# ToolsSimulationMemoryStaticNoiseMargin

From UVA ECE & BME wiki

# Tutorial for finding Static Noise Margin using simulation

**SNM Simulation Guide**

This is a step by step guide on how to obtain the static noise margin (SNM) of a SRAM cell through simulations in the Cadence Spectre environment. To start with, Fig. 1 shows a diagram of both an SRAM cell (6T) and the butterfly SNM curves of this cell.

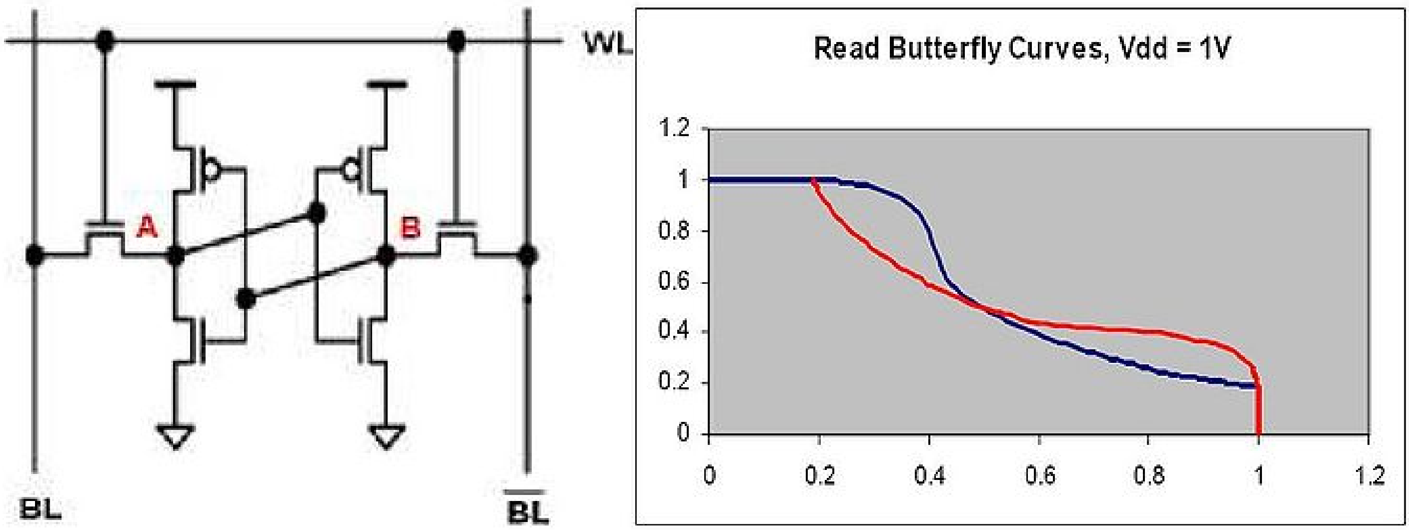
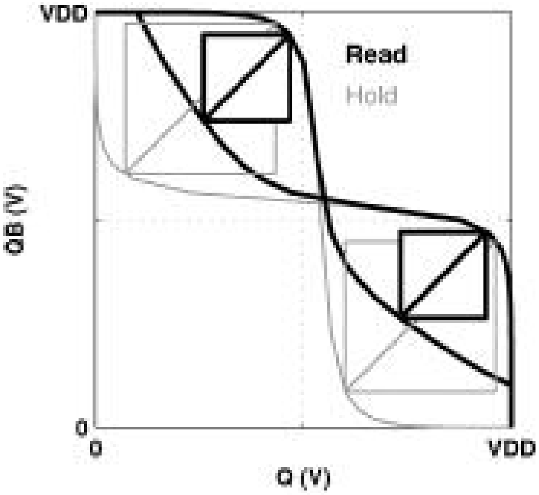


Figure 1: Left - 6T SRAM cell. Right: SNM curves during a read access.

