|  |  |  |  |
| --- | --- | --- | --- |
| Lab3 | | | |
| Project name |  | | |
| Document ref |  | | |
| Version |  | | |
| Release date |  | | |
| Author | Wu Fei | | |
| Classification |  | | |
| Distribution List |  | | |
| Approved by | Name | Signature | Date |
|  |  |  |  |



**Ming Hsieh Department of Electrical Engineering**

**University of Southern California, Los Angeles, CA 90089**

**Spring 2022**

**EE 477**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | List of changes | Author + Signature |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

C A T A L O G

[1 Motivation 1](#_Toc98332715)

[2 MUX design 2](#_Toc98332716)

[2.1 1bit mux2 2](#_Toc98332717)

[2.2 2bits mux 5](#_Toc98332718)

[3 8\_bit carry select adder design 10](#_Toc98332719)

[3.1 CSA\_8bit 11](#_Toc98332720)

[3.1.1 SCH 11](#_Toc98332721)

[3.2 TEST 13](#_Toc98332722)

[3.3 LAYOUT 15](#_Toc98332723)

# Motivation

The delay of **RCAs** (Ripple Carry Adders) increases linearly as the number of inputs grows. One of the faster adders is the CSA (Carry Select Adder) which uses parallelism to increase speed with higher area costs. You will design (the layout) of an 8-bit CSA.

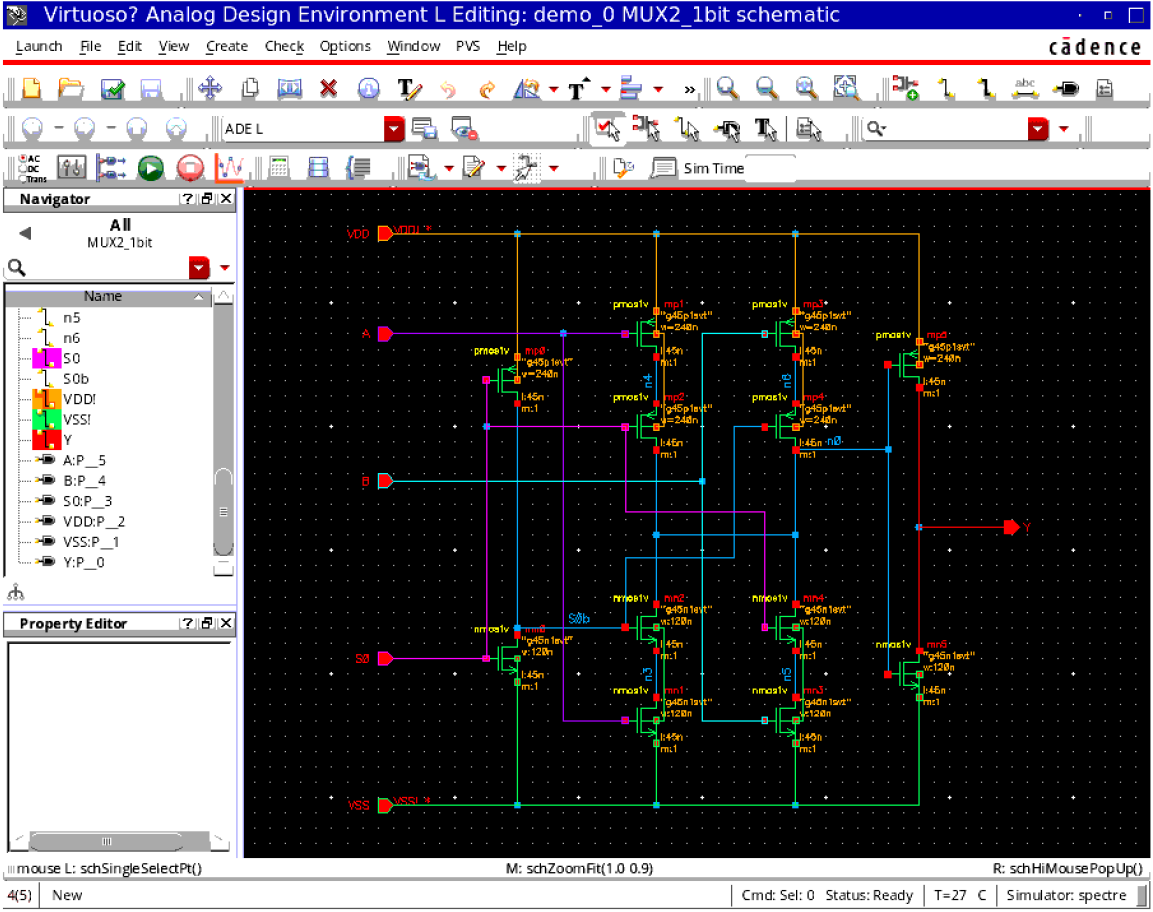
* Use **rise and fall time for all input stimuli = 10ps** (Note that it is NOT 100)
* For functionality waveforms, please write what the inputs and expected outputs are for each case. Example: Inputs are A=11111111 (255 in decimal), B=10000000 (128 in decimal), C0=0, expected outputs are C8=1, S=01111111. Then show the 9 output waveforms only.
* The input transition you chose for worst case delay of the CSA.
* Average delays for C8 for schematic and extracted.

