RC Mismatch: Monte-Carlo Simulation

Emmanuel Jesus R. Estallo
Electrical and Electronics Engineering Institute
University of the Philippines - Diliman
Quezon City, Philippines
emmanuel.estallo@eee.upd.edu.ph

I. General Approach

For this activity, a .MEAS command is used to get t_d. The SPICE software outputs a file that contains all the values t_d of each run. Python is then used for the statistical processing of the obtained data.

II. Netlist

```
.title RC Mismatch
*edited spice file from sir Louis Alarcon

.options savecurrents seed=random

* Polysilicon resistor models
.model rpoly_n R rsh=100 tc1=-800u tnom=27C
.model rpoly_p R rsh=180 tc1=200u tnom=27C

* MOM capacitor model
.model cmom C cj=50m tc1=30u tnom=27C
.model cmsub C cj=30m tc1=25u tnom=27C

* Capacitor with bottom-plate parasitic capacitance
.subckt cm  top bottom sub  w=1000u l=2000u
C1              top bottom      cmom w={w} l={l}
Csub    bottom sub      cmsub w={w} l={l}
.ends

R1              in out          rpoly_n w=2u l=20u
X1              out 0 0         cm w=1000u l=2000u

Vs              in 0            pulse(0 1)

.control

let mc_runs = 1000
let run = 1

define gauss(nom, var) (nom + nom*var * sgauss(0))

dowhile run <= mc_runs

        * mismatch
        alter @R1[l] = gauss(20u, 0.01)
        alter @R1[w] = gauss(2u, 0.01)

        let l1 = gauss(2000u, 0.01)
        alter @c.x1.c1[l] = l1
        alter @c.x1.csub[l] = l1

        let l2 = gauss(1000u, 0.01)
        alter @c.x1.c1[w] = l2
        alter @c.x1.csub[w] = l2

        * process
        altermod @rpoly_n[rsh] = gauss(100, 0.01)
        altermod @cmom[cj] = gauss(50m, 0.01)
        altermod @cmsub[cj] = gauss(30m, 0.01)

        tran 10u 1m
        echo {$run}
        meas TRAN t_d FIND time WHEN v(out) = 0.5

        let R1 = @r1[r]
        let C1 = @c.x1.c1[cap]
        set run = "$&run"

        echo {$&t_d} >> mc_RC.dat

        let run = run + 1
end
.endc

.end
```

III. Python Script

```
import numpy as np
import matplotlib.pyplot as plt
import statistics as stat

td = np.loadtxt('mc_RC.dat')
mean = np.mean(td)
std = stat.stdev(td)

mu = np.format_float_scientific(mean,precision=3)
sigma = np.format_float_scientific(std,precision=3)

plt.hist(td, bins=20, rwidth=0.85)
plt.title(f'$t_d$ ($\mu = {mu}$, $\sigma = {sigma}$)')
plt.xlabel('Time Delay [s]')
plt.ylabel('Occurence')
plt.grid(linestyle='--')
plt.show()
```

IV. Histogram

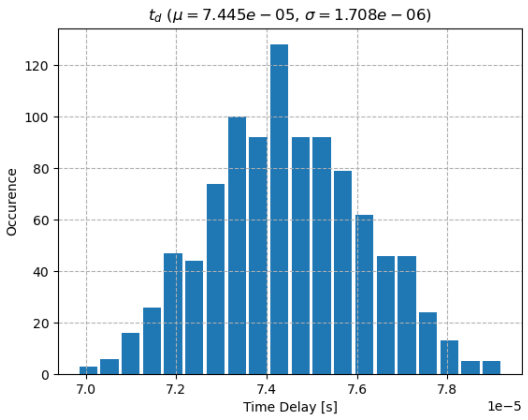

Fig. 1. Generated histogram, $N = 1000$