

# EE577A Lab3

HONGXIANG GAO 8095639536
*University of Southern California*
April 5, 2019

## Contents

**1 Introduction** **2**

**2 FinFET Circuit Design** **3**
2.1 Balanced Inverter Design . . . . . 3
2.2 Balanced NAND2 & NOR2 Design . . . . . 5
2.2.1 NAND2 Simulation . . . . . 5
2.2.2 NOR2 Simulation . . . . . 6
2.3 2x & 4x INV, NAND2 and NOR2 Simulation . . . . . 9
2.3.1 2x Cells Simulation . . . . . 9
2.3.2 4x Cells Simulation . . . . . 12
2.4 1x NOR2 Analysis . . . . . 16
2.5 Full Adder Simulation . . . . . 21
2.5.1 Circuit Build . . . . . 21
2.5.2 Simulation Result . . . . . 23

**3 PTM FinFET DRAM** **24**
3.1 Single Test Case . . . . . 24
3.2 Python Code . . . . . 25
3.3 Simulation Result . . . . . 27
3.4 MATLAB Analysis . . . . . 28
3.4.1 3-D Plot . . . . . 28
3.4.2 Evaluation . . . . . 31

# 1 Introduction

In this lab, we're going to apply the state-of-art FinFET technology to build some basic circuit cells including inverters, NAND gates, NOR gates, full adders and DRAM. Figure 1 shows the basic two structures of FinFET.

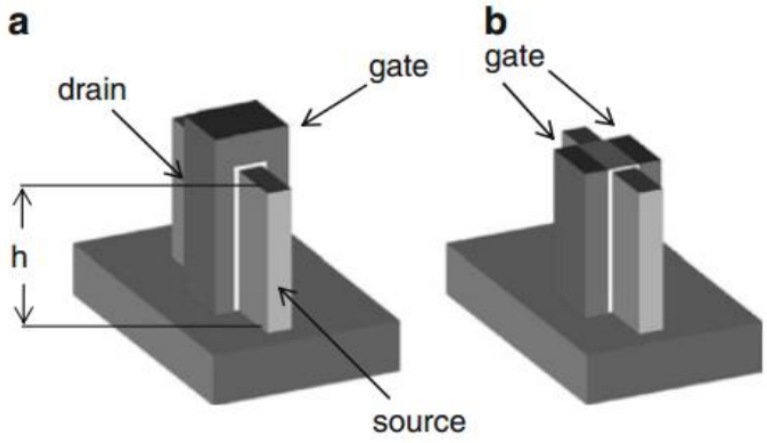


Figure 1: (a) SG-mode FinFET; (b) IG-mode FinFET

# 2 FinFET Circuit Design

## 2.1 Balanced Inverter Design

In this section, we need to design the number of fin of both PMOS and NMOS to balance the pull-up strength of PMOS and the pull-down strength of NMOS. In principle, modifying the number of fin should has the same effect when we change the width of a planar MOSFET. The Hspice code is shown below.

```
************************
*  INV_1X sample hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

*Define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 4
.param n_fin = 3
.param LoadCap = 1f

*Add transistors
mp1 Z A Vdd A pfet L=lg NFIN=p_fin
mn1 Z A Gnd A nfet L=lg NFIN=n_fin

*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VA A 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 10p 10n

*Print waveform file *.tr0
.print V(Z)
.print V(A)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout

*Measurement
```

```
.measure tran tdlay1 trig V(A) val=0.35 TD=1n rise=3
+ targ V(Z) val=0.35 fall=3
.measure tran tdlay2 trig V(A) val=0.35 TD=1n fall=3
+ targ V(Z) val=0.35 rise=3
.measure tran trise trig V(Z) val=0.14 TD=1n rise=2
+ targ V(Z) val=0.56 rise=2
.measure tran tfall trig V(Z) val=0.14 TD=1n fall=2
+ targ V(Z) val=0.56 fall=2
.measure tran avg_power AVG p(Cz)

.end
```

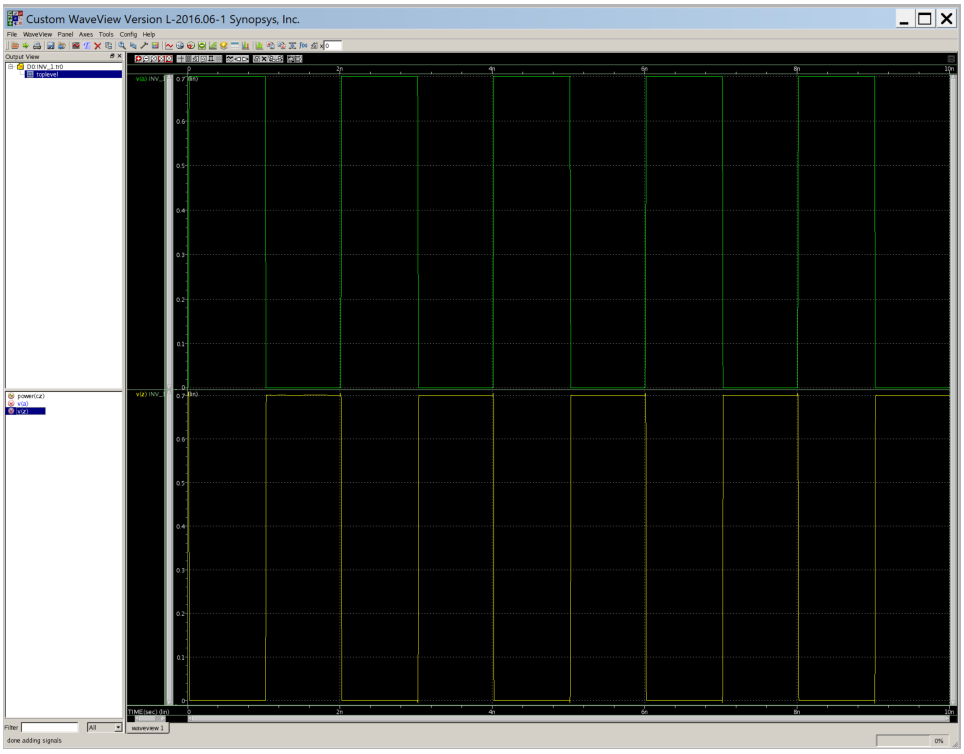


Figure 2: Functionality of 1xINV

The functionality diagram is shown in Figure 2. To balance the rising time and falling time, I choose the PMOS fin number as 4 and 3 for NMOS fin number. The rising delay is about 3.940ps, the falling delay is about 3.516ps. The rising time is 3.265ps and the falling time is 3.371ps, which is quite balanced.

## 2.2 Balanced NAND2 & NOR2 Design

For NAND2 and NOR2 gates design, I applied the 1XINV template, say the ratio between PMOS and NMOS is 4:3. Then the number of PMOS and NMOS of 1XNAND2 is 2 and 3. And similarly, for NOR2, they are 8 and 3.