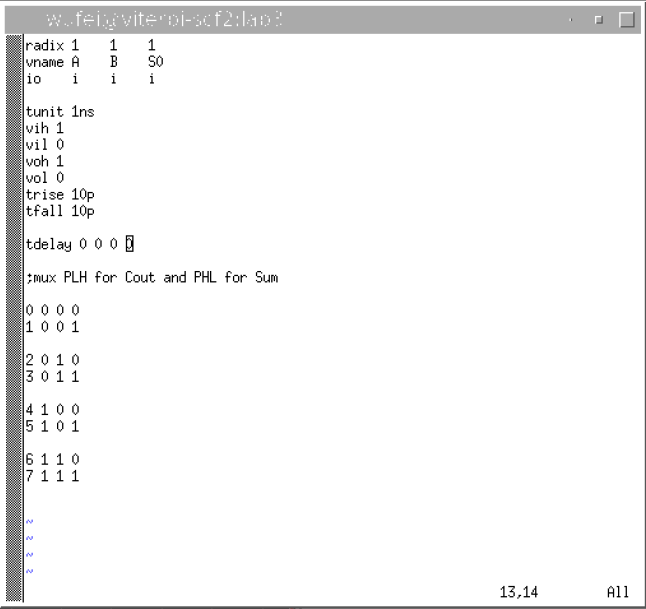
# MUX design

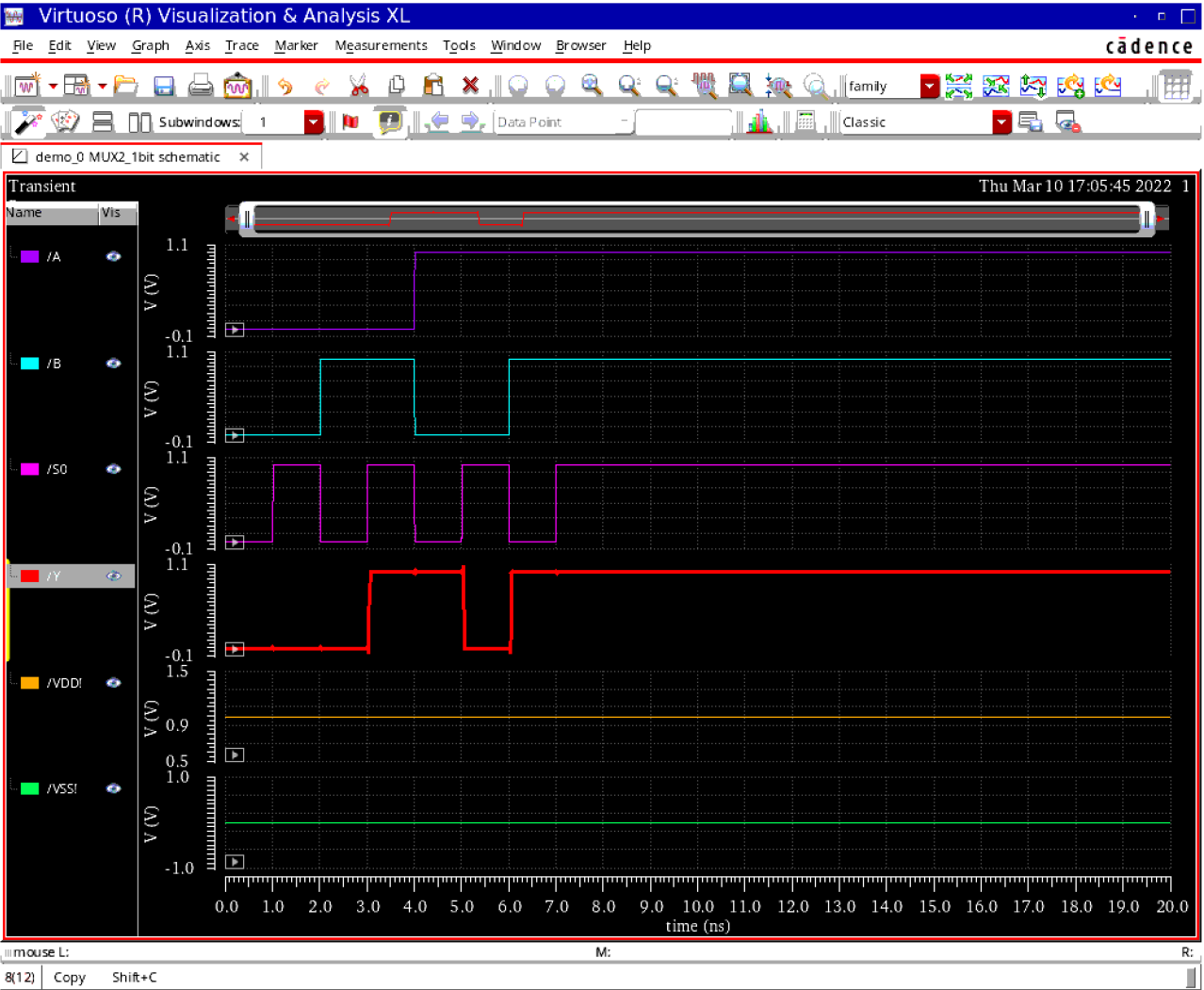
* An N-bit wide, M-to-1 multiplexer (MUX) has M inputs each of which is N-bits. It selects one of them to pass to output (which has N bits). Selection is done using log2M select bits which are inputs.
* ~~Design a 4-bit wide 2-to-1 MUX schematic. It will have 2 sets of inputs, each of which is 4 bits. It will also have a single select bit input and a 4-bit output.~~
* Design a 2-bit wide 2-to-1 MUX schematic. It will have 2 sets of inputs, each of which is 2 bits. It will also have a single select bit input and a 2-bit output. You only need 2 bit muxes in the CSA design.
* Draw the schematic and layout and verify the functionality