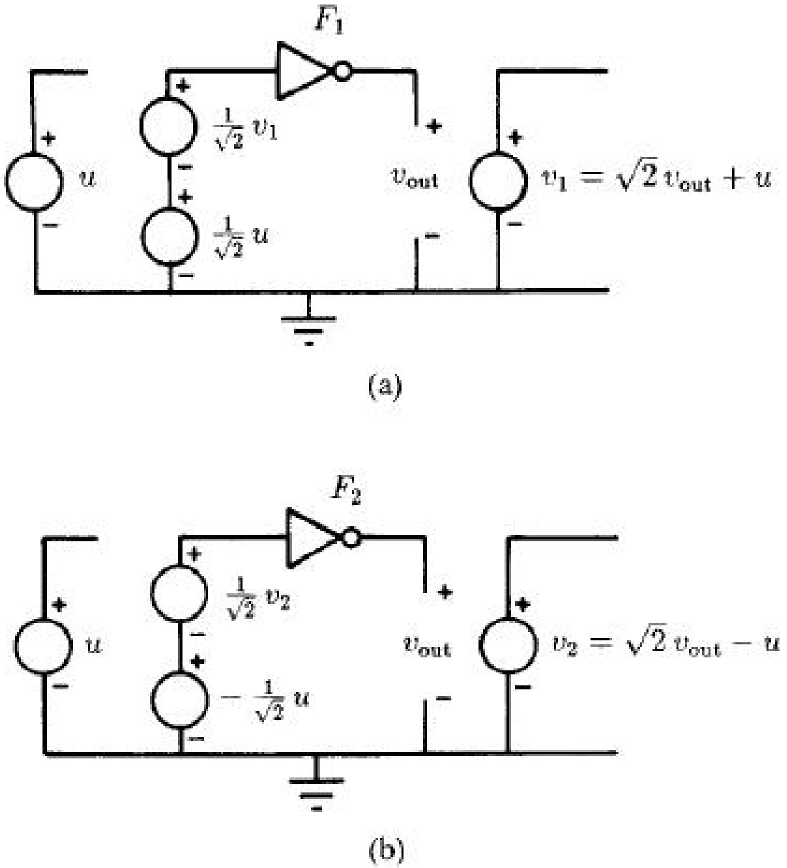
The butterfly curves on the right side of Fig. 1 represent the SNM of the cell, which is a way to quantify the stability of the SRAM cell in the presence of noise. The actual SNM can be graphically represented as shown in Fig 2. In this figure, two squares are fit between the voltage transfer characteristic curves. These squares are the largest possible squares that can be fit between the two inverter characteristic curves of the SRAM cell (actually 4, 2 for the read state, and 2 for the hold state). In order to find the SNM of the cell, you can measure the side lengths (not diagonal lengths!) of both of these squares, and the smaller of the two lengths is the SNM of the SRAM cell. In an ideal case (no variations considered), these squares should be of equal size, and therefore only one square would need to be considered, however considering variations and transistor mismatch, these curves may not be mirror images of each other, and therefore both squares should be considered.



- Figure 2: This figure shows the largest squares which can fit in between the butterfly curves. Taking the side length of the smaller square yields the SNM of the cell [2].

It is pretty easy to see how to obtain the SNM graphically, however it is not the most efficient way. Obtaining this SNM through simulation is considerably more difficult, thereby justifying this tutorial. To start with, please read (or at least have in hand) the paper referenced here [1] because I will refer to this work quite a bit throughout the remainder of this discussion. Seevinck et. al. develop an exact circuit model used to measure the SNM of an SRAM cell in [1]. For purposes of time and space, I will not revisit the steps they take to derive this model. The circuit model derived in [1] is shown here in Fig. 3. In this model, assuming a 6T cell, F1 represents one of the cross coupled inverters in the SRAM cell along with one access transistor, and F2 represents the other inverter with its corresponding access transistor. If you are using something other than a 6T cell, these F1 and F2 blocks should represent the characteristics of your particular SRAM cell.

Looking back at Fig. 1, in order to calculate the SNM, we first sweep node A from GND to VDD and observe the output at node B. This will give us one of the two curves (say the blue curve in Fig. 1) needed for the SNM. In this first plot, the voltage at node A is the independent variable (x –axis) and the voltage at node B is the dependent variable (Y-axis). Next we want to sweep node B from GND to VDD, and observe the output at node A. In this plot, we now have the voltage at node B as the independent variable, and the voltage at node A as the dependent variable. This will give us the second curve for the SNM. In order to find the largest square that can fit between the curves, we need to find the two points of largest separation between the two inverter curves. This is quite difficult to do, and that's what the model developed by Seevinck et. al. provides the ability to do.

