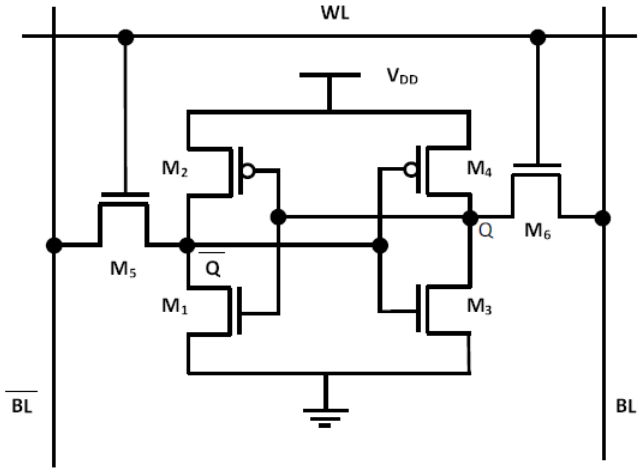
Name:_____

Session 2: Memory Design

The objective of this session is to evaluate the performance of different SRAM cell designs. We will evaluate them in terms of delay (read/write) as well as stability (i.e. noise margins).

1. Design of a 6T SRAM cell

The picture below describes the 6T cell design. This cell has a pair of inverters (M1-M4) and two access transistors M5 and M6.



This cell needs a careful transistor design as the strength (i.e. W/L ratio) of the transistors is crucial to write new values in the cell (Q and Q_b) and, at the same time, perform read operations without losing the content. The nMOS transistors in the cross-coupled inverters must be the strongest. The access transistors are of intermediate strengths and the pMOS transistors must be weak. To achieve a good layout density, tall of the transistors have to be relatively small. Thus, we will make the nMOS transistors in the inverters (Width/Length) 8/2λ, access transistors 4/2λ and pMOS transistors 3/3λ.

We will simulate a 256x32 bit memory array (256 rows and 32 columns). Each memory array has three components: bitline conditioning circuitry (aka. precharge logic), the memory array, the write driver and the sense amplifiers. In our experiments, as all of these components interact with the bitlines, we can simplify our designs by carefully defining/reading the bitline values. Thus, we need not to simulate the entire circuit, just the memory array.

Useful functionalities of spice:

- M=xx (multiplier). Add this to an instantiation of a module/subcircuit to reflect that there are xx of these modules/subcircuits in parallel.
- Define initial values of nodes with the IC command. E.g. `.IC V(Q) = 0V`

To perform a read operation, we will precharge the bitline to Vdd. After that, we will activate the wordline and wait for a 500mV difference between BL and BL_b.

1. Simulate the memory cell and assume the WL signal takes 0'5ns to move from 0 to 1 and from 1 to 0 (slope). Measure the read times for the two possible cell values (i.e. 0 and 1).
2. Simulate the cell again. Assume now that precharge voltage is Vdd/2.

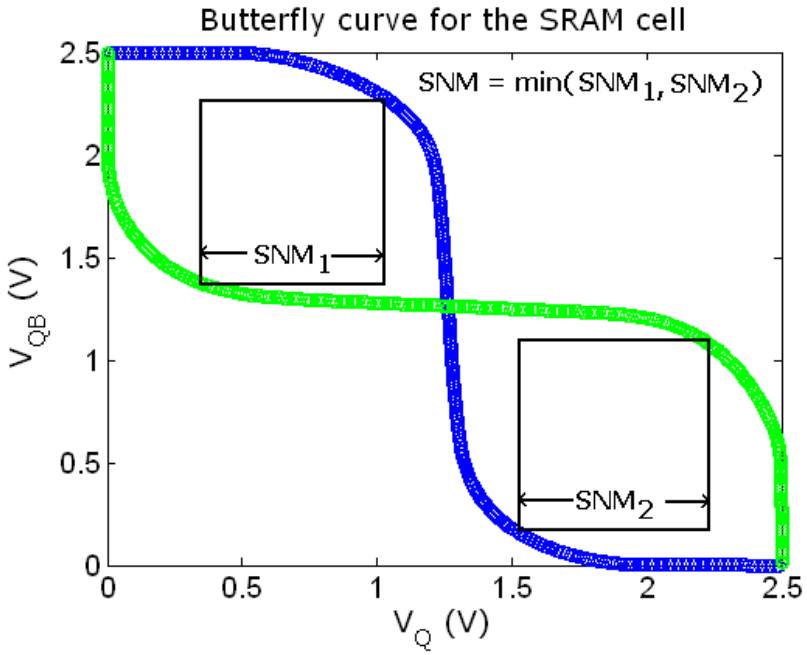
Questions:

1. Compute the read "0" and read "1" times (i.e. delay)
2. Draw the chronogram of the signals involved.
3. Does it take the same time in both cases?