### 2.2.1 NAND2 Simulation

For NAND2 functionality, the Hspice code and diagram are shown below.

```
************************
*  NAND2_1 sample hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

*Define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 4
.param n_fin = 6
.param LoadCap = 1f

*Add transistors
mp1 Z A Vdd A pfet L=lg NFIN=p_fin
mp2 Z B Vdd B pfet L=lg NFIN=p_fin
mn1 Z A Y A nfet L=lg NFIN=n_fin
mn2 Y B Gnd B nfet L=lg NFIN=n_fin

*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VB B 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VA A 0 PULSE (0 'vdd' 10ps 10ps 10ps 2ns 4ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 1p 20n

*Print waveform file *.tr0
.print V(Z)
.print V(A)
.print V(B)

```

```
*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout

*Measurement
.measure tran rise_delay trig V(B) val=0.35 TD=0.5n fall=1
+ targ V(Z) val=0.35 TD=0.5n rise=1
.measure tran fall_delay trig V(A) val=0.35 TD=3.5n rise=1
+ targ V(Z) val=0.35 TD=3.5n fall=1
.measure tran trise trig V(Z) val=0.14 TD=4.5n rise=1
+ targ V(Z) val=0.56 rise=1
.measure tran tfall trig V(Z) val=0.14 TD=7.5n fall=1
+ targ V(Z) val=0.56 fall=1
.measure tran avg_power AVG p(Cz)

.end
```

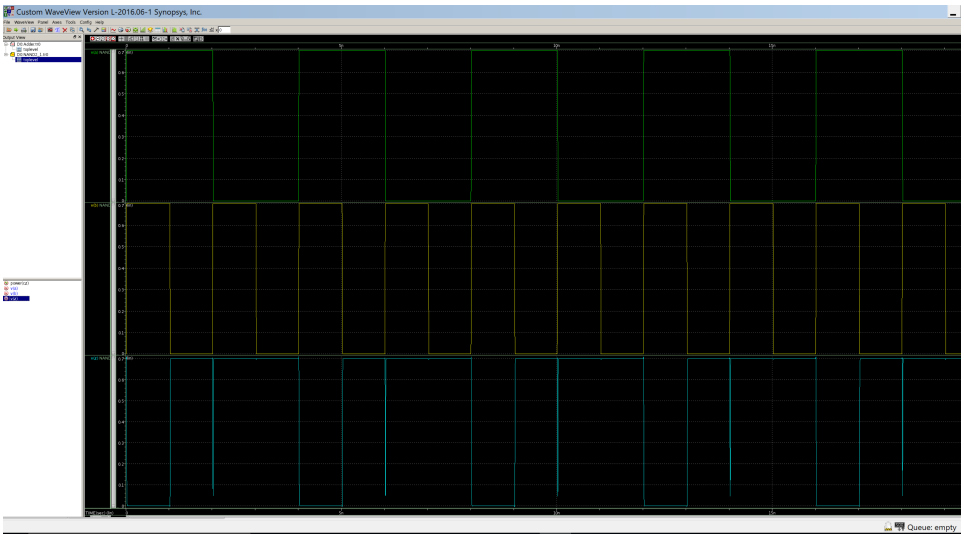

Figure 3: Functionality of 1xNAND2

The rising delay is about 4.150ps, the falling delay is about 4.069ps. The rising time is 3.405ps and the falling time is 3.136ps, which is alsmot balanced.

### 2.2.2 NOR2 Simulation

For NOR2 functionality, the Hspice code and diagram are shown below.

```
************************
*  NOR2_1X sample hspice
************************

.include './hp7nfet.pm'
```

```
.include './hp7pfet.pm'

*Define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 8
.param n_fin = 3
.param LoadCap = 1f

*Add transistors
mp1 Y A Vdd A pfet L=lg NFIN=p_fin
mp2 Z B Y B pfet L=lg NFIN=p_fin
mn1 Z A Gnd A nfet L=lg NFIN=n_fin
mn2 Z B Gnd B nfet L=lg NFIN=n_fin

*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VB B 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VA A 0 PULSE (0 'vdd' 1260ps 10ps 10ps 2ns 4ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 1p 20n

*Print waveform file *.tr0
.print V(Z)
.print V(A)
.print V(B)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout

*Measurement
.measure tran rise_delay trig V(B) val=0.35 TD=8.5n fall=1
+ targ V(Z) val=0.35 TD=8.5n rise=1
.measure tran fall_delay trig V(A) val=0.35 TD=8.5n rise=1
+ targ V(Z) val=0.35 TD=8.5n fall=1
.measure tran trise trig V(Z) val=0.14 TD=8.5n rise=1
+ targ V(Z) val=0.56 TD=8.5n rise=1
.measure tran tfall trig V(Z) val=0.14 TD=8.5n fall=1
+ targ V(Z) val=0.56 TD=8.5n fall=1
.measure tran avg_power_Cz AVG p(Cz)
.measure tran avg_power_mp1 AVG p(mp1)
.measure tran avg_power_mp2 AVG p(mp2)
```

```
.measure tran avg_power_mn1 AVG p(mn1)
.measure tran avg_power_mn2 AVG p(mn2)

.end
```

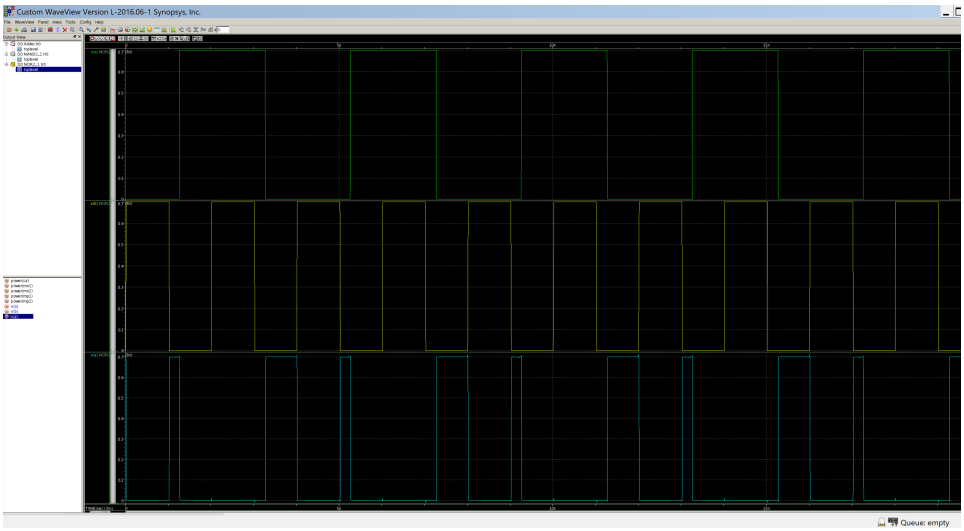

Figure 4: Functionality of 1xNOR2

The rising delay is about 3.282ps, the falling delay is about 4.645ps. The rising time is 3.447ps and the falling time is 3.756ps, which is almost balanced.

Table 1: Simulation of 1xINV, 1xNAND2, 1xNOR2

| | num_P | num_N | Rise Delay | Fall Delay | Rise Time | Fall Time |
|---|---|---|---|---|---|---|
| INV | 4 | 3 | 3.940(ps) | 3.516(ps) | 3.265(ps) | 3.371(ps) |
| NAND2 | 4 | 6 | 4.150(ps) | 4.069(ps) | 3.405(ps) | 3.136(ps) |
| NOR2 | 8 | 3 | 3.282(ps) | 4.645(ps) | 3.447(ps) | 3.756(ps) |

Table 1 shows the conclusion of number of fins, rise and fall delay, rise and fall time.

