

Lab Report 2: Cadence Custom Layout: Design Rules, Extraction, and Verification

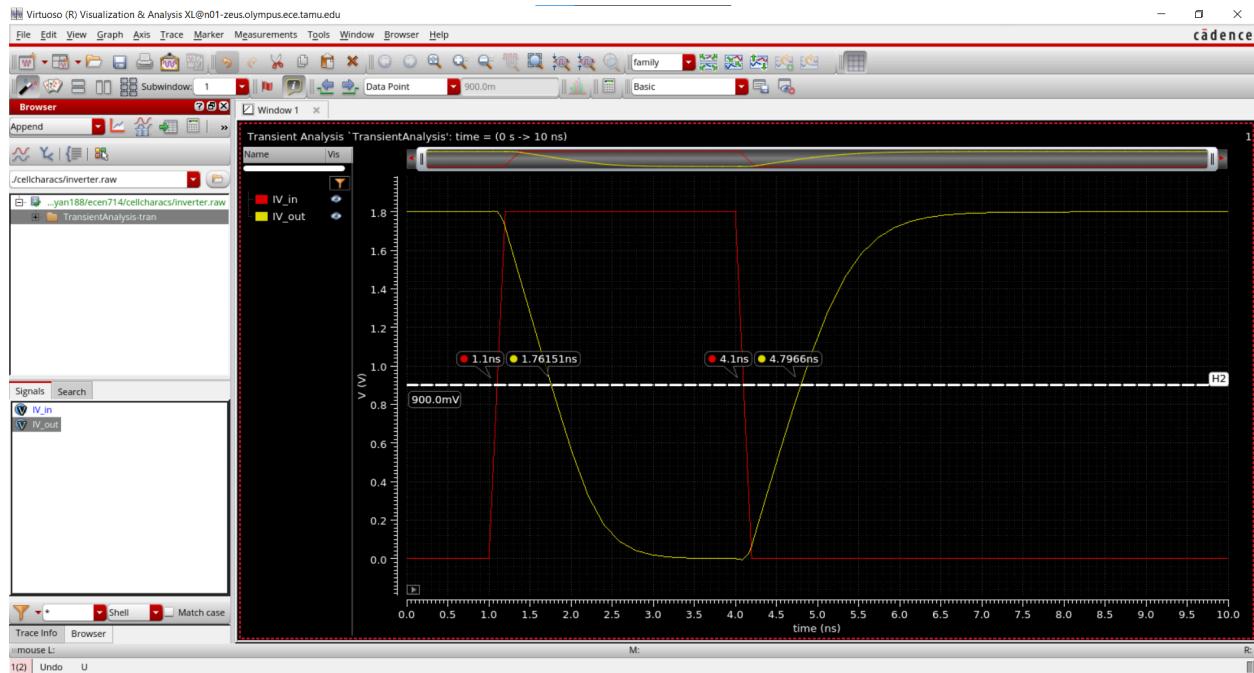
ECEN 454/714

1/29/2024

TA: Saichand Samudrala

Ryan Peng
631003156

Inverter waveforms



$$\text{Falling delay} = (1.76151 - 1.1) = 0.66151$$

$$\text{Rising delay} = (4.7966 - 4.1) = 0.697$$

$$\text{Error percentage} = \frac{\text{abs(rising - falling)}}{\min(\text{rising}, \text{falling})} * 100 = \frac{0.03549}{0.66151} * 100 = 5.365\%$$

