ECE115C – Digital Electronic Circuits
Project Testbench

This provides a method for you to test your design in Cadence. "□" denotes a blank space explicitly.

1. Copy the test file **`tb_project.tar`** to your working directory (default: `~/ee115c`)

`>cp□/w/class.1/ee/ee115v/ee115vta/ee115c_w20/tb_project.tar □~/ee115c`

2. From your `~/ee115c` directory, extract the archive with the following command:

`>tar□-xvf□tb_project.tar`

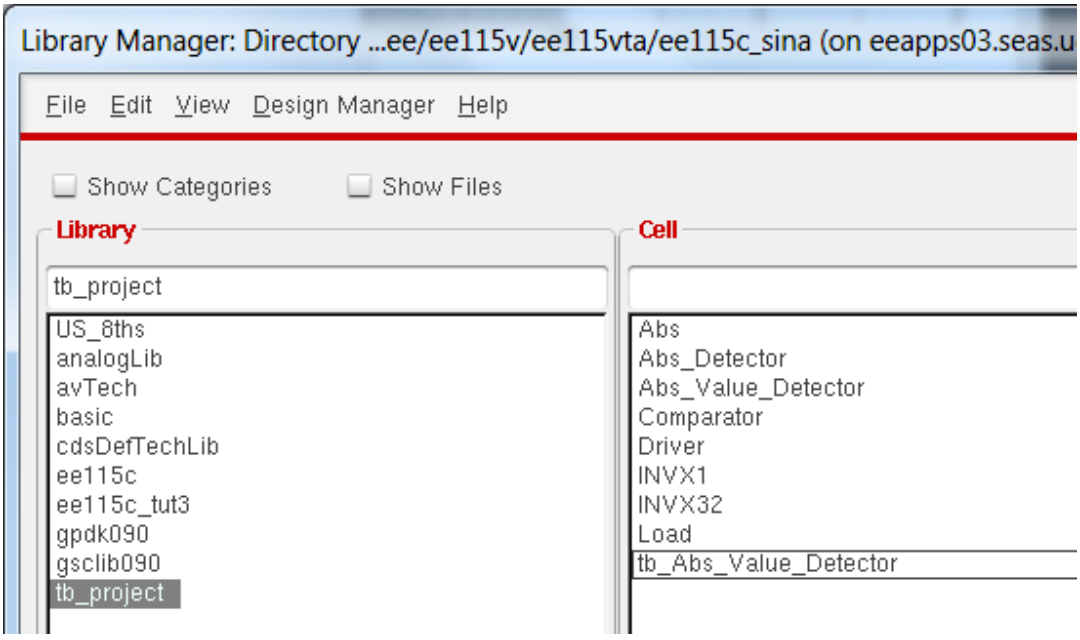
This would create a directory "**`tb_project`**" under your working directory.