## 2.3 2x & 4x INV, NAND2 and NOR2 Simulation

### 2.3.1 2x Cells Simulation

Here shows the simulation code of each gate. For 2xINV,

```
************************
*  INV_2X sample hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

*Define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 8
.param n_fin = 6
.param LoadCap = 1f

*Add transistors
mp1 Z A Vdd A pfet L=lg NFIN=p_fin
mn1 Z A Gnd A nfet L=lg NFIN=n_fin

*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VA A 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 10p 10n

*Print waveform file *.tr0
.print V(Z)
.print V(A)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout

*Measurement
.measure tran tdlay1 trig V(A) val=0.35 TD=1n rise=3
+ targ V(Z) val=0.35 fall=3
.measure tran tdlay2 trig V(A) val=0.35 TD=1n fall=3
+ targ V(Z) val=0.35 rise=3
.measure tran trise trig V(Z) val=0.14 TD=1n rise=2
```

```
+ targ V(Z) val=0.56 rise=2
.measure tran tfall trig V(Z) val=0.14 TD=1n fall=2
+ targ V(Z) val=0.56 fall=2
.measure tran avg_power AVG p(Cz)

.end
```

For 2xNAND2,

```
************************
*  NAND2_2X sample hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

*define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 8
.param n_fin = 12
.param LoadCap = 1f

*Add transistors
mp1 Z A Vdd A pfet L=lg NFIN=p_fin
mp2 Z B Vdd B pfet L=lg NFIN=p_fin
mn1 Z A Y A nfet L=lg NFIN=n_fin
mn2 Y B Gnd B nfet L=lg NFIN=n_fin

*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VB B 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VA A 0 PULSE (0 'vdd' 10ps 10ps 10ps 2ns 4ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 1p 20n

*Print waveform file *.tr0
.print V(Z)
.print V(A)
.print V(B)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout
```

```

*Measurement
.measure tran rise_delay trig V(B) val=0.35 TD=0.5n fall=1
+ targ V(Z) val=0.35 TD=0.5n rise=1
.measure tran fall_delay trig V(A) val=0.35 TD=3.5n rise=1
+ targ V(Z) val=0.35 TD=3.5n fall=1
.measure tran trise trig V(Z) val=0.14 TD=4.5n rise=1
+ targ V(Z) val=0.56 rise=1
.measure tran tfall trig V(Z) val=0.14 TD=7.5n fall=1
+ targ V(Z) val=0.56 fall=1
.measure tran avg_power AVG p(Cz)

.end
```

For 2xNOR2,

```
************************
*  NOR2_2X sample hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

*Define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 16
.param n_fin = 6
.param LoadCap = 1f

*Add transistors
mp1 Y A Vdd A pfet L=lg NFIN=p_fin
mp2 Z B Y B pfet L=lg NFIN=p_fin
mn1 Z A Gnd A nfet L=lg NFIN=n_fin
mn2 Z B Gnd B nfet L=lg NFIN=n_fin

*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VB B 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VA A 0 PULSE (0 'vdd' 1260ps 10ps 10ps 2ns 4ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 1p 20n

*Print waveform file *.tr0
```

```
.print V(Z)
.print V(A)
.print V(B)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout

*Measurement
.measure tran rise_delay trig V(B) val=0.35 TD=8.5n fall=1
+ targ V(Z) val=0.35 TD=8.5n rise=1
.measure tran fall_delay trig V(A) val=0.35 TD=8.5n rise=1
+ targ V(Z) val=0.35 TD=8.5n fall=1
.measure tran trise trig V(Z) val=0.14 TD=8.5n rise=1
+ targ V(Z) val=0.56 TD=8.5n rise=1
.measure tran tfall trig V(Z) val=0.14 TD=8.5n fall=1
+ targ V(Z) val=0.56 TD=8.5n fall=1
.measure tran avg_power AVG p(Cz)

.end
```

Table 2: Simulation of 2xINV, 2xNAND2, 2xNOR2

| | num_P | num_N | Rise Delay | Fall Delay | Rise Time | Fall Time |
|---|---|---|---|---|---|---|
| INV | 8 | 6 | 2.794(ps) | 2.196(ps) | 3.641(ps) | 2.950(ps) |
| NAND2 | 8 | 12 | 3.407(ps) | 3.124(ps) | 3.500(ps) | 2.509(ps) |
| NOR2 | 16 | 6 | 2.301(ps) | 3.833(ps) | 2.459(ps) | 3.193(ps) |

Table 2 shows the simulation of all 2x cells.

### 2.3.2 4x Cells Simulation

Here shows the simulation code of each gate. For 4xINV,

```
************************
*  INV_4X sample hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

*Define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 16
.param n_fin = 12
.param LoadCap = 1f

```

```
*Add transistors
mp1 Z A Vdd A pfet L=lg NFIN=p_fin
mn1 Z A Gnd A nfet L=lg NFIN=n_fin

*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VA A 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 10p 10n

*Print waveform file *.tr0
.print V(Z)
.print V(A)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout

*Measurement
.measure tran tdlay1 trig V(A) val=0.35 TD=1n rise=3
+ targ V(Z) val=0.35 fall=3
.measure tran tdlay2 trig V(A) val=0.35 TD=1n fall=3
+ targ V(Z) val=0.35 rise=3
.measure tran trise trig V(Z) val=0.14 TD=1n rise=2
+ targ V(Z) val=0.56 rise=2
.measure tran tfall trig V(Z) val=0.14 TD=1n fall=2
+ targ V(Z) val=0.56 fall=2
.measure tran avg_power AVG p(Cz)

.end
```

For 4xNAND2,

```
************************
*  NAND2_2X sample hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

*define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 16
```

```
.param n_fin = 24
.param LoadCap = 1f

*Add transistors
mp1 Z A Vdd A pfet L=lg NFIN=p_fin
mp2 Z B Vdd B pfet L=lg NFIN=p_fin
mn1 Z A Y A nfet L=lg NFIN=n_fin
mn2 Y B Gnd B nfet L=lg NFIN=n_fin

*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VB B 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VA A 0 PULSE (0 'vdd' 10ps 10ps 10ps 2ns 4ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 1p 20n

*Print waveform file *.tr0
.print V(Z)
.print V(A)
.print V(B)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout

*Measurement
.measure tran rise_delay trig V(B) val=0.35 TD=0.5n fall=1
+ targ V(Z) val=0.35 TD=0.5n rise=1
.measure tran fall_delay trig V(A) val=0.35 TD=3.5n rise=1
+ targ V(Z) val=0.35 TD=3.5n fall=1
.measure tran trise trig V(Z) val=0.14 TD=4.5n rise=1
+ targ V(Z) val=0.56 rise=1
.measure tran tfall trig V(Z) val=0.14 TD=7.5n fall=1
+ targ V(Z) val=0.56 fall=1
.measure tran avg_power AVG p(Cz)

.end
```

For 4xNOR2,

```
************************
*  NOR2_4X sample hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

```

```
*Define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 32
.param n_fin = 12
.param LoadCap = 1f

*Add transistors
mp1 Y A Vdd A pfet L=lg NFIN=p_fin
mp2 Z B Y B pfet L=lg NFIN=p_fin
mn1 Z A Gnd A nfet L=lg NFIN=n_fin
mn2 Z B Gnd B nfet L=lg NFIN=n_fin

*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VB B 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VA A 0 PULSE (0 'vdd' 1260ps 10ps 10ps 2ns 4ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 1p 20n

*Print waveform file *.tr0
.print V(Z)
.print V(A)
.print V(B)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout

*Measurement
.measure tran rise_delay trig V(B) val=0.35 TD=8.5n fall=1
+ targ V(Z) val=0.35 TD=8.5n rise=1
.measure tran fall_delay trig V(A) val=0.35 TD=8.5n rise=1
+ targ V(Z) val=0.35 TD=8.5n fall=1
.measure tran trise trig V(Z) val=0.14 TD=8.5n rise=1
+ targ V(Z) val=0.56 TD=8.5n rise=1
.measure tran tfall trig V(Z) val=0.14 TD=8.5n fall=1
+ targ V(Z) val=0.56 TD=8.5n fall=1
.measure tran avg_power AVG p(Cz)

.end
```

Table 3 shows the simulation of all 4x cells.

