EE 201C

Project 2

[Due on Feb 29, 2013]

Submit code and report to:

On Campus Students: Xiao shi(xshi2091@gmail.com)

Online Students: Wei Wu (weiw@seas.ucla.edu)

Email Subject: EE201C_PRJ1_Name_UID

Failure Probability Estimation of SRAM Bit-Cell

- Problem: Given the variations of six threshold voltages in one SRAM bit-cell, try to use Monte Carlo and Quasi-Monte Carlo to estimate the failure probability considering reading access failure.

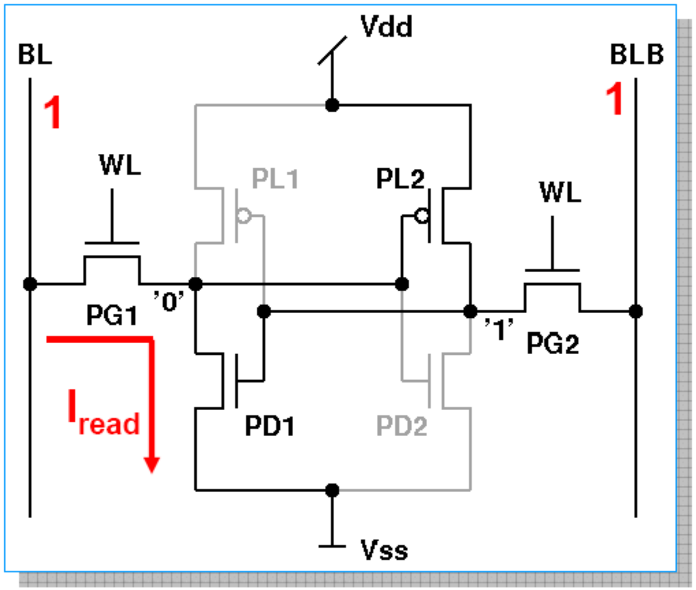
- Six random variables: each Gaussian variable models the process variation on one threshold voltage; the mean and standard deviation are given.

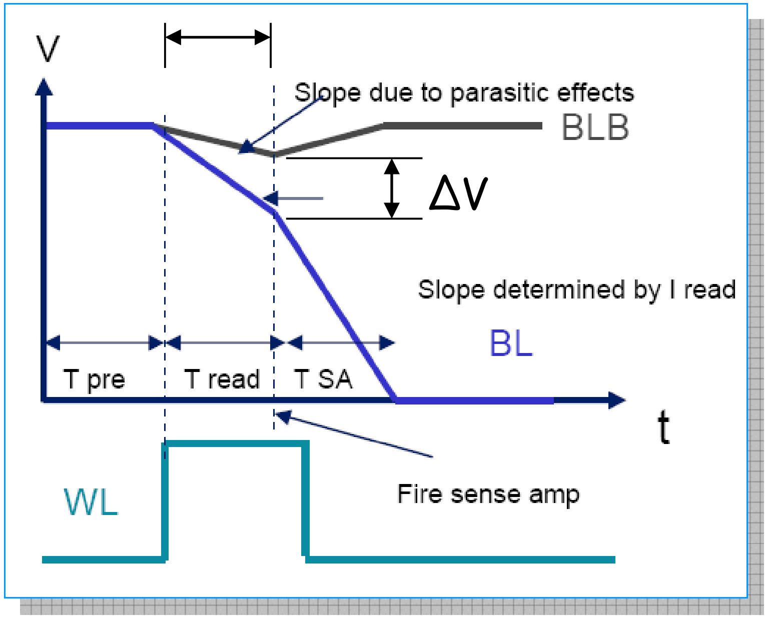
- Use Monte Carlo and Quasi-Monte Carlo to generate samples of these random variables.

- Simulate these samples with any SPICE simulation tools (e.g., hspice, spectre, ngspice, Ltspice, etc.).

- Calculate the figure-of-merit and failure probability: plot two figures to clearly show the convergence of your estimation process.

Reading Access Failure of SRAM bit-cell





- Initially, BL and BLB are pre-charged to '1' (high voltage).
- When reading the SRAM cell, the WL becomes '1', and hold for a while.
- When WL becomes '1', the BLB starts to discharge from high voltage, and produces a voltage difference ΔV between it and BL.
- The time for BLB to produce a large enough ΔV is t_d.
- If t_d is larger than the threshold, this leads to an reading failure.