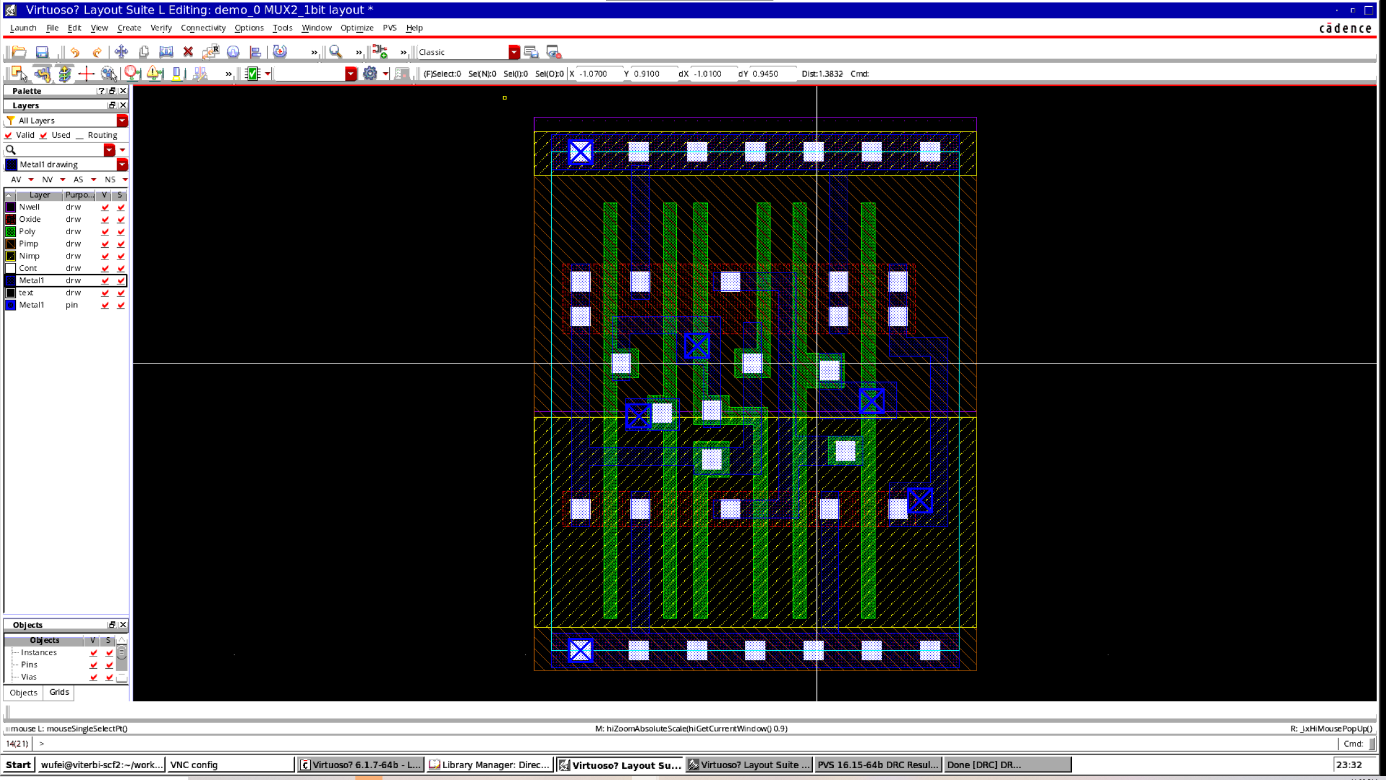
## 1bit mux2

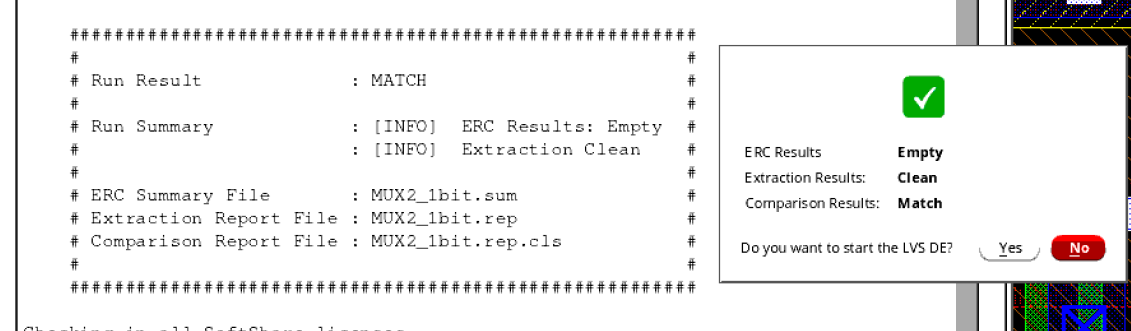
Schematic

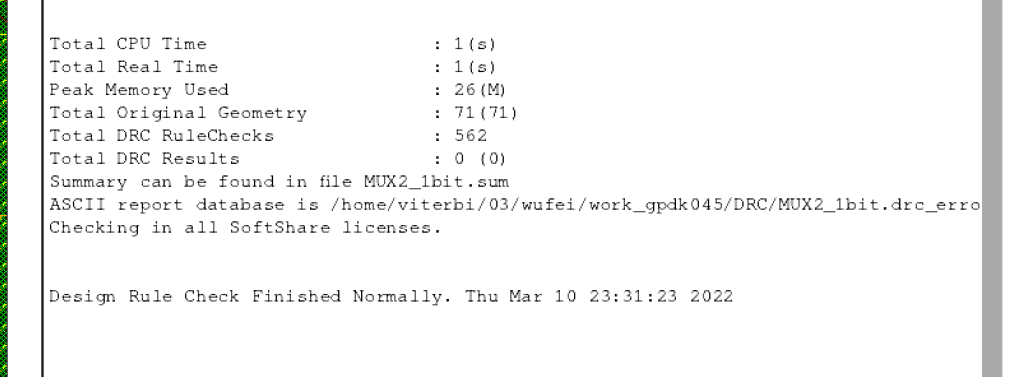






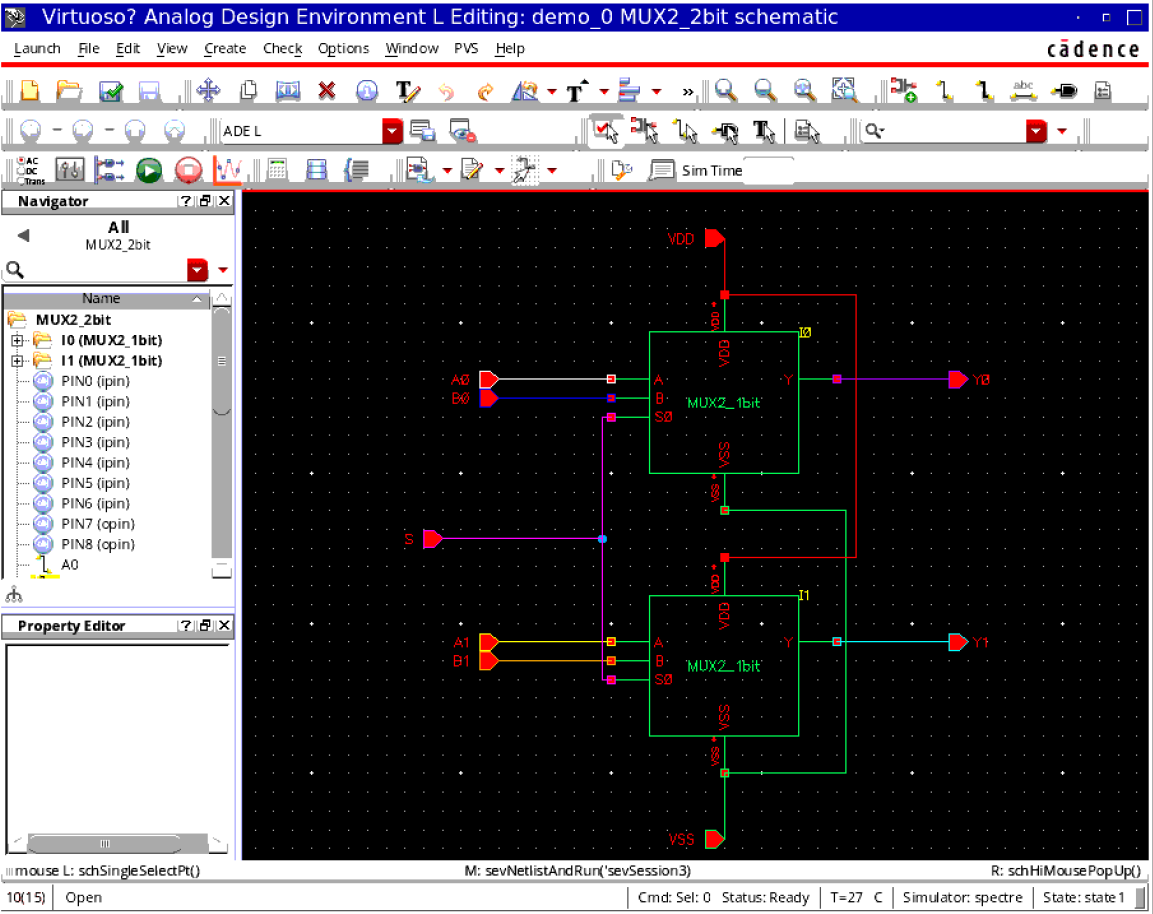


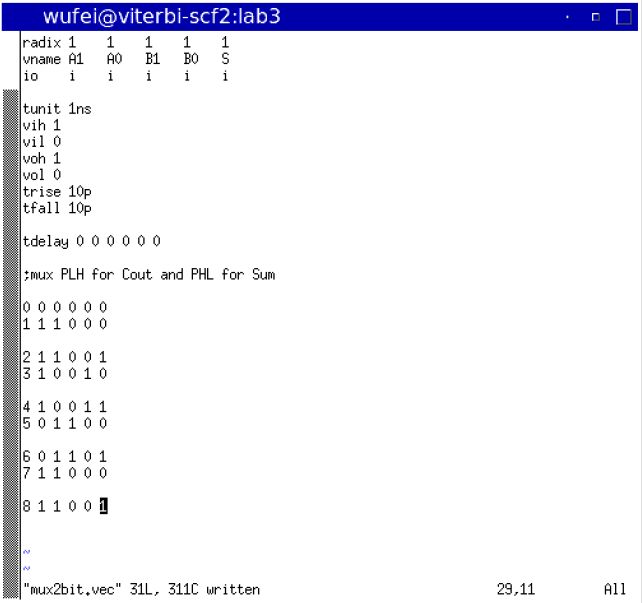




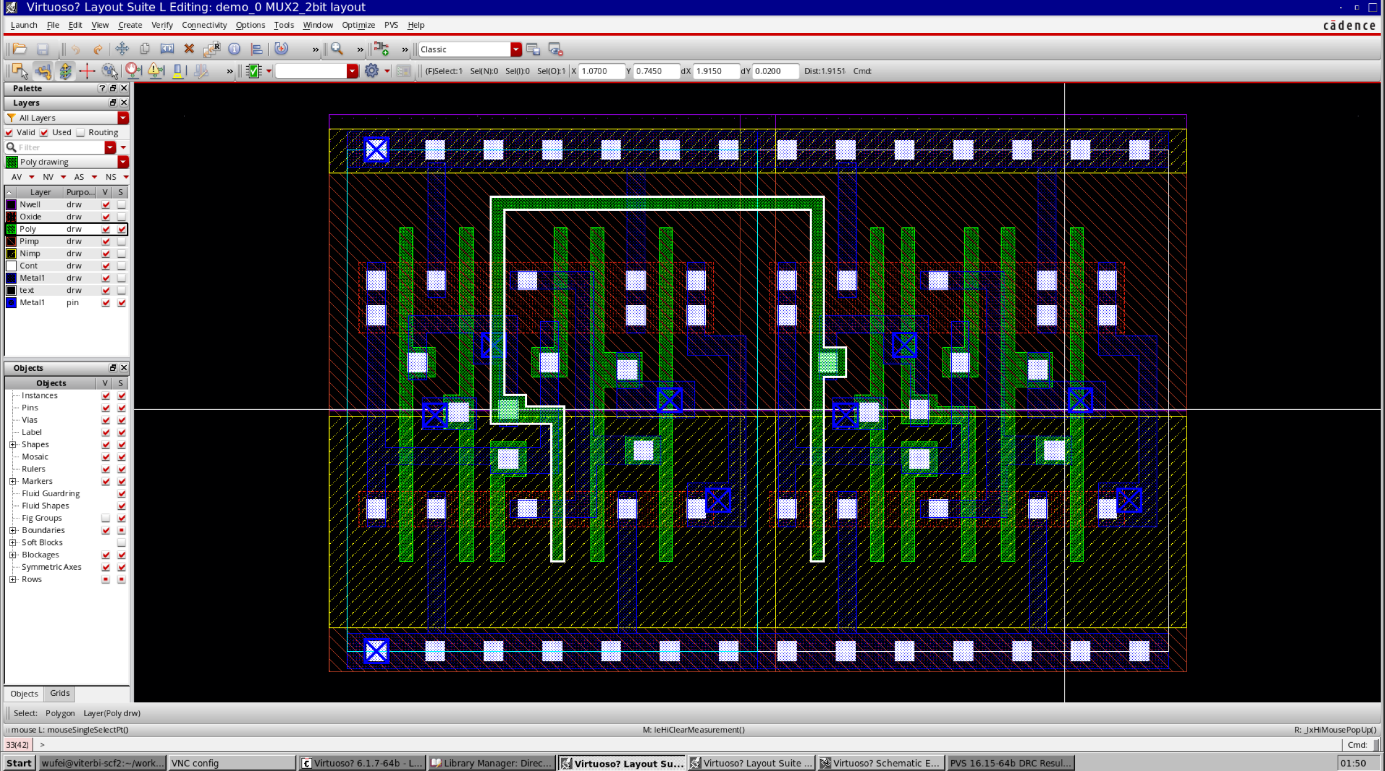
## 2bits mux

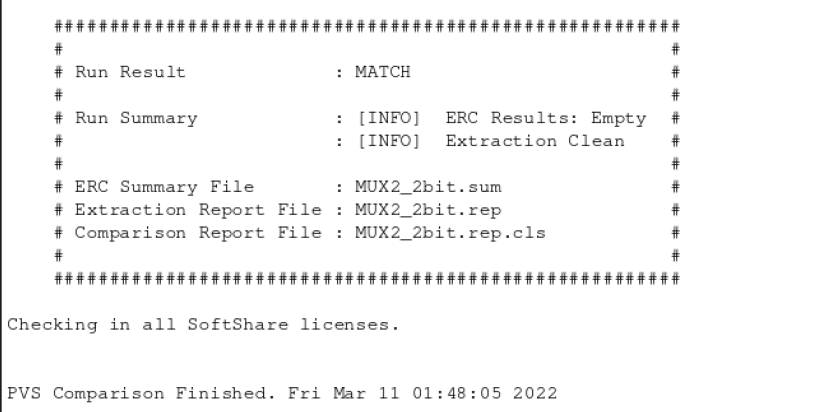
Schematic