3. Edit **`cds.lib`** file in your working directory to add the following line at the end.

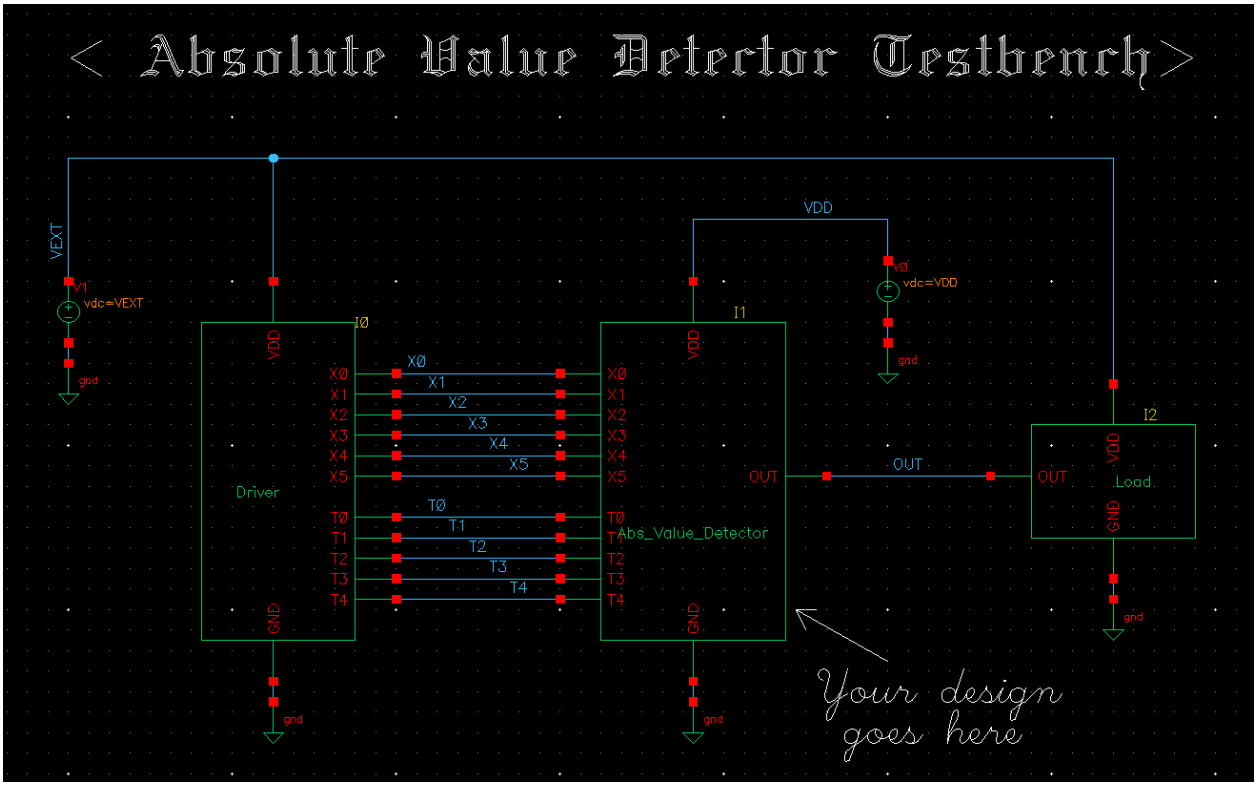
`DEFINE□tb_project□~/ee115c/tb_project`

This tells Cadence to consider the directory we just copied as a Cadence library.

4. Now launch Cadence as usual. You should now see a new library test_project.



Open the schematic "**`tb_Abs_Value_Detector`**".



Replace the **"Abs_Value_Detector"** (this is an example design) in the library "tb_project" **with your own absolute value detector** design. Please make sure that your design is named **"Abs_Value_Detector"**. Also make sure that you have the following pins on your design:

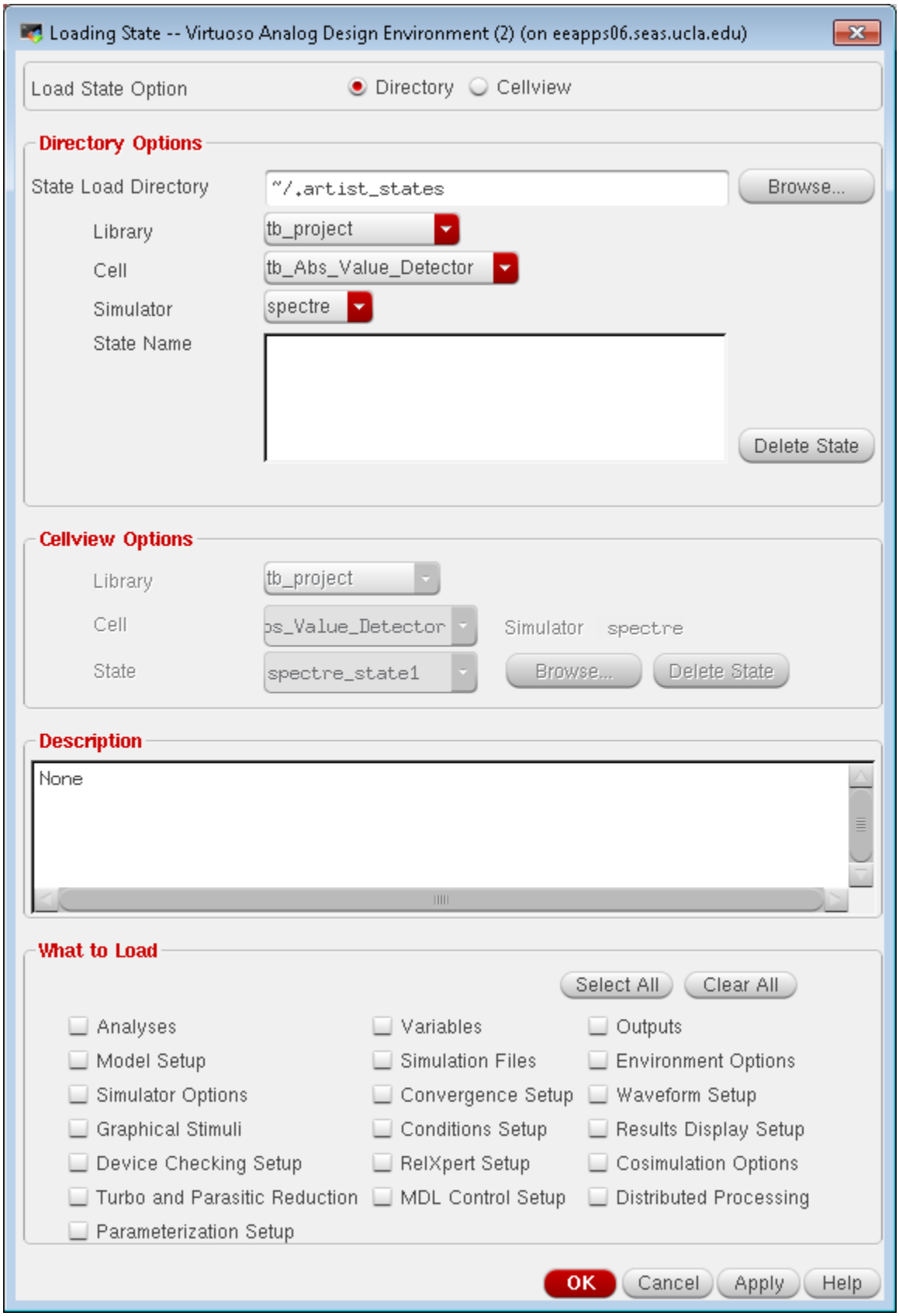
> **X0, X1, X2, X3, X4, X5, T0, T1, T2, T3, T4, VDD, GND,** and **OUT.**

The above pins are case sensitive. It is important to maintain this consistency in your design name and pin names of the design, for the verification steps to be posted later. Please check the port names in the sample design provided with this testbench to verify that your design and port names are correct.