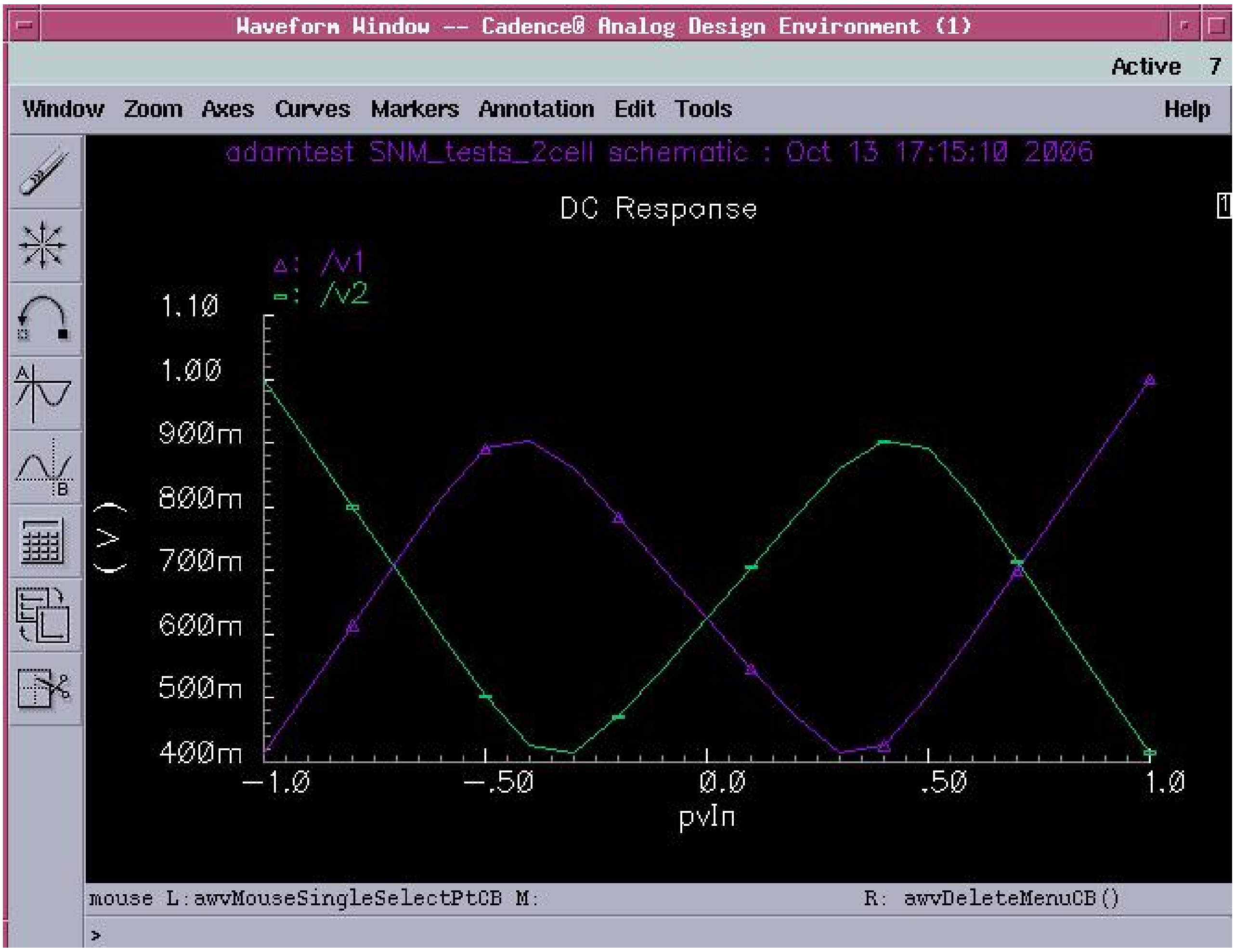


- Figure 3: Models developed by Seevinck et. al. to derive the obtain the SNM of an SRAM cell [1].

In the model shown in Fig 3., u is the DC voltage source to sweep, in order to obtain the transfer characteristics. V1 and V2 are the respective outputs, but they are transposed onto what is called the “u-v” axis in [1]. Plotting V1 and V2 will look something like what is shown in Fig. 4. As you can see, these two curves are the butterfly curves, but they are transposed onto a different axis. The nice thing here is that in order to find the maximum square that can fit in between each side of the transfer characteristic, we can simply take the difference of the two curves and observe the maximum and minimum of the result. An example is shown in Fig. 5. The curve /v1mv2 in Fig. 5 represents the difference between the two curves shown in Fig. 4, and the circled points represent the max and min points of concern. First take the absolute value of these two points, then multiply the smaller of the two by 1/sqrt(2) to obtain the SNM of the cell (you multiply by this factor because you want the length of the side of the square, not the diagonal).