STEPS

- (1) Netlist (provided)

```
*test the normal SRAM_cell during reading status
*45nm technology
*test using hspice C-2009.09-SP1


* BSIM4 model file
.include "./45nm_bsim4.txt"
.param nvth1 = vthn1
.param nvth2 = vthn2
.param nvth3 = vthn3
.param nvth4 = vthn4

.param pvth1 = vthp1
.param pvth2 = vthp2

.param lmda = 0.05u

.GLOBAL VDD!
.PARAM VDD = 1.3


.PARAM BITCAP = 1e-12
CBL BLB 0 BITCAP
CBLB BL 0 BITCAP

.option post=1

* one inverter
MPL data ndata vdd_node vdd_node  pmos1 l='2*lmda' w='6.5*lmda'
MNL data ndata 0 0 nmos1  l='2*lmda' w='7.5*lmda'

```

declare six random variables

- (1) Netlist (cont.)

```
* SRAM control signals
v_wd word 0 DC=VDD

* Initial status
.IC V(bit) = VDD
.IC V(nbit) = VDD

* SRAM Initial Value
.nodeset v(data)=VDD
.nodeset v(ndata)=0



* Simulation/Anaylsis settings
.temp 27
.op
.tran 1ps 30ps SWEEP DATA=data


*** .DATA goes here
.include "./sweep_data_mc"

* Ouptuts
.print v(bit) v(nbit)
.option post=1


.end
```

Include the data file containing samples

• (2) generate samples in Matlab and write samples into data file (file name is “sweep_data_mc”)

```
% write input file -
fp1 = fopen([path,'/sweep_data_mc'],'w');
fprintf(fp1,'.DATA data\n');
fprintf(fp1,'vthp1  vthp2       vthn1    vthn2    vthn3    vthn4\n');
for a = 1:num
    for b = 1:6
        fprintf(fp1, '%e\t', sample(b,a));
    end
        fprintf(fp1,'\n');
end
fprintf(fp1,'.ENDDATA\n');
fclose(fp1);
```

NMOS:
nominal: 0.466
standard deviation: 0.0466

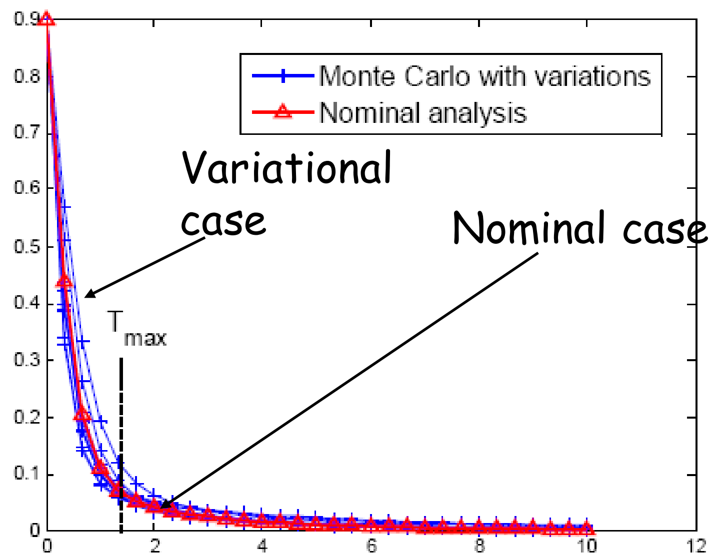
PMOS:
nominal: -0.4118
standard deviation: 0.04118

sweep_data_mc

```
.DATA data
vthp1  vthp2     vthn1    vthn2    vthn3    vthn4
-4.689974e-001  -4.352734e-001  4.033184e-001   4.250936e-001   5.561480e-001   4.454504e-001
-3.994472e-001  -3.878639e-001  4.860415e-001   4.642179e-001   4.354452e-001   4.650910e-001
-4.628508e-001  -4.069125e-001  4.049511e-001   3.995278e-001   5.050741e-001   3.812365e-001
-4.212593e-001  -3.706952e-001  3.525092e-001   4.938265e-001   4.508054e-001   3.942045e-001
-4.606722e-001  -3.938518e-001  5.124492e-001   5.143030e-001   5.171882e-001   4.548818e-001
```

```
-4.048194e-001  -4.484024e-001  4.555632e-001   5.443233e-001   4.979355e-001   4.446302e-001
-4.706913e-001  -3.731160e-001  4.603148e-001   4.713754e-001   4.955397e-001   4.475328e-001
-3.618851e-001  -4.428855e-001  4.692261e-001   5.071763e-001   4.854085e-001   5.242950e-001
-4.475827e-001  -3.885075e-001  3.919548e-001   4.716359e-001   4.808368e-001   4.140370e-001
-3.954146e-001  -4.358929e-001  5.463290e-001   5.341201e-001   4.711644e-001   4.888523e-001
-4.084437e-001  -3.812864e-001  4.183331e-001   5.157581e-001   4.941270e-001   4.528994e-001
.ENDDATA
```

