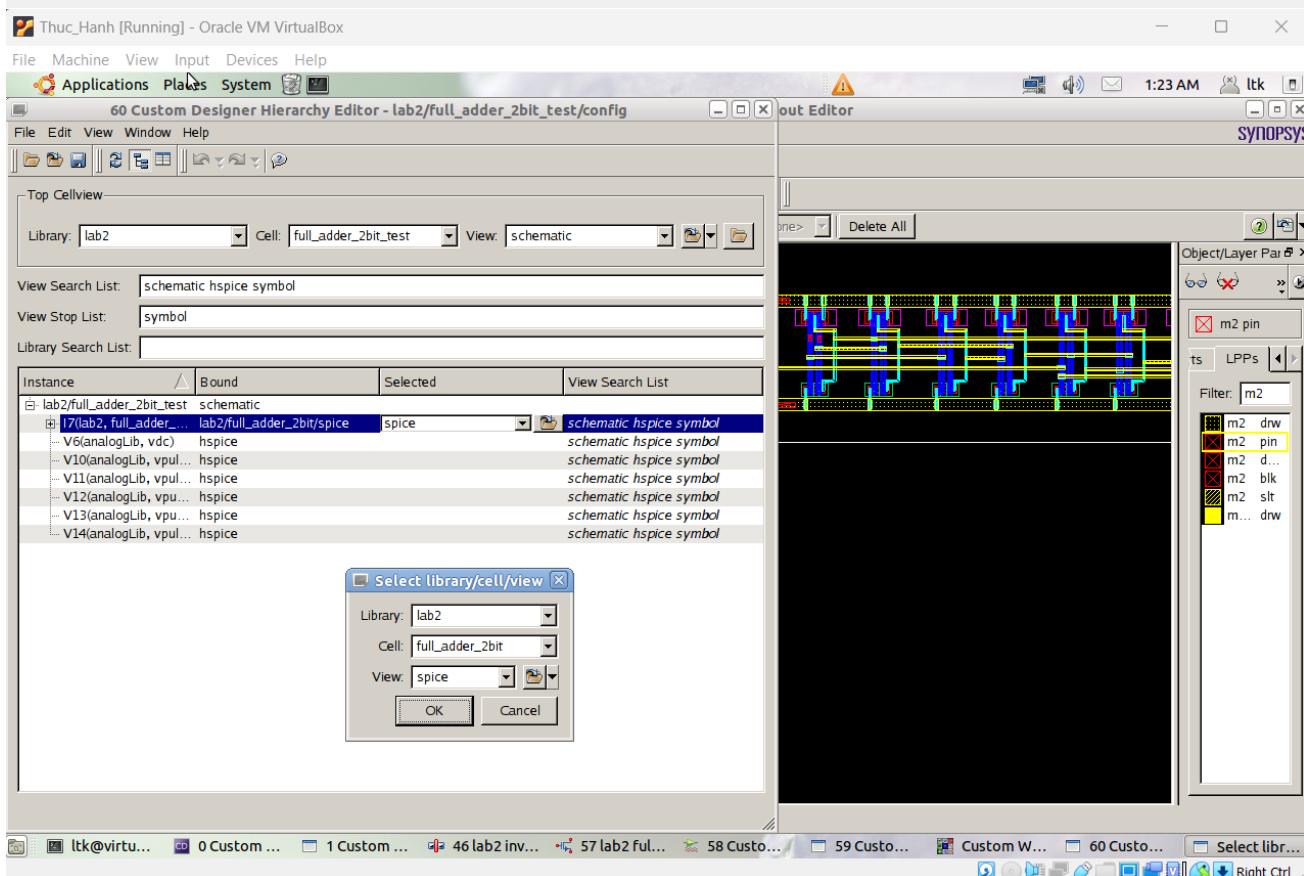
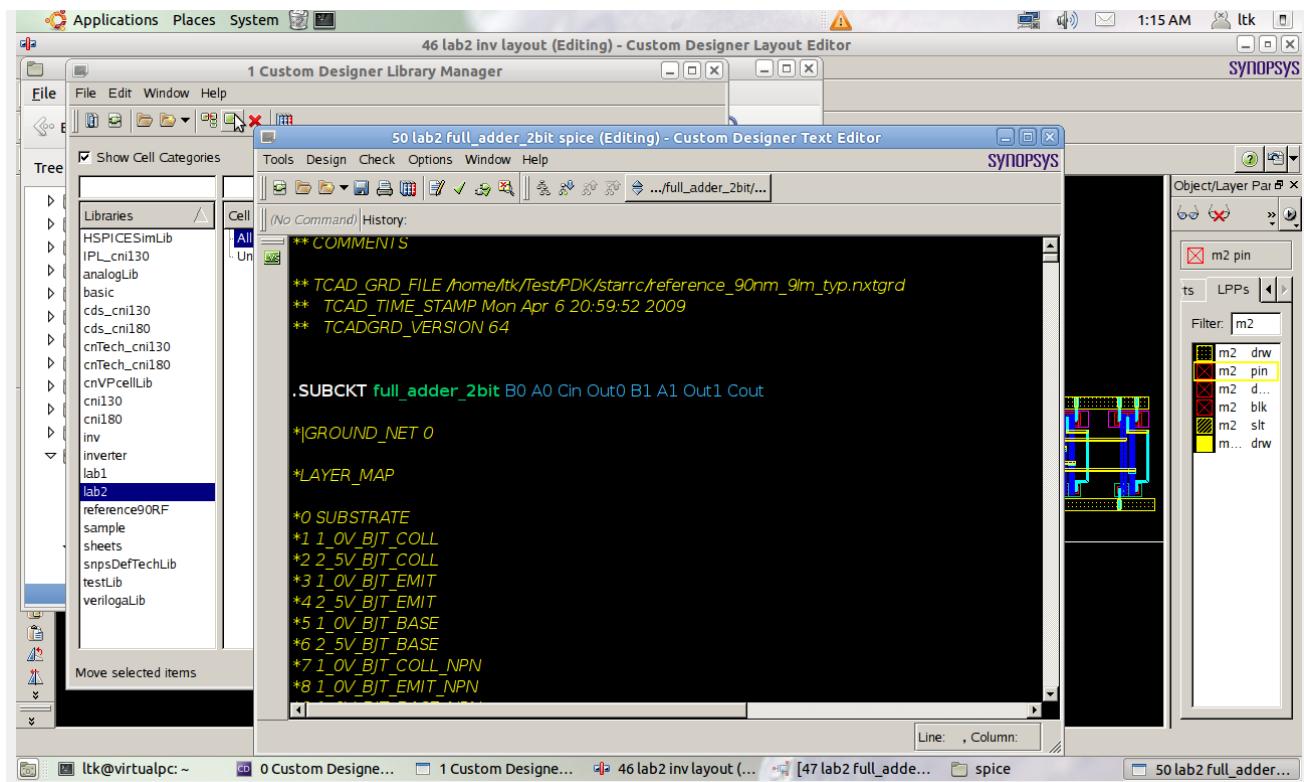
*0 SUBSTRATE
*1 1_0V_BJT_COLL
*2 2_5V_BJT_COLL
*3 1_0V_BJT_EMIT
*4 2_5V_BJT_EMIT
*5 1_0V_BJT_BASE
*6 2_5V_BJT_BASE
```

Plain Text Tab Width: 8 Ln 1, Col 1 INS

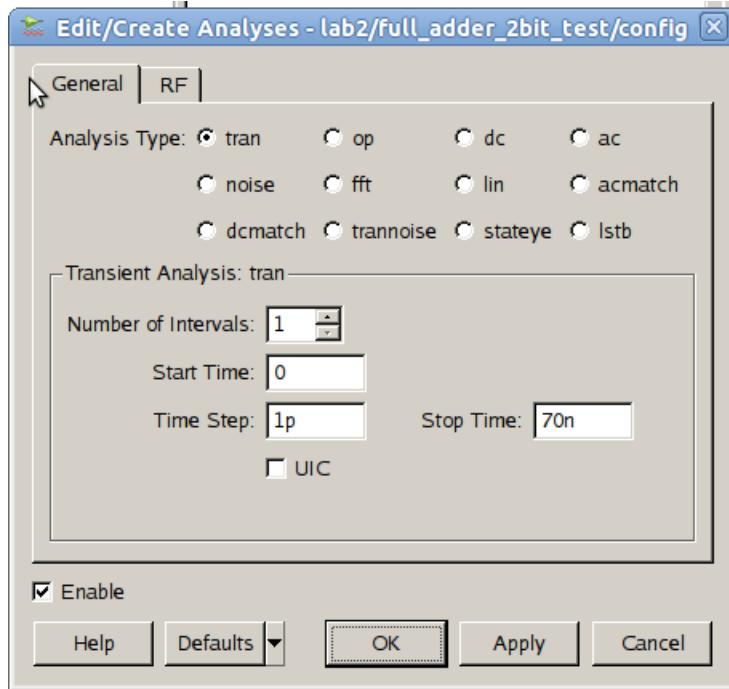
Custom Des... 1 Custom Des... 46 lab2 inv layo... 47 lab2 full_ad... 48 Custom Des... L2

5. Simulate PostLayout and SEA simulation:

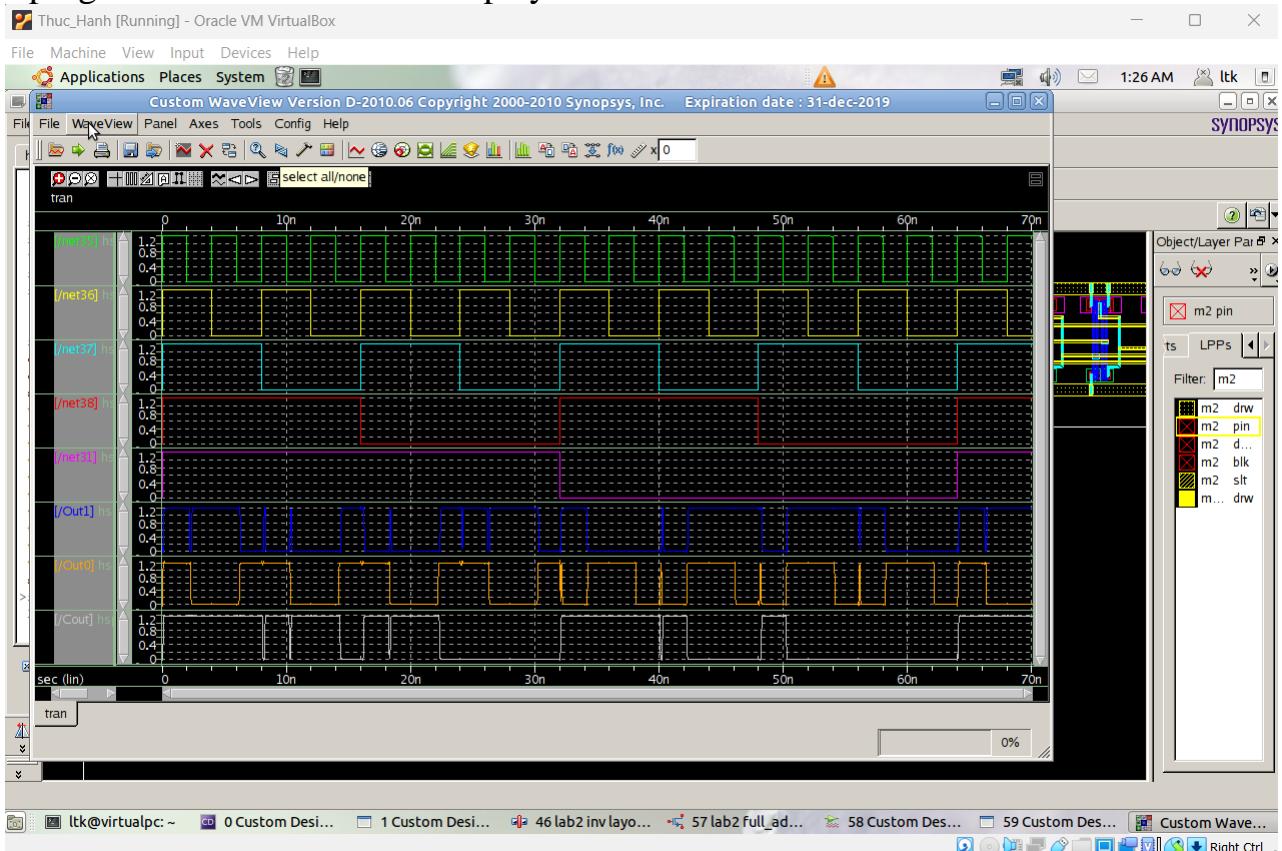
- Following a successful extraction of parasitic in the layout, we obtain the Fa2.sp file, which is a hspice file used for PostLayout simulation. Conducting PostLayout simulation enables us to achieve more precise results by accounting for the effects of resistors and capacitors. Then we will generate Spice files within full adder 2bit cells and substituting the contents of the spice.spc file with those of the fa2.sp file.