Table 3: Simulation of 4xINV, 4xNAND2, 4xNOR2

| | num_P | num_N | Rise Delay | Fall Delay | Rise Time | Fall Time |
|---|---|---|---|---|---|---|
| INV | 16 | 12 | 2.027(ps) | 1.861(ps) | 2.233(ps) | 1.871(ps) |
| NAND2 | 16 | 24 | 2.833(ps) | 2.492(ps) | 3.160(ps) | 2.082(ps) |
| NOR2 | 32 | 12 | 1.705(ps) | 3.022(ps) | 2.057(ps) | 2.862(ps) |

## 2.4 1x NOR2 Analysis

In this section, we need to apply four structures of a NOR2 gate according to the material given about the NAND2 gate and analyze its performance. The four structures of the NOR2 gate is shwon in Figure 5. From left to right, they are SG, LP, IG and IGLP.

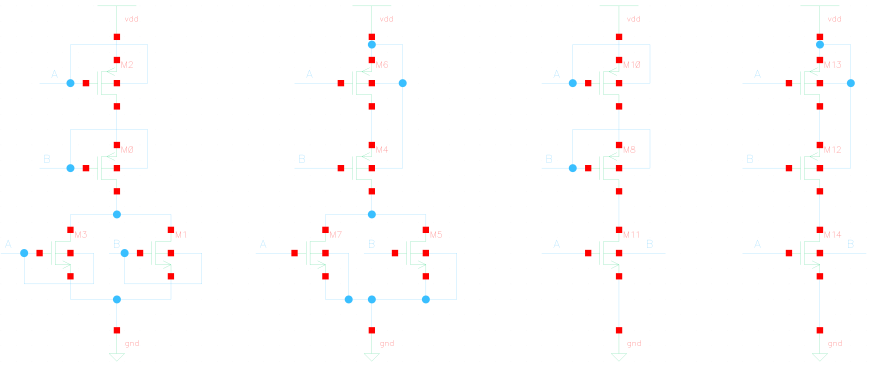

Figure 5: 4 Structures of NOR2 gate. From left to right, they are SG, LP, IG and IGLP.

The code of each mode is given below. LP mode:

```
************************
*  NOR2_1X_LP sample hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

*Define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 8
.param n_fin = 3
.param LoadCap = 1f

```

```
*Add transistors
mp1 Y A Vdd Vdd pfet L=lg NFIN=p_fin
mp2 Z B Y Vdd pfet L=lg NFIN=p_fin
mn1 Z A Gnd Gnd nfet L=lg NFIN=n_fin
mn2 Z B Gnd Gnd nfet L=lg NFIN=n_fin

*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VB B 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VA A 0 PULSE (0 'vdd' 1260ps 10ps 10ps 2ns 4ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 1p 20n

*Print waveform file *.tr0
.print V(Z)
.print V(A)
.print V(B)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout

*Measurement
.measure tran rise_delay trig V(B) val=0.35 TD=8.5n fall=1
+ targ V(Z) val=0.35 TD=8.5n rise=1
.measure tran fall_delay trig V(A) val=0.35 TD=8.5n rise=1
+ targ V(Z) val=0.35 TD=8.5n fall=1
.measure tran trise trig V(Z) val=0.14 TD=8.5n rise=1
+ targ V(Z) val=0.56 TD=8.5n rise=1
.measure tran tfall trig V(Z) val=0.14 TD=8.5n fall=1
+ targ V(Z) val=0.56 TD=8.5n fall=1
.measure tran avg_power AVG p(Cz)
.measure tran avg_power_mp1 AVG p(mp1)
.measure tran avg_power_mp2 AVG p(mp2)
.measure tran avg_power_mn1 AVG p(mn1)
.measure tran avg_power_mn2 AVG p(mn2)
.end
```

IG mode:

```
************************
*  NOR2_1X_IG sample hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

```

```
*Define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 8
.param n_fin = 6
.param LoadCap = 1f

*Add transistors
mp1 Y A Vdd A pfet L=lg NFIN=p_fin
mp2 Z B Y B pfet L=lg NFIN=p_fin
mn1 Z A Gnd B nfet L=lg NFIN=n_fin

*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VB B 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VA A 0 PULSE (0 'vdd' 1260ps 10ps 10ps 2ns 4ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 1p 20n

*Print waveform file *.tr0
.print V(Z)
.print V(A)
.print V(B)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout

*Measurement
.measure tran rise_delay trig V(B) val=0.35 TD=8.5n fall=1
+ targ V(Z) val=0.35 TD=8.5n rise=1
.measure tran fall_delay trig V(A) val=0.35 TD=8.5n rise=1
+ targ V(Z) val=0.35 TD=8.5n fall=1
.measure tran trise trig V(Z) val=0.14 TD=8.5n rise=1
+ targ V(Z) val=0.56 TD=8.5n rise=1
.measure tran tfall trig V(Z) val=0.14 TD=8.5n fall=1
+ targ V(Z) val=0.56 TD=8.5n fall=1
.measure tran avg_power AVG p(Cz)
.measure tran avg_power_mp1 AVG p(mp1)
.measure tran avg_power_mp2 AVG p(mp2)
.measure tran avg_power_mn1 AVG p(mn1)

.end
```

IGLP mode:

```
************************
*  NOR2_1X_IGLP sample hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

*Define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 8
.param n_fin = 6
.param LoadCap = 1f

*Add transistors
mp1 Y A Vdd Vdd pfet L=lg NFIN=p_fin
mp2 Z B Y Vdd pfet L=lg NFIN=p_fin
mn1 Z A Gnd B nfet L=lg NFIN=n_fin

*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VB B 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VA A 0 PULSE (0 'vdd' 1260ps 10ps 10ps 2ns 4ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 1p 20n

*Print waveform file *.tr0
.print V(Z)
.print V(A)
.print V(B)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout

*Measurement
.measure tran rise_delay trig V(B) val=0.35 TD=8.5n fall=1
+ targ V(Z) val=0.35 TD=8.5n rise=1
.measure tran fall_delay trig V(A) val=0.35 TD=8.5n rise=1
+ targ V(Z) val=0.35 TD=8.5n fall=1
.measure tran trise trig V(Z) val=0.14 TD=8.5n rise=1
+ targ V(Z) val=0.56 TD=8.5n rise=1
```

```
.measure tran tfall trig V(Z) val=0.14 TD=8.5n fall=1
+ targ V(Z) val=0.56 TD=8.5n fall=1
.measure tran avg_power AVG p(Cz)
.measure tran avg_power_mp1 AVG p(mp1)
.measure tran avg_power_mp2 AVG p(mp2)
.measure tran avg_power_mn1 AVG p(mn1)

.end
```

For IG and IGLP mode, since there is only one NMOS for the pull down part, the number of fins are compressed into one NMOS, then the number of fin of NMOS is 6 in these two modes. However, for IG and IGLP mode, because of the asymmetry of the structure, the pull down strength of gate and back is far different from each other, which lead to the error of functionality, which is shown in Figure 6. When the input signal A falls from '1' to '0' while B is still at '1', the output signal would also pull down, however, when B falls from '1' to '0' while A is still at '1', the output signal would just give a glitch instead of pull down the output signal, which means the pull-down strength of A is much stronger than B.