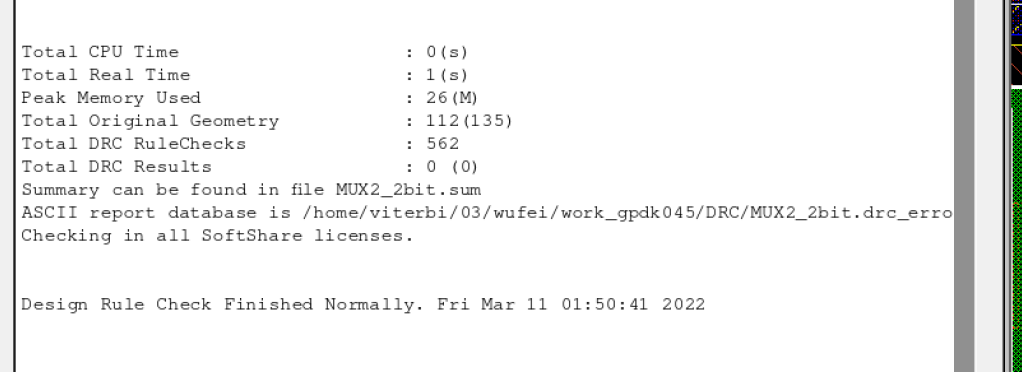










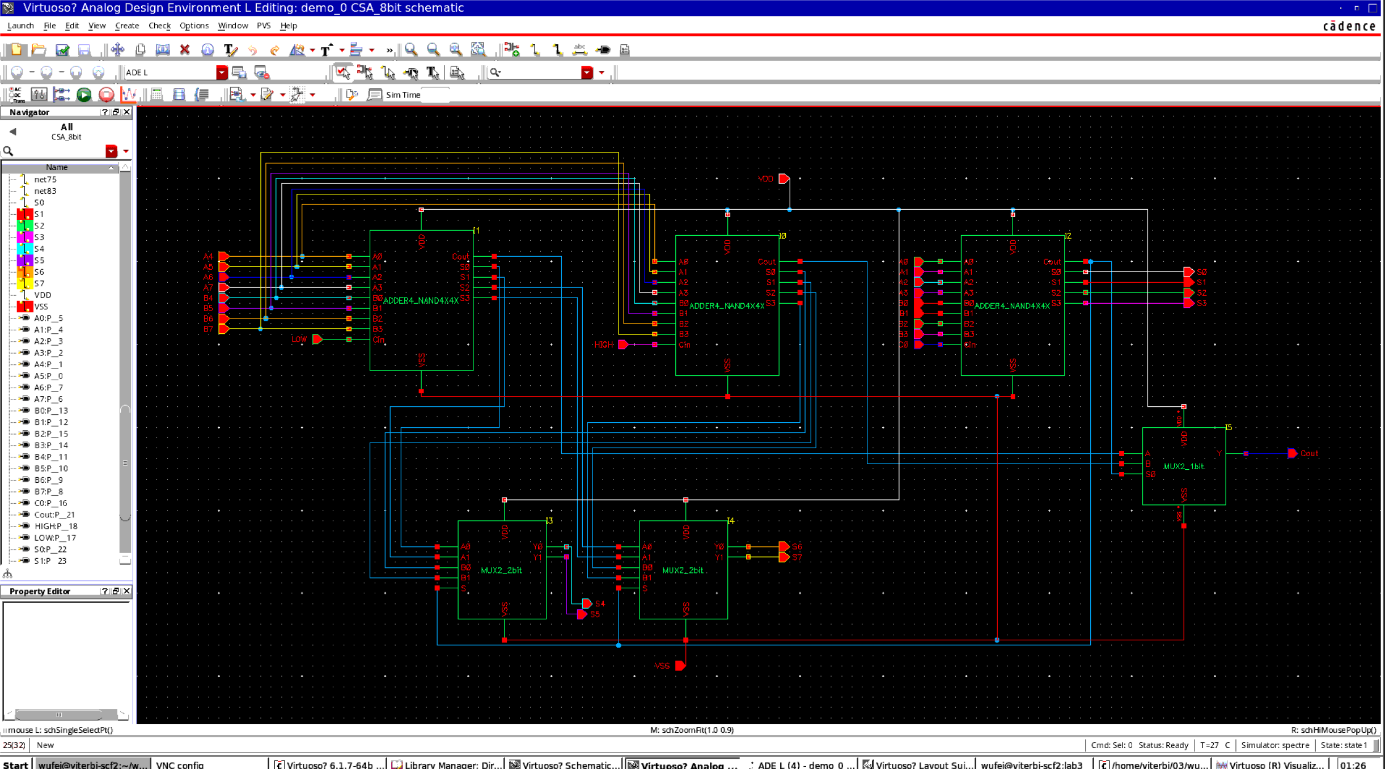


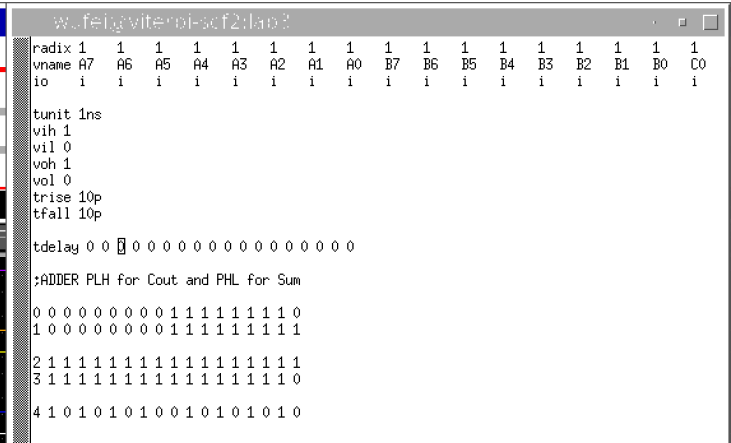
# 8\_bit carry select adder design

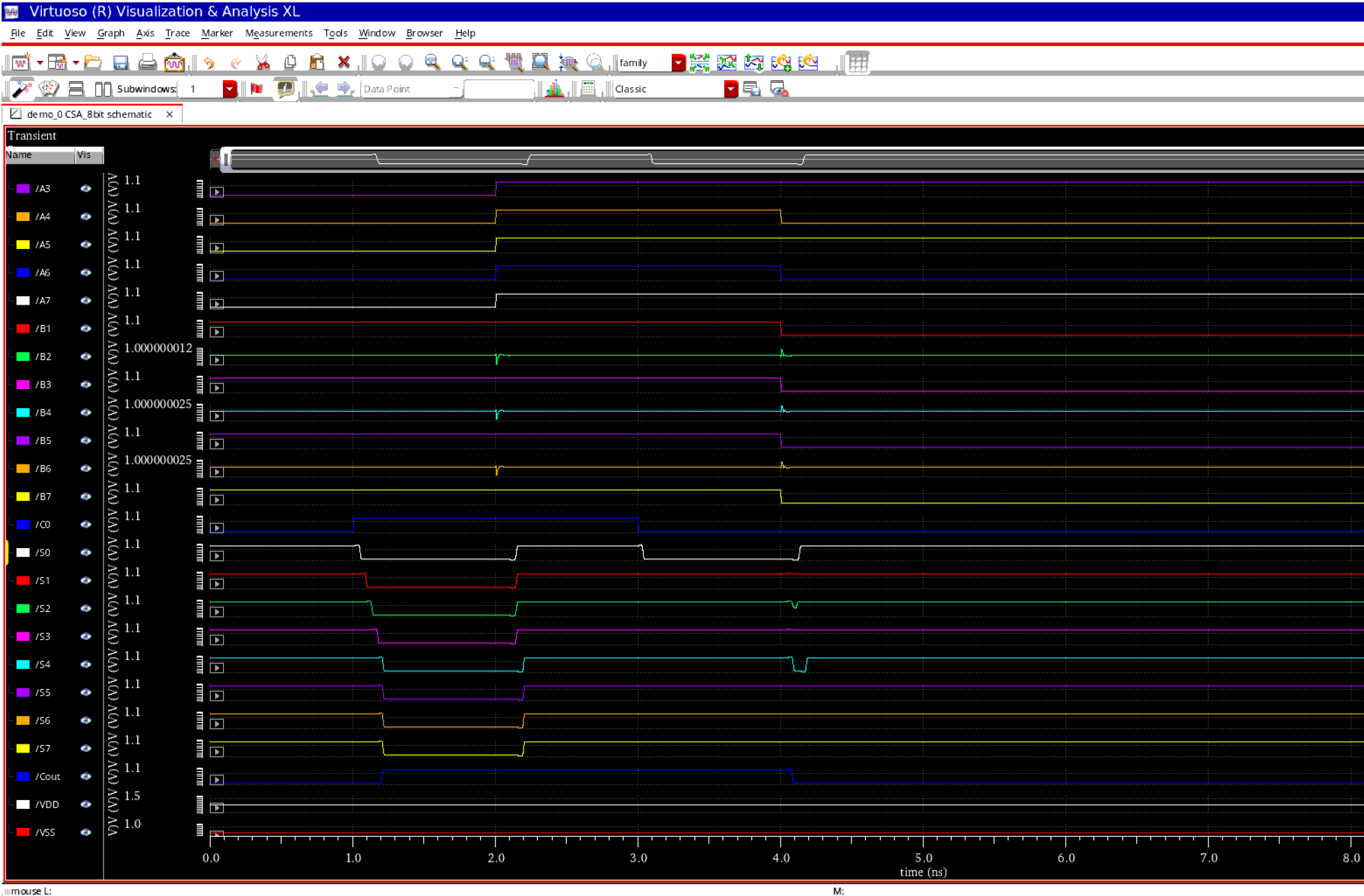
* A CSA computes the outputs for the later stages, i.e., the one processing the higher significant bits, for two cases of input carry as 1 and also as 0, both, while the earlier stages work on the addition of lower significant bits. By the time, the actual carry bit is calculated by the earlier stages, the higher stages are also done and all it takes to produce the complete result is to use the multiplexer to pass the correct results.
* You may choose any of the circuit recommendations presented during lecture and discussion classes.
* You are welcome to use your 4-bit RCA of lab2 as part of this lab.
* Do a functionality test. Use at least 3 different combinations for <a[8:1],b[8:1],c0>.
* Build a delay measurement circuit. All inputs <a[8:1],b[8:1],c0> should be applied through INV\_1X. Connect an INV\_4X for each output (i.e., <c8,s[8:1]>).</c8,s[8:1]></a[8:1],b[8:1],c0></a[8:1],b[8:1],c0>
* Find an input transition which results in worst case delay for output C8. Repeat for S8.
* For this input transition, measure the average delay.
* Draw the layout for the 8-bit CSA.
* Draw layout for the delay measurement circuit and measure average delay of C8 on the extracted view using the same worst case input transition.