Inverter code

```
GNU nano 2.3.1          File: inverter.spi          Modified ^

//Ryan Peng 631003156
;Spice netlist for an inverter and a capacitor
simulator lang=spice

include "~/ecen714/cellcharacs/model18.spi"
include "~/ecen714/cellcharacs/cell18.spi"

vgnd (gnd 0) vsource dc=0
vdd (vdd 0) vsource dc=1.8

vpwl (IV_in 0) vsource type=pwl wave=[0n 0 1n 1.2n 1.8 4n 1.8 4.2n 0]

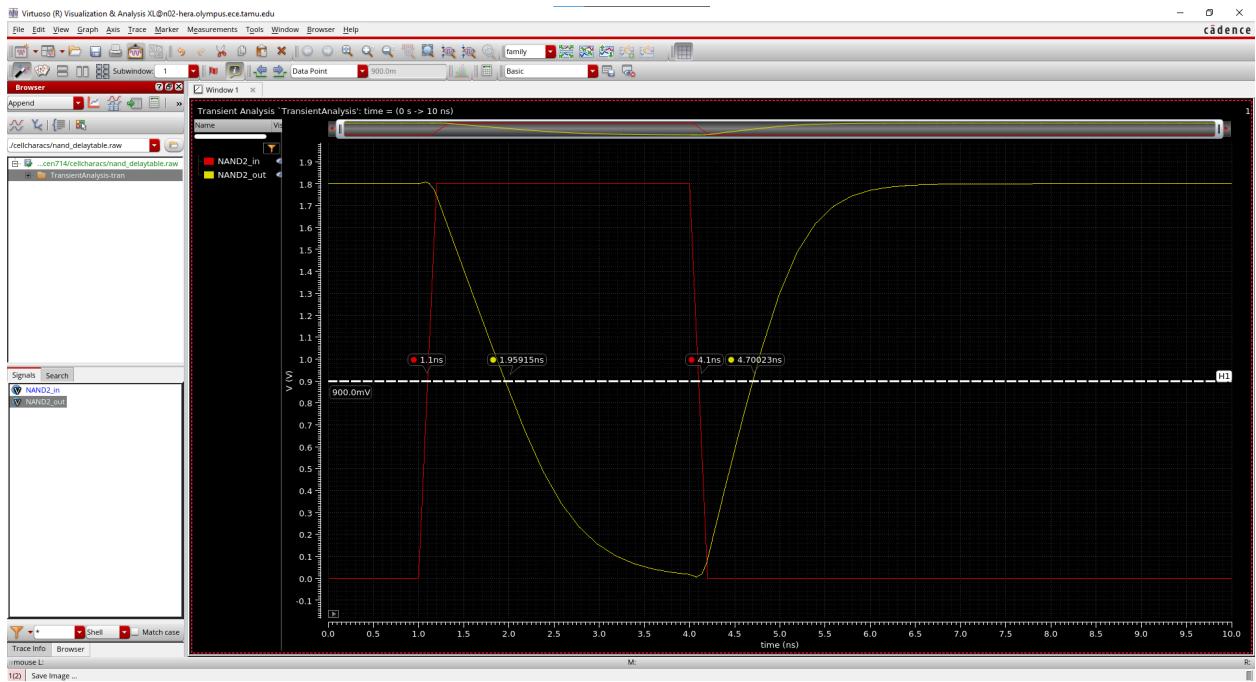
X1 (IV_in IV_out vdd gnd) IV wp=0.6u lp=0.2u wn=0.3u ln=0.3u

R1 (IV_out 1) resistor r=1
C1 (1 0) capacitor c=100f

TransientAnalysis tran start=0 stop=10ns step=1ps
save IV_in IV_out

[ Read 23 lines (Converted from DOS format) ]
```