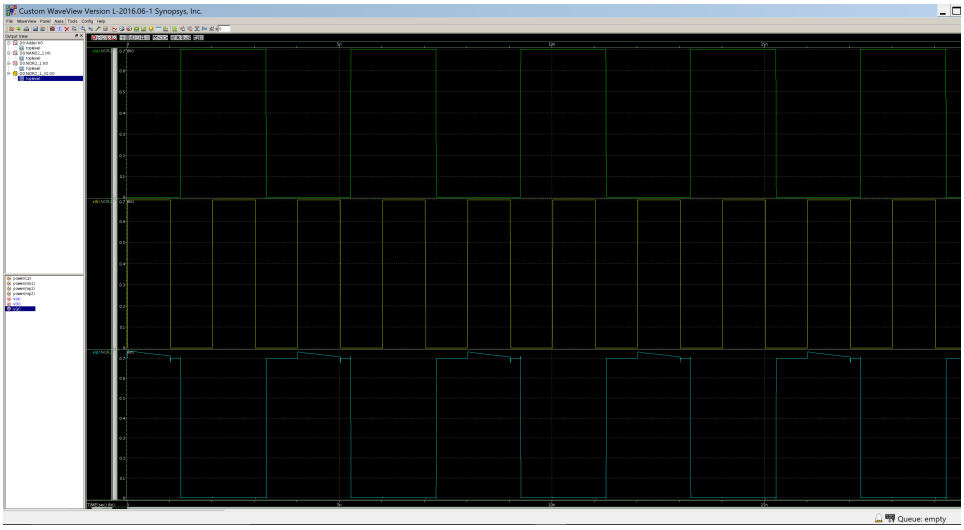

Figure 6: Error Functionality of IG Mode

Table 4: Simulation of SG, LP, IG, IGLP Mode 1xNOR2

| | Rise Delay | Fall Delay | Rise Time | Fall Time | Leakage Power | Switch Power | Switch Energy |
|---|---|---|---|---|---|---|---|
| SG | 3.282(ps) | 4.645(ps) | 3.447(ps) | 3.756(ps) | 305.39(nW) | 309.766(nW) | 1.239(fJ) |
| LP | 3.273(ps) | 4.644(ps) | 3.439(ps) | 3.745(ps) | 305.07(nW) | 309.87(nW) | 1.239(fJ) |
| IG | 2.253(ps) | 2.970(ps) | 2.753(ps) | 3.124(ps) | 177.700(nW) | 183.361(nW) | 0.837(fJ) |
| IGLP | 2.253(ps) | 2.970(ps) | 2.753(ps) | 3.123(ps) | 177.710(nW) | 183.327(nW) | 0.733(fJ) |

Table 4 shows the parameters of each mode. The timing parameters for IG and IGLP mode are based on input A.

## 2.5 Full Adder Simulation

### 2.5.1 Circuit Build

The schematic of a full adder is given in Figure 7. The Hspice is also given below.