- Then we will simulate PostLayout using SAE with parameters and setup model file are below:



- To see the waveform of inverter circuit we plot the input and output of circuit, a program called WaveView displays simulation results as shown below:



=> In this waveform, we can see the circuit simulation results have shown the correct properties of the Full-Adder 2bit between the relationship of input and output

_ Explain: The green signal is A0, the yellow signal is B0, the light blue signal is A1, the red signal is B1 and finally the pink signal is C. 1 signal cycle of B0 is equal to 2 signal cycles of A0, the cycle The signal cycle of A1 is equal to 2 cycles of B0, 1 cycle of B1 is equal to 2 cycles of A1 and 1 cycle of C is equal to 2 cycles of B1. Looking at the waveform, we can draw the following table:

A0	B0	A1	B1	C	Out0	Out1	Cout
1	1	1	1	1	1	1	1
0	1	1	1	1	0	1	1
1	0	1	1	1	1	0	1
0	0	1	1	1	0	0	1
1	1	0	1	1	0	1	1
0	1	0	1	1	1	1	0
1	0	0	1	1	0	0	1
0	0	0	1	1	1	0	0
1	1	1	0	1	1	0	1
0	1	1	0	1	0	0	1
1	0	1	0	1	0	1	1
0	0	1	0	1	1	1	0
1	1	0	0	1	0	0	1
0	1	0	0	1	0	1	0
1	0	0	0	1	1	1	0
0	0	0	0	1	0	1	0
1	1	1	1	0	1	0	1
0	1	1	1	0	0	0	1
1	0	1	1	0	0	1	1
0	0	1	1	0	1	1	0
1	1	0	1	0	0	0	1
0	1	0	1	0	1	0	0
1	0	0	1	0	1	1	0
0	0	0	1	0	0	0	1
1	1	1	0	0	0	1	1
0	1	1	0	0	1	1	0
1	0	1	0	0	0	0	1
0	0	1	0	0	1	0	0
1	1	0	0	0	1	1	0
0	1	0	0	0	0	1	0

1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0

=> The Full Adder 2bit we designed is running correctly