Measure of read delay		
Stored Value	**Action**	**Delay (ns)**
0	Read	
1	Read	
Chronogram		
Does it take the same time to read 0 and 1?	YES	NO
Is there any difference in time when precharging to Vdd/2?	YES **How many ns?**	NO

1b. Static Noise Margin (SNM) computation for SRAM cells

The concept of static noise margin (SNM) for an SRAM cell is shown the figure below. The SNM is defined as the minimum noise voltage present at each of the cell storage nodes necessary to flip the state of the cell. A basic understanding of the SNM is obtained by drawing and mirroring the inverter characteristics and then finding the maximum square between them.



In the figure above, Vdd=2.5V, V_Q is the voltage at node Q and V_{QB} is the voltage at Q_bar.

Simulation method based on graphical technique

We can determine the SNM graphically from the butterfly curve. However, we have to perform one SPICE simulation having clear initialization values. Alternatively, we can compute it through the equations presented in the following paper.You can use the following reference for details.

> Seevinck, E., List, F. J., and Lohstroh, J., "Statis-Noise Margin Analysis of MOS SRAM Cells,", IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 5, October, 1987.

Simulation method based on the insertion of noise voltage sources.

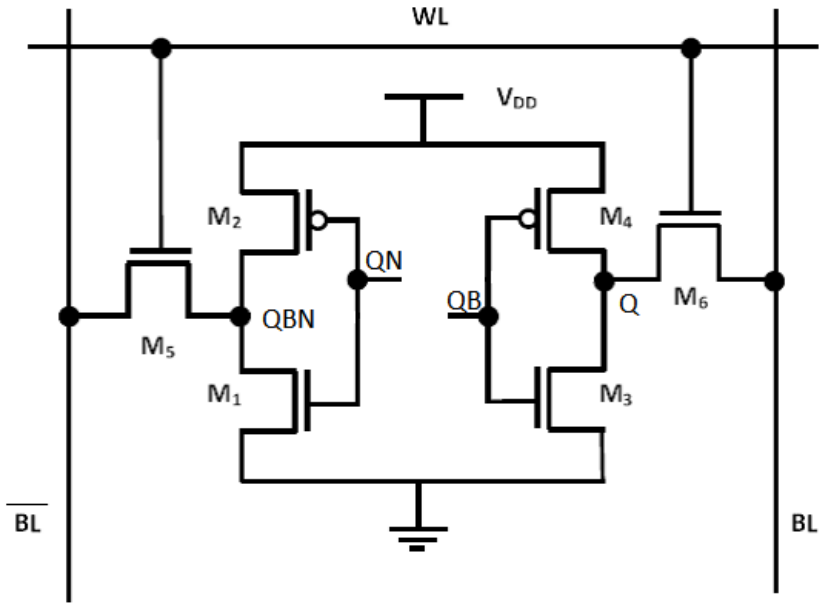
The technique we will implement to find the SNM will be a "trial-and-error" approach. We know beforehand that the SNM is a noise voltage source. Thus, we can include this source in our circuit and just try different values to obtain the smallest voltage where the cell starts to fail. Failing here means that the value stored in the cell is flipped (in hold mode and in read mode). When writing, we can compute also the (write) SNM, in that case, failing means not possible to write.

When read process is initiated, i.e., WL is made logic high, the inverter characteristics changes and hence the SRAM cell gets vulnerable to noise. We are asked to determine the SNM while it's reading (WL=1, Read SNM) and when it is holding a value (WL=0, Hold SNM).

To do so, we will have to decouple the actual voltage values at the inputs of the inverters to the store nodes (Q, and Q_bar). We will insert a voltage source in each of these nodes to simulate the noise. To simplify the design, we will use a "voltage controlled voltage source (VCVS)". That means that the voltage across an ideal voltage source is determined by some other voltage in the circuit. In other words, the voltage at point X is determined as a function of the voltage in point Y. In this way, we will make sure that the noise voltage inserted is the same at both ends of the cell.