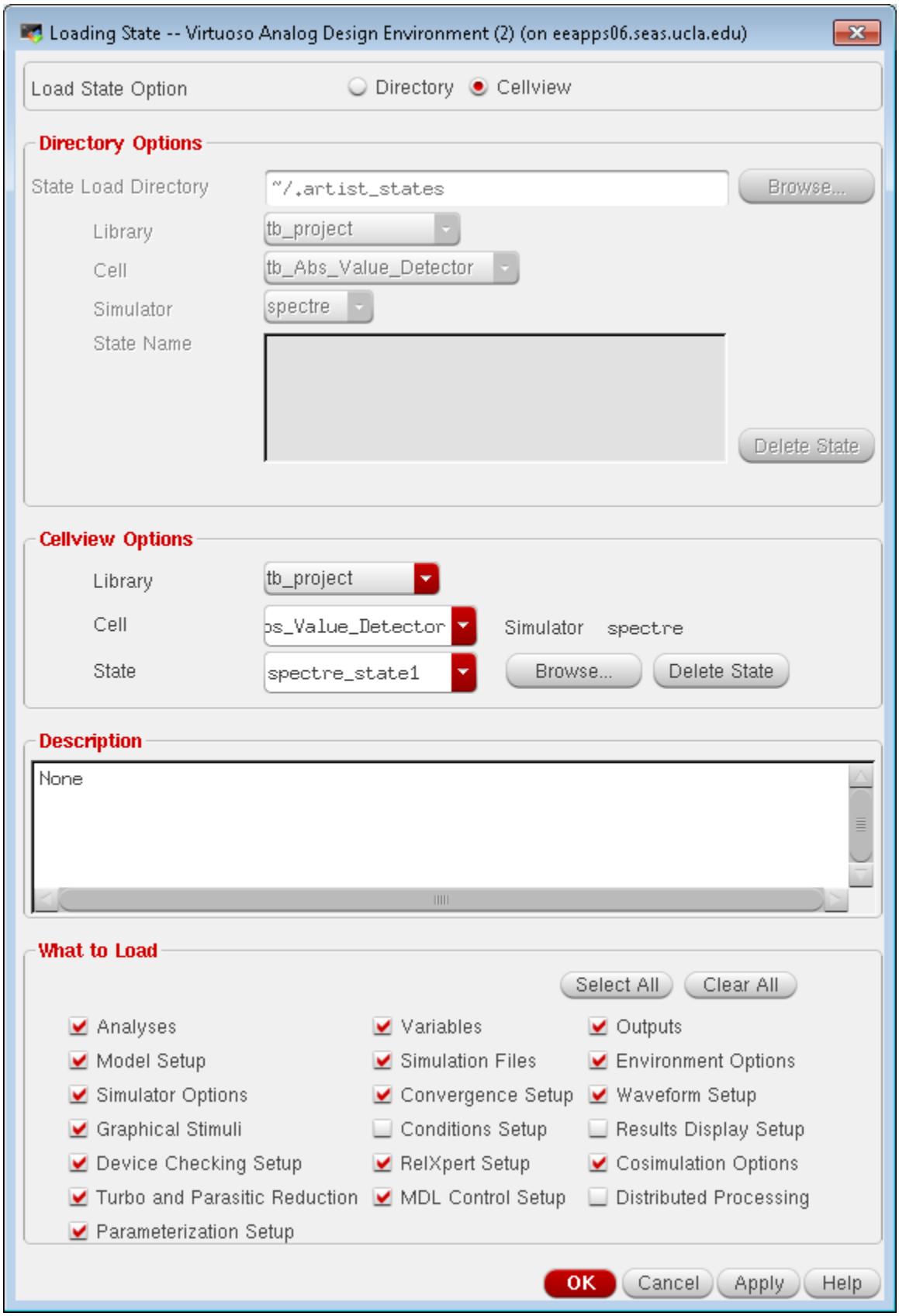
Note: Once you make sure that the pins names and design name is consistent, you can copy the schematic view of your Absolute Value Detector design to the "tb_project" library. **You can still retain the symbol view** that you have been provided, to allow very easy replacement of the adder in the testbench schematic.

Save the schematic for "tb_Abs_Value_Detector" after your design has been replaced.

Now in the schematic window of tb_Abs_Value_Detector click "Launch → ADE L". When Analog Design Environment opens, go to: "Session → Load State…". The following window shows up.



Choose "`Cellview`" in "`Load State Option`", the window becomes:

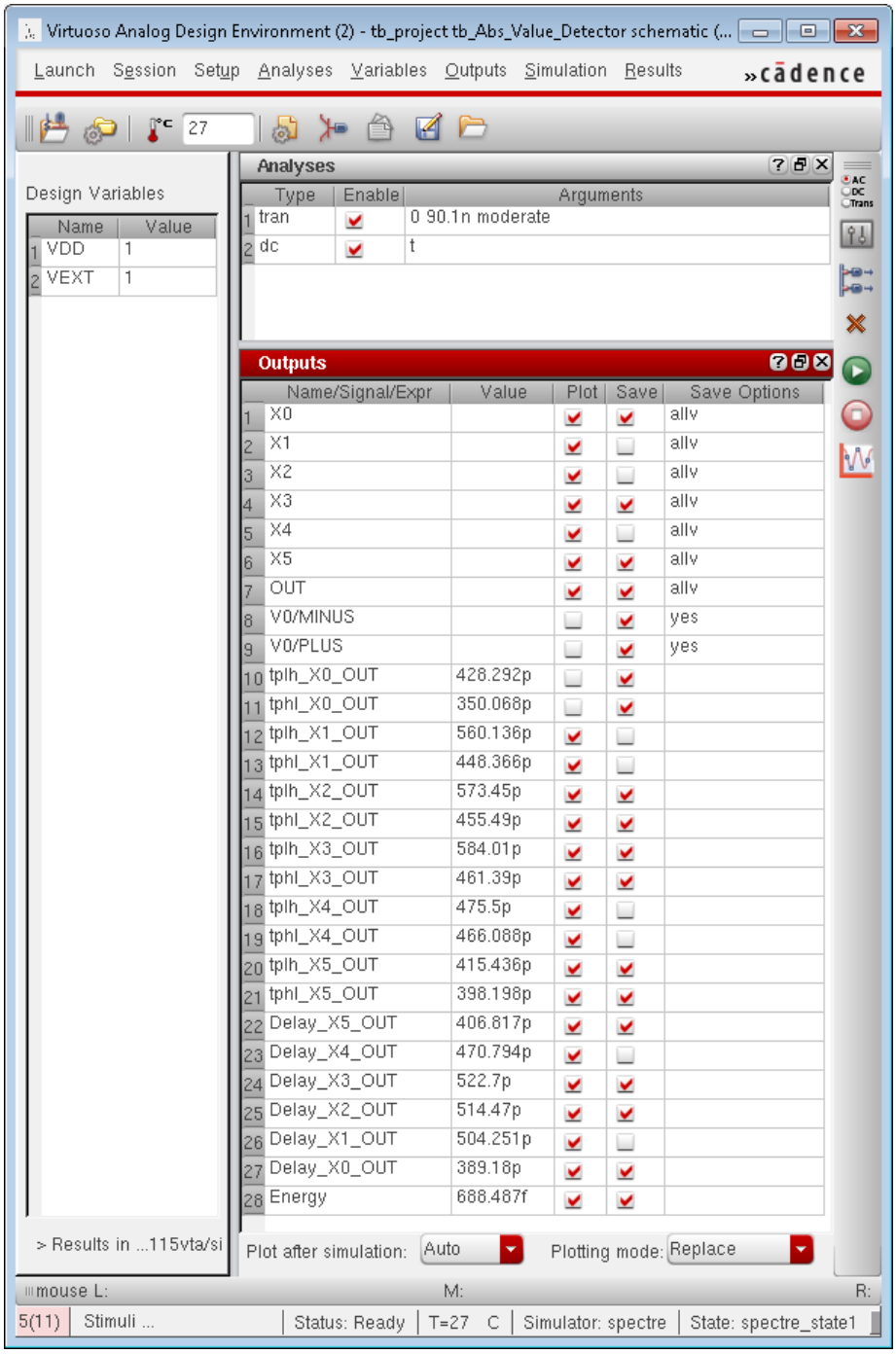


The state "`spectre_state1`" is just the simulation settings we need, click "`OK`".

This will set the variables needed for the testbench, the outputs to be evaluated and the analysis to be performed.