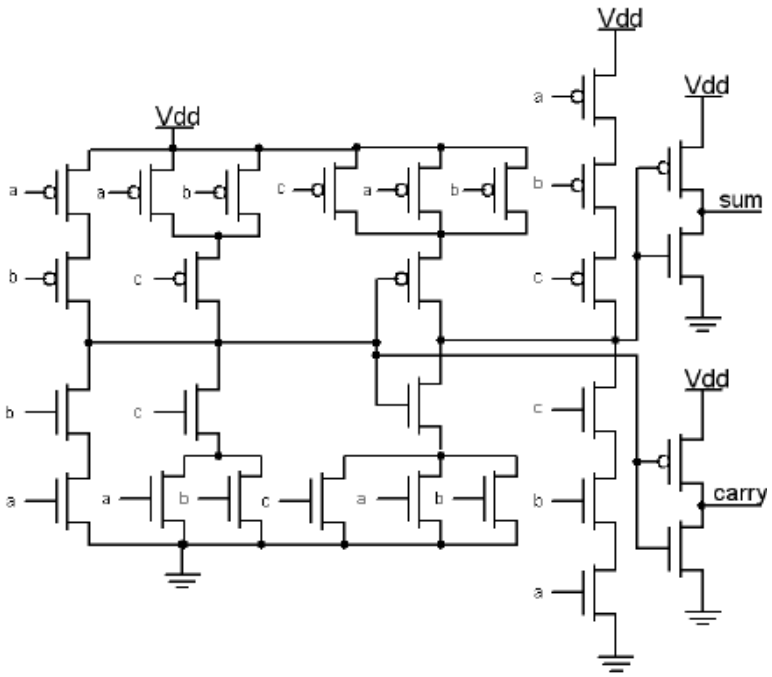


Figure 7: Schematic of Full Adder

```
************************
*  Full Adder hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

*Define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param p_fin = 4
.param n_fin = 3
.param LoadCap = 1f

*Add transistors
mp1 U1 A Vdd A pfet L=lg NFIN=p_fin
```

```
mp2 Cout_bar C U1 C pfet L=lg NFIN=p_fin
mp3 U1 B Vdd B pfet L=lg NFIN=p_fin
mp4 V1 A Vdd A pfet L=lg NFIN=p_fin
mp5 Cout_bar B V1 B pfet L=lg NFIN=p_fin
mp6 W1 A Vdd A pfet L=lg NFIN=p_fin
mp7 W1 B Vdd B pfet L=lg NFIN=p_fin
mp8 W1 C Vdd C pfet L=lg NFIN=p_fin
mp9 S_bar Cout_bar W1 Cout_bar pfet L=lg NFIN=p_fin
mp10 X1 A Vdd A pfet L=lg NFIN=p_fin
mp11 Y1 B X1 B pfet L=lg NFIN=p_fin
mp12 S_bar C Y1 C pfet L=lg NFIN=p_fin
mp13 Cout Cout_bar Vdd Cout_bar pfet L=lg NFIN=p_fin
mp14 S S_bar Vdd S_bar pfet L=lg NFIN=p_fin
mn1 U2 A Gnd A nfet L=lg NFIN=n_fin
mn2 Cout_bar C U2 C nfet L=lg NFIN=n_fin
mn3 U2 B Gnd B nfet L=lg NFIN=n_fin
mn4 V2 A Gnd A nfet L=lg NFIN=n_fin
mn5 Cout_bar B V2 B nfet L=lg NFIN=n_fin
mn6 W2 A Gnd A nfet L=lg NFIN=n_fin
mn7 W2 B Gnd B nfet L=lg NFIN=n_fin
mn8 W2 C Gnd C nfet L=lg NFIN=n_fin
mn9 S_bar Cout_bar W2 Cout_bar nfet L=lg NFIN=n_fin
mn10 X2 A Gnd A nfet L=lg NFIN=n_fin
mn11 Y2 B X2 B nfet L=lg NFIN=n_fin
mn12 S_bar C Y2 C nfet L=lg NFIN=n_fin
mn13 Cout Cout_bar Gnd Cout_bar nfet L=lg NFIN=n_fin
mn14 S S_bar Gnd S_bar nfet L=lg NFIN=n_fin

*Add cap
Cz1 S Gnd 'LoadCap'
Cz2 Cout Gnd 'LoadCap'

*Add voltage sourse
VA A 0 PULSE (0 'vdd' 10ps 10ps 10ps 1ns 2ns)
VB B 0 PULSE (0 'vdd' 10ps 10ps 10ps 2ns 4ns)
VC C 0 PULSE (0 'vdd' 10ps 10ps 10ps 4ns 8ns)
VDD Vdd 0 'vdd'
VSS Gnd 0 'vss'

*Do transient analysis
.tran 10p 12n

*Print waveform file *.tr0
.print V(A)
.print V(B)
.print V(C)
.print V(Cout)
.print V(S)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
```

```
.OPTION Post Brief NoMod probe measout

*Measurement Cout Delay
.measure tran Cout_tdlay1 trig V(A) val=0.35 TD=2.5n fall=1
+ targ V(Cout) val=0.35 fall=1
.measure tran Cout_tdlay2 trig V(A) val=0.35 TD=3.5n rise=1
+ targ V(Cout) val=0.35 rise=1
.measure tran Cout_trise trig V(Cout) val=0.14 TD=3.5n rise=1
+ targ V(Cout) val=0.56 rise=1
.measure tran Cout_tfall trig V(Cout) val=0.14 TD=2.5n fall=1
+ targ V(Cout) val=0.56 TD=2.5n fall=1

*Measurement S Delay
.measure tran S_tdlay1 trig V(A) val=0.35 TD=0.5n fall=1
+ targ V(S) val=0.35 fall=1
.measure tran S_tdlay2 trig V(A) val=0.35 TD=2.5n fall=1
+ targ V(S) val=0.35 rise=1
.measure tran S_trise trig V(S) val=0.14 TD=2.5n rise=1
+ targ V(S) val=0.56 rise=1
.measure tran S_tfall trig V(S) val=0.14 TD=0.5n fall=1
+ targ V(S) val=0.56 TD=0.5n fall=1

.measure tran avg_power AVG p(Cz1)
.measure tran avg_power AVG p(Cz2)

.end
```

### 2.5.2 Simulation Result

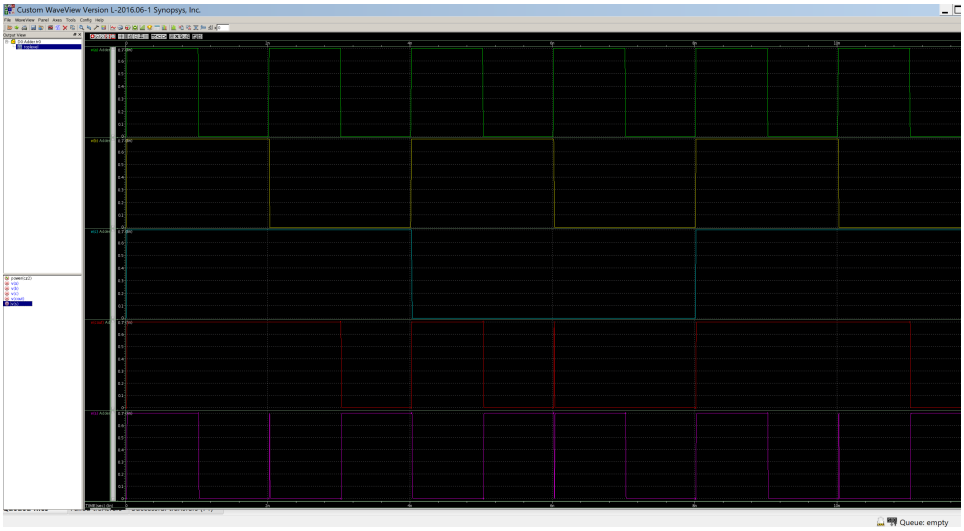

Figure 8: Functionality Simulation of Full Adder

Figure 8 shows the functionality simulation of the full adder. The rise and fall delay of $C_{out}$ is 7.204ps and 4.551ps. For $S$, they are 6.653ps and 9.554ps.

# 3 PTM FinFET DRAM

## 3.1 Single Test Case

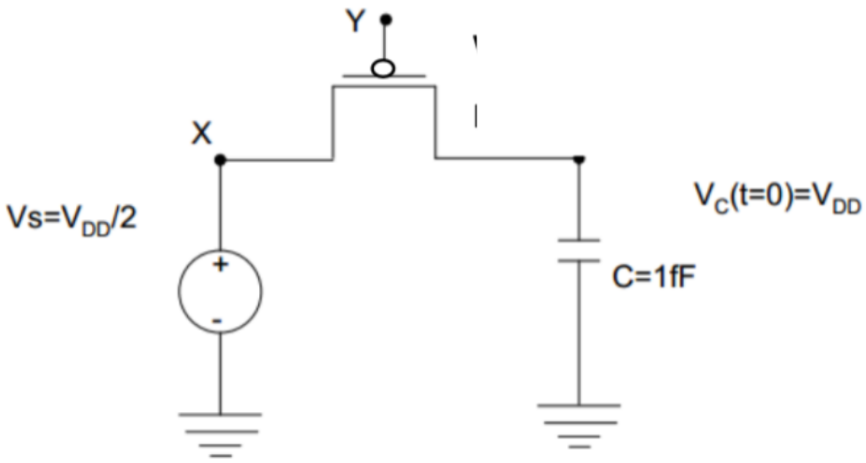


Figure 9: 1-Bit DRAM Setup

A basic 1-bit DRAM circuit setup is shown in Figure 9. The Y node is connected to the gate of the PMOS, which is set to be '0'. We measure the discharge time, average current and average power to analyze this circuit.