The corresponding SPICE code to determine the SNM is given next

First, in your design, rename the storage nodes to match the following figure:



Initialization (i.e. precharge) of the bitlines and internal values:

```
.IC V(BL)   = 1.0V
.IC V(BLB) = 1.0V
.IC V(Q)   = 1.0V
.IC V(QB) = 0V
```

Define the WL as a voltage source (as shown below, WL=0, modify it accordingly):

```
VWL WL 0 DC 0V
```

Insert the following statement to define the VCVS:

```
EVNOISER Q QN VOL='V(QB)-V(QBN)'
```

Insert the next statement to define the noise voltage source:

```
VVNOISEL QB QBN pwl(0ns 0 10ns 0 15ns 1.0 20ns 1.0 25ns 0)
```

Note that we are defining the voltage difference between to nodes. Thus, if the difference is 0, both nodes will have the same voltage (i.e. value). In the definition above, the noise source starts to be 0. Thus the cell should be stable (i.e. V(Q) should be stable for the first 10ns). If this is not the case, then there is a problem in your design. After the 10ns, the noise voltage starts to rise. There is a point where the cell flips its stored value (i.e. V(Q) should move to 0V).

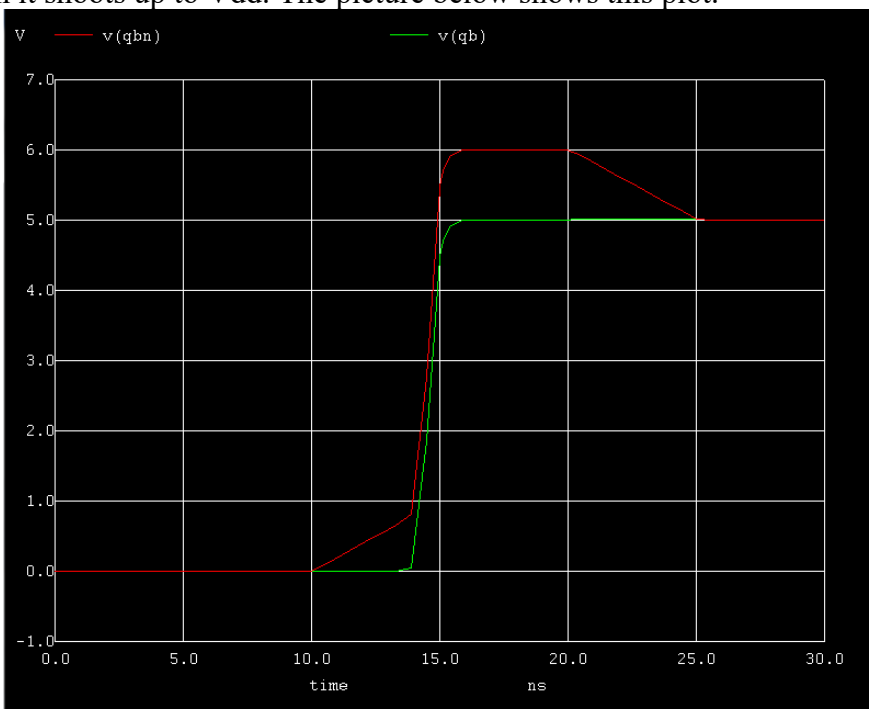
The voltage sources defined here correspond to the initial values stated. If the initial values change, so do the noise voltage definitions (i.e. the polarity has to change).

Finally, insert the simulation time.

```
.TRAN 1n 30n
```

Make sure Q changes its value to 0V.

Plot v(qbn) and v(qb) together. Remember that qbn is the resulting node after inserting noise to qb. Note that qb starts to rise while qb stays at 0. Nevertheless, there is a point where both shoot-up to Vdd (5V in the figure below). This is the trip point. That's the place when the noise is strong enough to flip the value. You can easily see that by the slope of signal V(qbn) that grows up steadily and then it shoots up to Vdd. The picture below shows this plot.



Graphically compute V(qbn). This should be the SNM value.

We can also compute the SNM analytically. We will do a sweep of different noise voltages one by one. For each case, we will check if the value has flipped or not. The smallest noise voltage that flips the value will be the SNM.

To the code above, make the following changes:

1) Change the definition of the noise voltage source to the following:
   `VVNOISE QBN QB DC 0V`

2) Add the following measure statement after the ".tran…"
   `.MEASURE TRAN SNM FIND V(QN) WHEN V(Q) = 0V TD=10ns`
   We want to print out the SNM when the value has flipped. If the value does not flip we will get a failure error (i.e. "out of interval") because Q has never gotten to 0V.

To sweep VVNOISE we will have to insert the following control statements (alternatively, we could type them directly at the ngspice command prompt):

```
.control
destroy all
let testing_noise=0.0
* loop to test values below 1.0V
while (testing_noise<1)
      echo "*********************************************************"
```

```
      echo "*********************************************************"
      echo "Testing robustness for a noise voltage of: $&testing_noise"
      alter vvnoise testing_noise
      run
      echo "*********************************************************"
      echo "*********************************************************"
      let testing_noise=testing_noise+0.1
end
.endc
```

The code above will make 10 simulations and for each one of them it will have a different value of the noise voltage. Replace the loop initialization, bound and step to find out the SNM value with 2 decimals.