I have attached the sub-circuits that I used (thanks to Professor Calhoun for creating most of these) to do these simulations. The design variables are as follows:

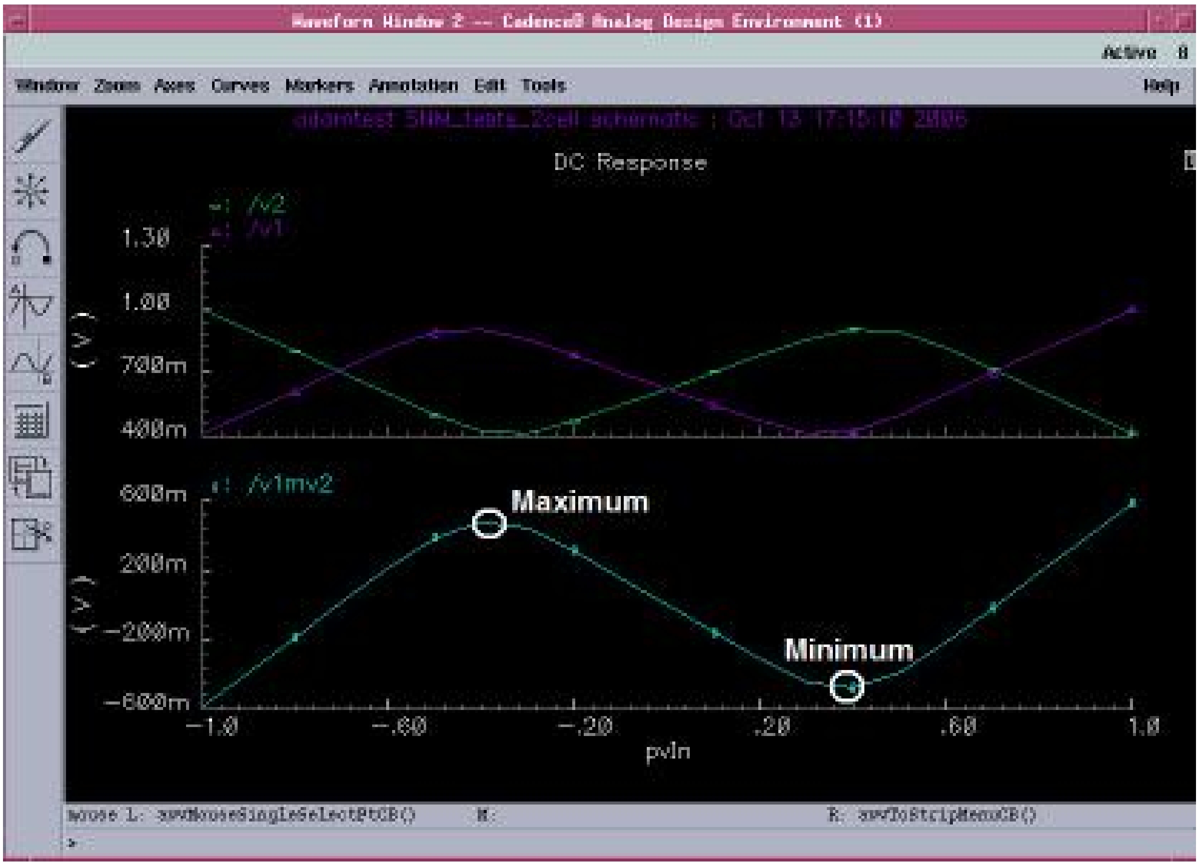
- **Aw:** access transistor width
- **Pw:** PMOS transistor width
- **Dw:** drive transistor width
- **PvIn:** the value of “u” as described above. Sweep this variable from –Vdd to Vdd for your SNM simulations.
- **Vsrc:** Vdd

- 

- Figure 4: Butterfly curves transposed onto the "u-v" coordinate axis as discussed in [1].

These sub-circuits are setup for a particular 90 nm environment, so they may require some tweaking to fit to the particular environment you’re working in. For instance, it may be necessary to update the transistor model names, and change the gate lengths for these netlists to work. I have also attached a picture of the actual schematic I have created (A note here: you have to zoom in on the attachment or the schematics will look garbled, this goes for the other schematic that I attached as well). It is nice to sometimes connect the actual sub-circuit netlist to something visual in order to really understand what’s going on.

One last thing to note is that the sub-circuits and schematics provided here are setup to provide the hold SNM. To calculate read SNM, connect the proper word lines (gates of the access transistors in F1 and F2) to VDD and not to GND. This is the only change necessary to go from hold to read SNM. Hope this tutorial helps, I will post another short blurb soon on how to use this to calculate the DRV, and how to get this DRV using monte carlo simulations as well.



- Figure 5: The /v1mv2 (blue) curve shows the difference between the upper two curves.

- [1] E. Seevinck et. al., "Static-Noise Margin Analysis of MOS SRAM Cells." IEEE Journal of Solid States Circuits, Vol.SC-22, No. 5, Oct. 1987.
- [2] B. Calhoun et. al., "Static Noise Margin Variation for Sub-threshold SRAM in 65-nm CMOS." IEEE Journal of Solid State Circuits, Vol. 41, No. 7, July 2006.

-- Main.AdamCabe - 31 Oct 2006