The Hspice code is given below.

```
************************
*  DRAM hspice
************************

.include './hp7nfet.pm'
.include './hp7pfet.pm'

*Define parameters
.param vdd=0.7
.param vss=0
.param fin_height=18n
.param fin_width=7n
.param lg=11n
.param number_fin = 1
.param step = 0.01p
```

```
.param length = 1n
.param LoadCap = 1f

*Add transistors
mp1 X Y Z Gnd pfet L=lg NFIN=number_fin


*Add cap
Cz Z Gnd 'LoadCap'

*Add voltage sourse
VX X 0 'vdd/2'
VY Y 0 'vss'
VSS Gnd 0 'vss'

*Define the initial condition of V(Z)
.IC V(Z)='vdd'

*Do transient analysis
.tran 'step' 'length'

*Print the V(Z) to waveform file *.tr0
.print V(Z)

*Simulation options (you can modify this. Post is needed for .tran
    analysis)
.OPTION Post Brief NoMod probe measout

*Measurement
.measure tran RTL TRIG AT=0 TARG v(Z) VAL=0.35 FALL=1
.measure tran avg_current AVG I(Cz) from 0 to 'RTL'
.measure tran avg_power AVG p(Cz)

.end
```

## 3.2 Python Code

The Python code contains two parts. First part is to read from the sample Hspice file and generate a new one whose parameters, say load capacitance, number of fin, time step and time length has been changed. For this part, the Python code is shown below. For simplicity, the number of fins follows the value in 1xINV template.

```
import re
import sys
filein = open('sample.sp','r')
fileout = open('new.sp','w')
stepfind = '.param step = ' + sys.argv[1]
stepsub = '.param step = '+ sys.argv[2]
lengthfind = '.param length = ' + sys.argv[3]
```

```
lengthsub = '.param length = ' + sys.argv[4]
Cffind = '.param LoadCap = ' + sys.argv[5]
Cfsub = '.param LoadCap = '+ sys.argv[6]
finfind = '.param number_fin = ' + sys.argv[7]
finsub = '.param number_fin = ' + sys.argv[8]

for i in filein:
  if re.search(stepfind, i) != None:
    u = re.sub(stepfind, stepsub, i)
    fileout.write(u)
  elif re.search(lengthfind, i) != None:
    v = re.sub(lengthfind, lengthsub, i)
    fileout.write(v)
  elif re.search(Cffind, i) != None:
    w = re.sub(Cffind, Cfsub, i)
    fileout.write(w)
  elif re.search(finfind, i) != None:
    x = re.sub(finfind, finsub, i)
    fileout.write(x)
  else:
    fileout.write(i)
filein.close()
fileout.close()
```

For second part, which is the operating part, it provides the value of each case and apply a large for loop to do the iteration. And I used the "os" operation to run Hspice simulation in the command line. After each simulation, data will also be collected automatically. This part of code is given below.

```
import re
import os
import sys

def write_array(files,array):
  for i in array:
    files.write(i+'\t')
  files.write('\n')

def write_data(filesin,filesout):
  count = 0
  for i in filesin:
    ss = i.split()
    if len(ss) == 4:
      count = count + 1
      if count != 1:
        write_array(filesout,ss[0:-1])

fileout=open('dram.txt','w')
step = ['0.01p','0.01p','0.01p','2p','2p','2p','2p','0.5n','0.5n',
    '0.5n','0.5n']
```

```
tlength = ['1n','1n','1n','150n','150n','150n','150n','8u','8u','8
    u','8u']
Cf = ['1f','2.951f','8.705f','25.69f','75.79f','223.6f','659.8f','
    1947f','5743f','16950f','50000f']
fin = ['1','2','3','4','5']
for i in range(len(fin)):
  for j in range(len(tlength)):
    if i == 0 and j == 0:
      cmd = 'python sub.py'+' '+step[j]+' '+step[j]+' '+tlength[j
    ]+' '+tlength[j]+' '+Cf[j]+' '+Cf[j]+' '+fin[j]+' '+fin[j]
    else:
      cmd = 'python sub.py'+' '+step[0]+' '+step[j]+' '+tlength
    [0]+' '+tlength[j]+' '+Cf[0]+' '+Cf[j]+' '+fin[0]+' '+fin[i]
    os.system(cmd)
    os.system('hspice new.sp > new.lis')
    filein = open('new.mt0','r')
    write_data(filein,fileout)
  fileout.write('\n')
```

### 3.3 Simulation Result

The simulation result is given below. The first column is the RTL time, second column is the average current and third column is the average power. They are negative because the load capacitor is discharging.

```
7.397e-11  -4.442e-06   -1.707e-07
1.737e-10  -5.791e-06   -5.348e-07
7.812e-10  -3.863e-06   -1.605e-06
1.588e-09  -5.642e-06   -3.188e-08
5.232e-09  -5.065e-06   -9.410e-08
1.376e-08  -5.685e-06   -2.764e-07
4.621e-08  -4.997e-06   -8.135e-07
1.545e-07  -4.386e-06   -4.511e-08
4.181e-07  -4.798e-06   -1.337e-07
1.183e-06  -5.010e-06   -3.945e-07
3.911e-06  -4.473e-06   -1.162e-06

4.112e-11  -7.179e-06   -1.507e-07
7.344e-11  -1.341e-05   -5.161e-07
3.450e-10  -8.668e-06   -1.581e-06
1.051e-09  -8.498e-06   -3.171e-08
2.561e-09  -1.033e-05   -9.409e-08
6.766e-09  -1.156e-05   -2.776e-07
2.082e-08  -1.109e-05   -8.154e-07
7.106e-08  -9.487e-06   -4.495e-08
1.991e-07  -1.006e-05   -1.335e-07
5.054e-07  -1.172e-05   -3.950e-07
1.749e-06  -9.999e-06   -1.162e-06

1.709e-11  -1.722e-05   -1.469e-07
```

```
7.316e-11  -1.324e-05  -5.031e-07
1.711e-10  -1.734e-05  -1.576e-06
4.357e-10  -2.044e-05  -3.166e-08
1.572e-09  -1.682e-05  -9.403e-08
5.214e-09  -1.499e-05  -2.776e-07
1.367e-08  -1.689e-05  -8.155e-07
3.077e-08  -2.180e-05  -4.505e-08
1.530e-07  -1.307e-05  -1.330e-07
4.045e-07  -1.464e-05  -3.947e-07
1.184e-06  -1.477e-05  -1.164e-06

1.959e-11  -1.394e-05  -1.354e-07
6.101e-11  -1.567e-05  -4.928e-07
1.753e-10  -1.679e-05  -1.553e-06
5.077e-10  -1.748e-05  -3.156e-08
1.320e-09  -2.001e-05  -9.380e-08
3.927e-09  -1.990e-05  -2.774e-07
9.706e-09  -2.378e-05  -8.163e-07
3.570e-08  -1.870e-05  -4.492e-08
6.666e-08  -2.993e-05  -1.329e-07
3.594e-07  -1.647e-05  -3.951e-07
7.091e-07  -2.466e-05  -1.167e-06

1.620e-11  -1.653e-05  -1.297e-07
4.440e-11  -2.074e-05  -4.699e-07
8.423e-11  -3.470e-05  -1.536e-06
2.333e-10  -3.792e-05  -3.157e-08
1.143e-09  -2.308e-05  -9.364e-08
2.717e-09  -2.875e-05  -2.774e-07
9.088e-09  -2.539e-05  -8.176e-07
1.944e-08  -3.418e-05  -4.425e-08
5.711e-08  -3.487e-05  -1.328e-07
1.674e-07  -3.533e-05  -3.946e-07
6.597e-07  -2.650e-05  -1.166e-06
```