## CSA\_8bit

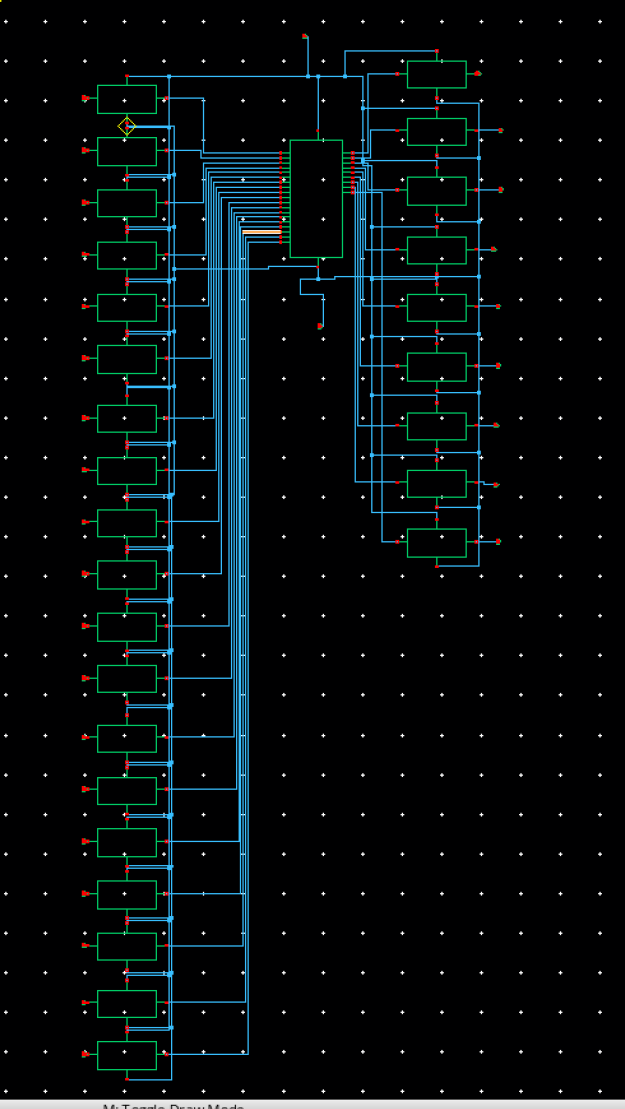
### SCH

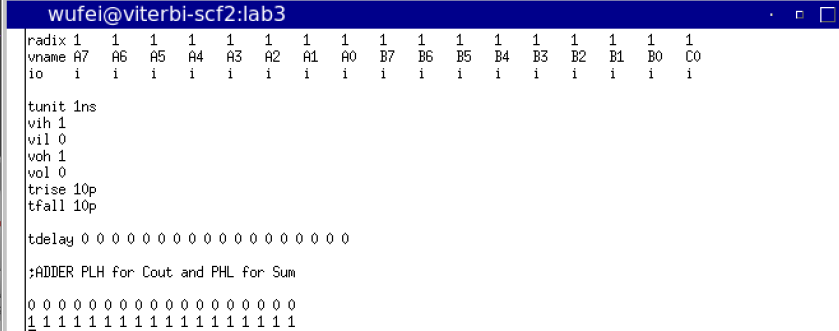


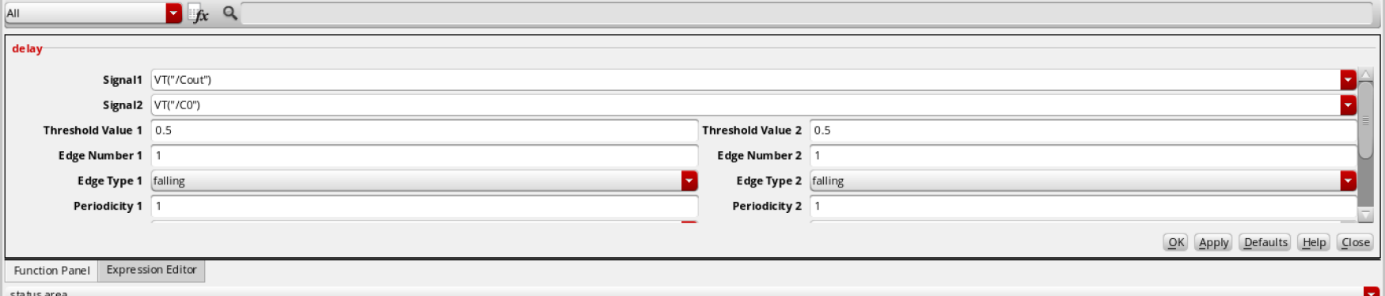


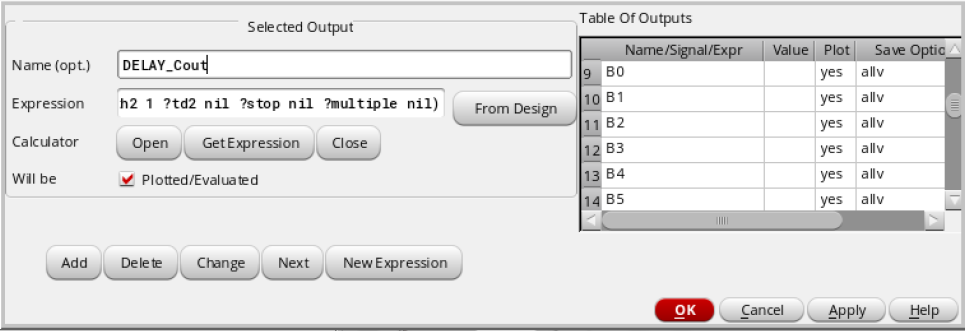


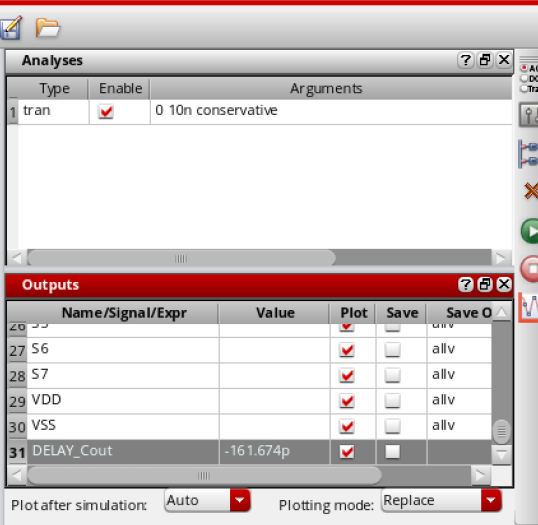
## TEST











## LAYOUT