- (3) run simulations on these samples to get their performance merits.
  - run transient simulation on each sample : .tran 1ps 30ps SWEEP DATA=data
  - calculate the voltage difference between BL and BLB: v(bit) – v(nbit)
  - if voltage difference larger than 1.296 volt (based on hspice simulation results), this sample is successful. Otherwise, this is a failed sample.
  - With different SPICE simulator, the failure probability would be different. This is okay only if you can demonstrate the convergence figures of your estimation!



- (4) Plot two figures regarding figure-of-merit and failure probability.
  - – Figure of Merit (FOM): extensively used to quantify the accuracy of probability estimation

$$\rho = \frac{\sqrt{\sigma^2_{\mathcal{P}_r}}}{\mathcal{P}_r}.$$

Pr is the probabilistic estimation $\sigma_{\mathcal{P}_r}$ is the standard deviation of ***Pr***.

– Figure-of-Merit: should be smaller than 0.05 so as to stop the Monte Carlo and Quasi Monte Carlo method.

• (5) The figure-of-merit and failure probability for Monte Carlo method. You should generate similar figures for your own MC and Quasi-MC methods.

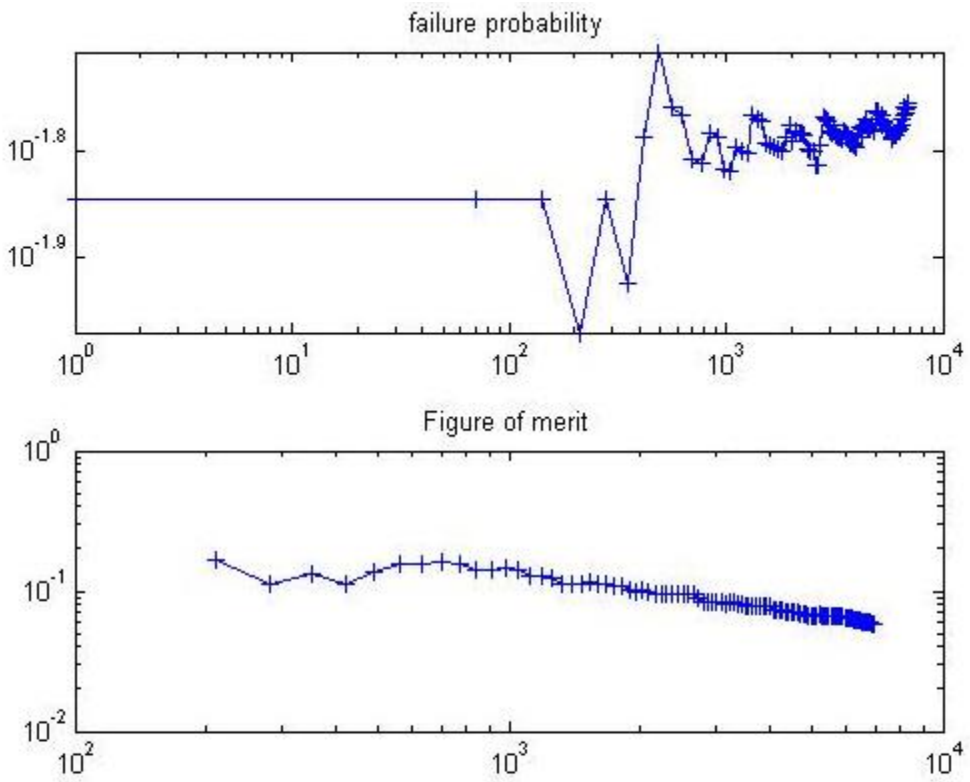


figure-of-merit:=0.05882 and failure probability:=0.017571 with total 7000 samples

Submission List

- the values of failure probability along with figure-of-merit for your MC and QMC methods.

- the figures including FOM and failure probability for MC and QMC (there should be four figures in total!)

- the matlab code to generate samples and write them into data file.

Due on Feb. 29, 2013 midnight!