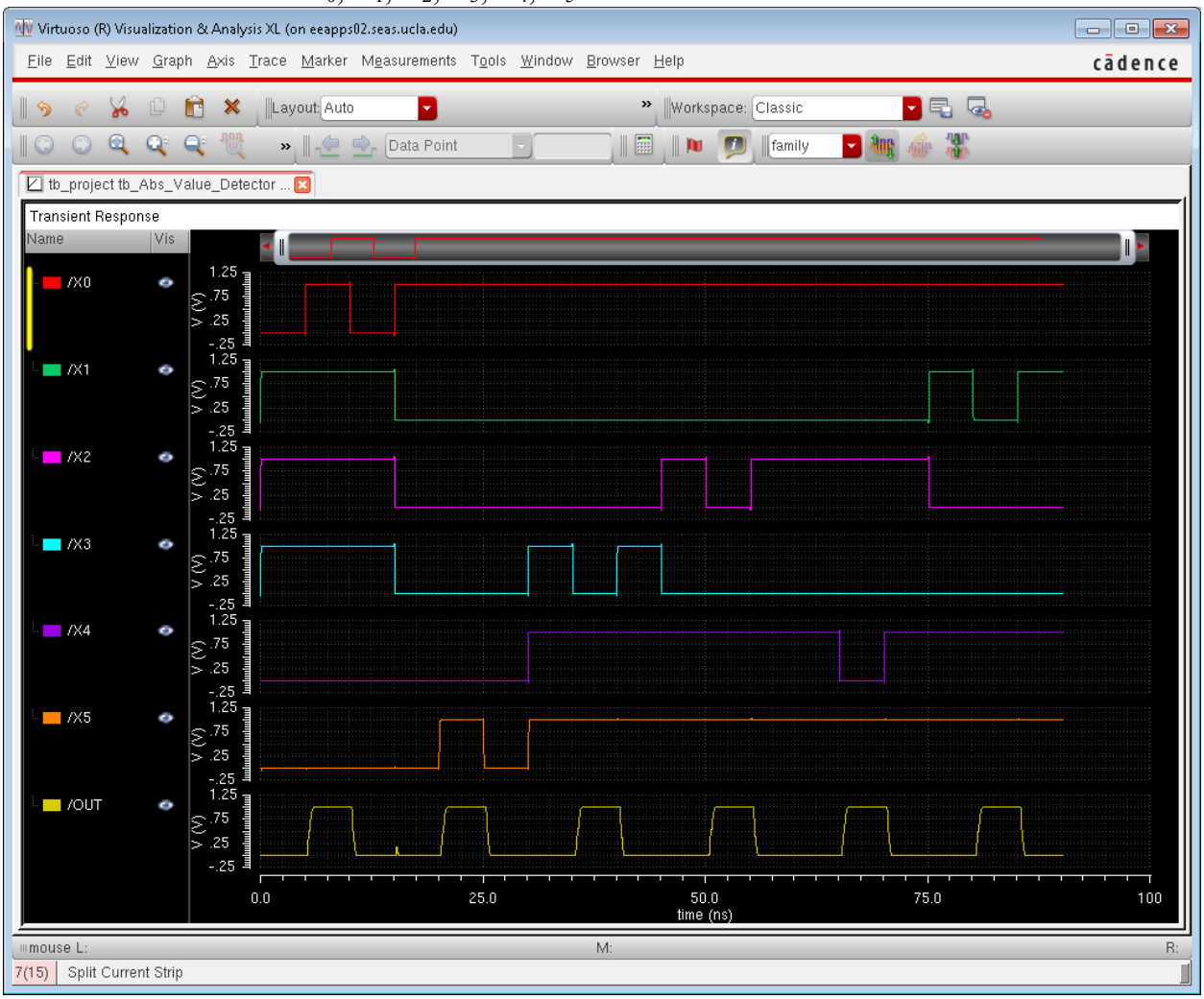
The inputs include 16 transitions with input pattern changes due to the probability distribution given earlier in the project description. Another pattern due to the same probability distribution would be used to check your final submission later.

Now you can run the simulation. For the sample design given to you, operating at 1V, you will see an output as follows: (`VDD` and `VEXT` have to always match).



This shows the delay of this design from X_0, X_1, X_2, X_3, X_4 and X_5 to OUT.
Energy Consumption in this example is 688.487fJ.
Delay is the average of t_{pLH} and t_{pLH} in each case (Maximum delay and maximum t_{pLH} or t_{pLH}, should all be less than 1ns). In the example shown above Delay_X3_OUT is the critical path.

Transient waveforms of X_0, X_1, X_2, X_3, X_4, X_5 and OUT are shown below:



While only a few input combinations are plotted here, you have to make sure that your design functions for all combinations (even the ones that are not tested in this testbench).

GOOD LUCK!