Questions:

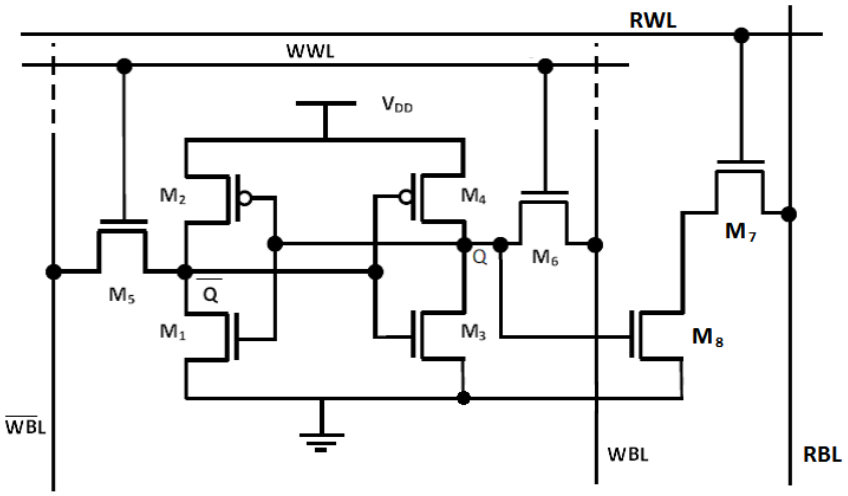
4. Compute the Read and Hold SNM both graphically an analytically. Give the result with 2 decimal values.

Measure of Read Static Noise Margin		
	Graphic value (V)	**Analytic value (V)**
Value of Read SNM (in mV)		
Value of Hold SNM (in mv)		

Attach a print out of the plot with V(q) and Vvnoise for both the SNM voltage and the voltage just before (i.e. the one where Q does not flip).

2. Design of an 8T cell SRAM

The picture below describes the 8T cell design. As you may notice, this is a 6T design extended with 2 (M7 and M8) transistors. Note that the wordline and bitline names also have changed. If you have a closer look at the cell, you can notice that the new transistors and connectors just change the value of RBL according to the cell contents. Hence, this bitline can only be used for read operations (hence the "R" prefix). On the contrary, the other bitlines can be used for both read and write operations (as it was the case in the 6T cell). Nevertheless, in this design, they are only used for write operations.



There are a couple of considerations in order to size these transistors properly. On one side, these transistors serve as a bitline (RBL) discharge path when the value stored in the cell (Q) is "1". As we want to discharge the bitline fast, we would like to make these transistors as wide as possible (to reduce resistance through this path). On the other hand, if we reduce resistance through this path, it means the leakage path through these transistors (when not accessing the cell) is large (as there is small resistance). Plus, having wider transistors will increase the capacity of the wordline (RWL) and the internal node Q. The first may affect the delay of activating the wordline. The second will result in a slower switch speed of the cell. Consequently, we have to find a balance.

One possible design decision to break this tie is to keep the same access speed as the 6T cell. To do so, we have to make sure that (1) the RWL has the same capacity as the WL in the 6T cell; and (2) make sure that the discharge path of the cell is as resistant as the original design. If we start by the second, it means that the combined resistance of M7 and M8 (they are in series) must be equivalent to the resistance of M3 (or M1).

Questions:

1. Size M7 and M8 accordingly. Does it comply with the above criteria 1 and 2?

8T read path transistor sizing			
Transistor	**Width (λ)**	**Length (λ)**	
M7			
M8			
Does the path from RBL to ground have the same resistance as 6T's path from BL to ground?		YES	NO
Does RWL have the same capacitance as the 6T's WL?		YES	NO **Why?**

Simulate the 8T cell. Assume that precharge voltage is Vdd. Assume a voltage difference to Vdd of 500mV to finish the read operation (when RBL falls).

2. Compute the read "0" and read "1" times (i.e. delay)
3. Draw the chronogram of the signals involved.
4. Does it take the same time in both cases?

Measure of read delay		
Stored Value	**Action**	**Delay (ns)**
0	Read	
1	Read	
Chronogram		
Does it take the same time to read 0 and 1?	YES	NO
Discussion **How do you think we can distinguish between reading 0 and a 1 in the 8T cell?** *(hint: read on single-ended sensing)*		

Questions:

5. Compute the Read and Hold SNM.

Measure of Read Static Noise Margin		
	Graphic value (V)	**Analytic value (V)**
Value of Read SNM (in mV)		
Value of Hold SNM (in mv)		

Lab session 2 review form		
How long did it take you to complete the lab?		
Is the documentation clear?	YES	NO
What can be improved?		