## 3.4 MATLAB Analysis

### 3.4.1 3-D Plot

```
clear all
Cf =
    [1,2.951,8.705,25.69,75.79,223.6,659.8,1947,5743,16950,50000];
fin = [1 2 3 4 5];

archive = fopen('dram.txt');
b = fscanf(archive,'%f',[3,60]);
b=b';
for i = 1:11
    rtl(1,i) = b(i,1);
    current(1,i) = b(i,2);
```

```
    power(1,i) = b(i,3);
end
for i = 12:22
    rtl(2,i-11) = b(i,1);
    current(2,i-11) = b(i,2);
    power(2,i-11) = b(i,3);
end
for i = 23:33
    rtl(3,i-22) = b(i,1);
    current(3,i-22) = b(i,2);
    power(3,i-22) = b(i,3);
end
for i = 34:44
    rtl(4,i-33) = b(i,1);
    current(4,i-33) = b(i,2);
    power(4,i-33) = b(i,3);
end
for i = 45:55
    rtl(5,i-44) = b(i,1);
    current(5,i-44) = b(i,2);
    power(5,i-44) = b(i,3);
end
h=gca;

surf(log(Cf),fin,log(rtl))
title('R.T.L')
xlabel('Cf in log scale')
ylabel('Num of Fin')
figure;
surf(log(Cf),fin,log(abs(current)))
title('Current')
xlabel('Cf in log scale')
ylabel('Num of Fin')
figure;
surf(log(Cf),fin,log(abs(power)))
title('Power')
xlabel('Cf in log scale')
ylabel('Num of Fin')
```

The MATLAB code for 3-D plot is given above.

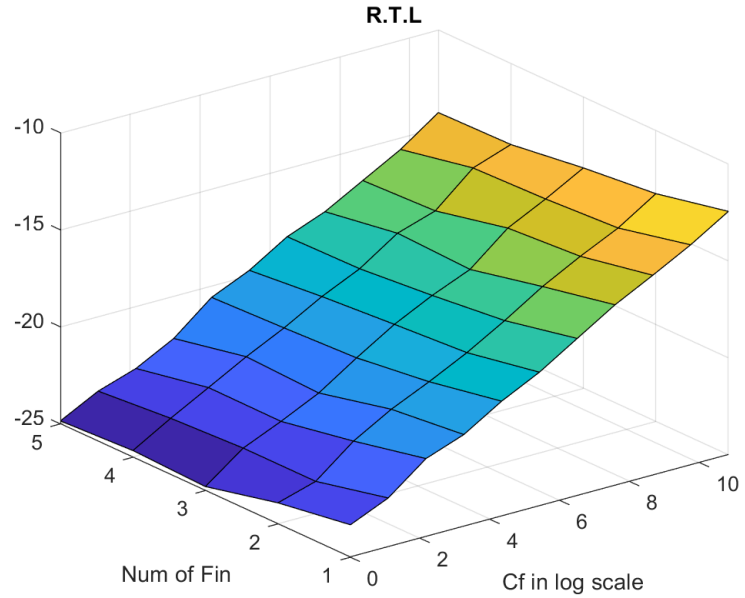


Figure 10: 3-D Plot of R.T.L

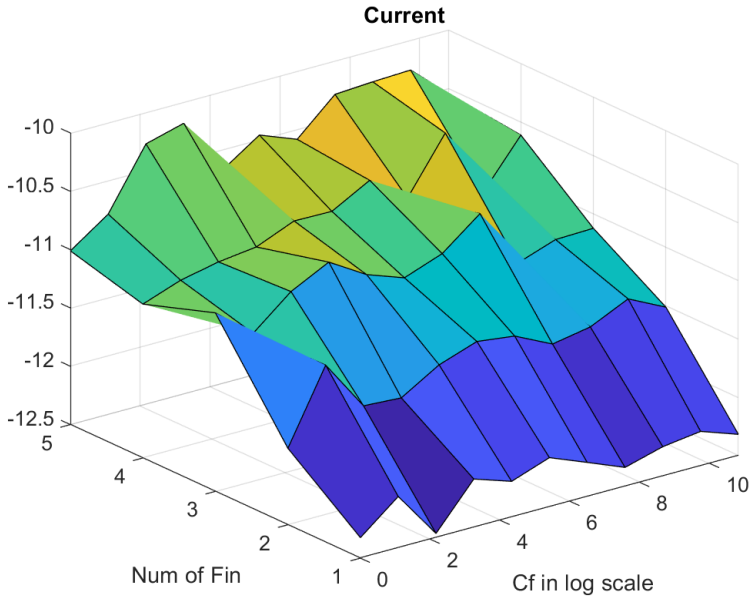


Figure 11: 3-D Plot of Average Current

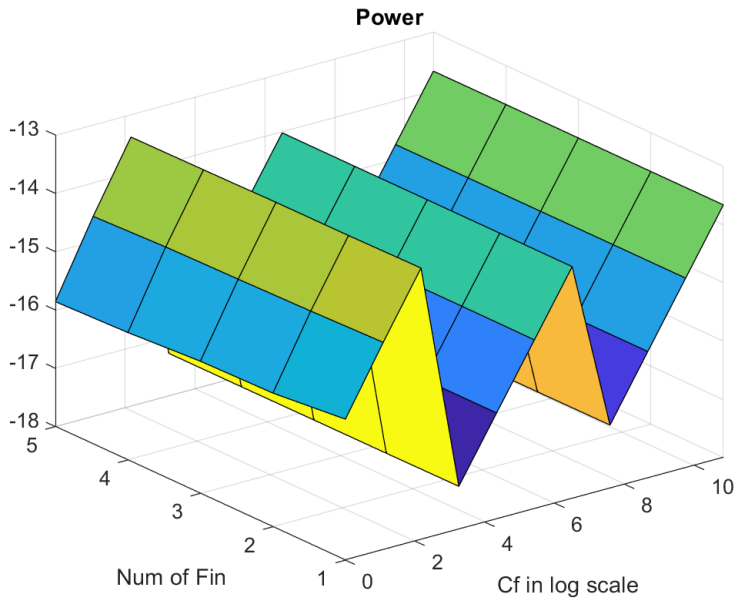


Figure 12: 3-D Plot of Average Power

### 3.4.2 Evaluation

The MATLAB code for calculating statistic values are given below.

```
clear all
Cf =
    [1,2.951,8.705,25.69,75.79,223.6,659.8,1947,5743,16950,50000];
fin = [1 2 3 4 5];

archive = fopen('dram.txt');
b = fscanf(archive,'%f',[3,60]);
b=b';
for i = 1:55
    rtl(1,i) = b(i,1);
    current(1,i) = b(i,2);
    power(1,i) = b(i,3);
end
RTL_min = min(rtl);
RTL_max = max(rtl);
RTL_avg = mean(rtl);
RTL_std = std(rtl);
current_min = min(current);
current_max = max(current);
current_avg = mean(current);
current_std = std(current);
power_min = min(power);
power_max = max(power);
```

```
power_avg = mean(power);
power_std = std(power);
```

The minimum value, maximum value, average value and standard deviation is given in the table below.

Table 5: Statistic Values of R.T.L, Average Current and Average Power

| | **Minimum** | **Maximum** | **Average** | **Standard Deviation** |
|---|---|---|---|---|
| R.T.L | 16.20(ps) | 3.91($\mu$s) | 221.59(ns) | 612.38(ns) |
| Average Current | 3.86($\mu$A) | 37.92($\mu$A) | 16.05($\mu$A) | 9.16($\mu$A) |
| Average Power | 31.56(nW) | 1.61($\mu$W) | 470.53(nW) | 489.64(nW) |

# References