C=100f

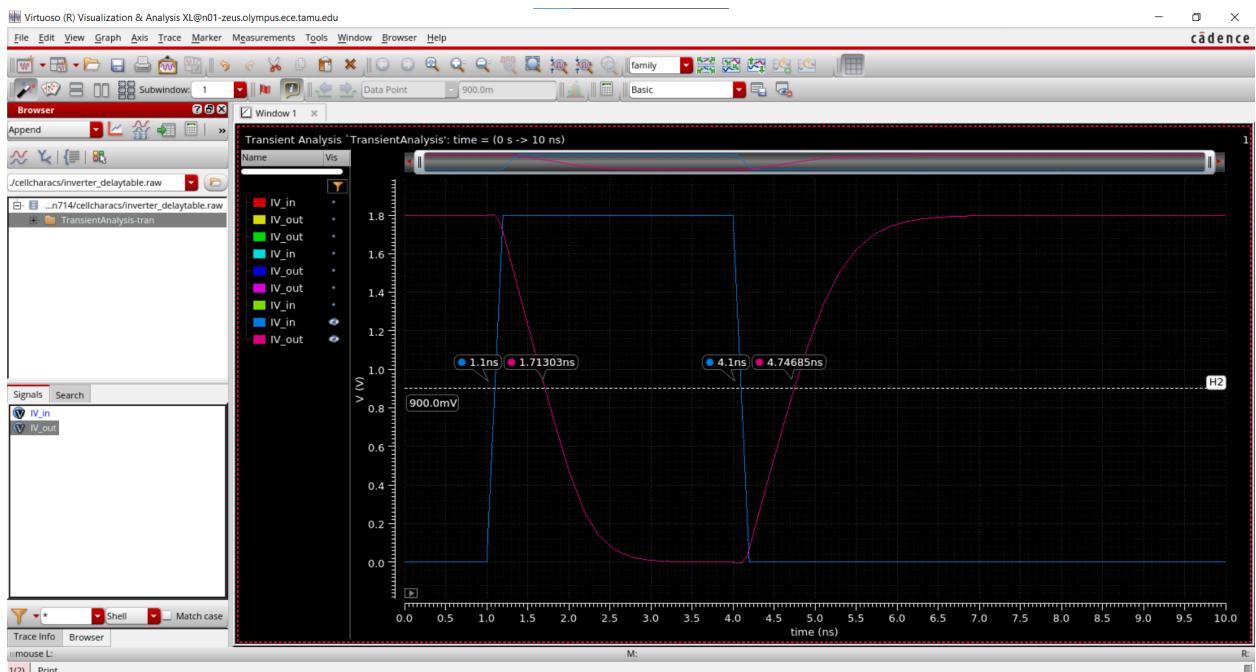


$$\text{Rising delay} = (1.95915 - 1.1) = 0.859$$

$$\text{Falling delay} = (4.70023 - 4.1) = 0.6$$

$$\text{Error \%} = \text{abs}(\text{rising} - \text{falling}) / \min(\text{rising}, \text{falling}) * 100 = (0.697 - 0.662) / 0.662 * 100 = 5.287\%$$

C=92

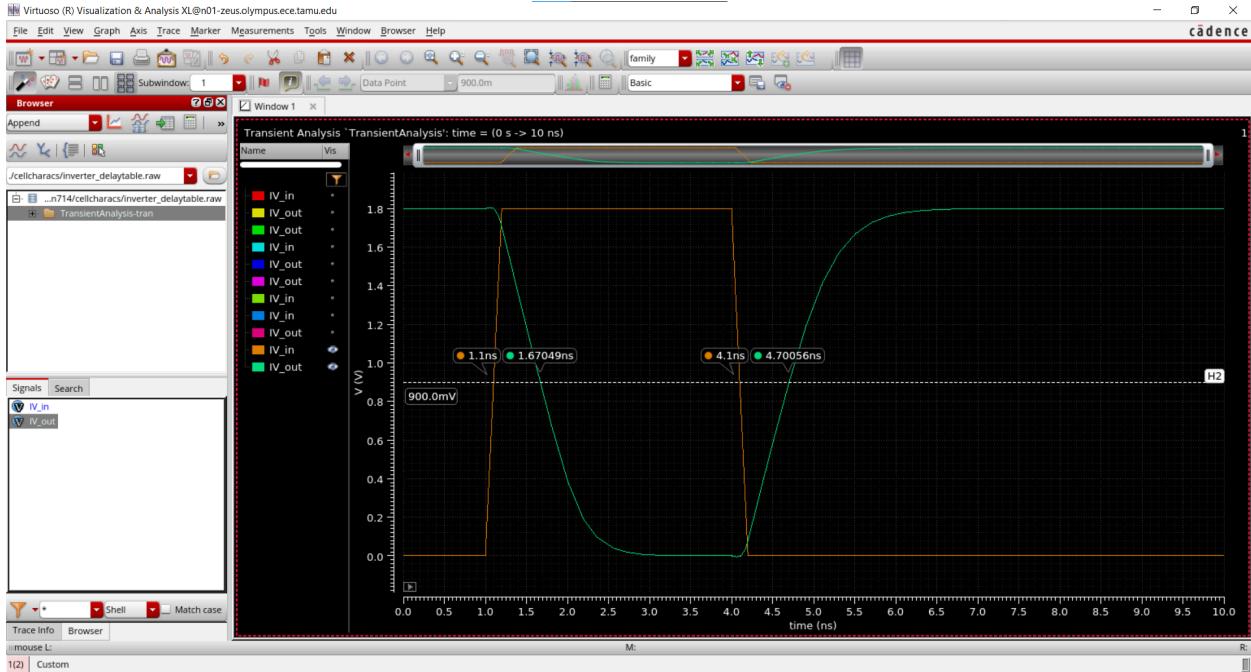


$$\text{Rising delay} = (1.71303 - 1.1) = 0.613$$

$$\text{Falling delay} = (4.74685 - 4.1) = 0.647$$

$$\text{Error \%} = \text{abs}(\text{rising} - \text{falling}) / \min(\text{rising}, \text{falling}) * 100 = (0.647 - 0.613) / 0.613 * 100 = 5.546\%$$

C=85

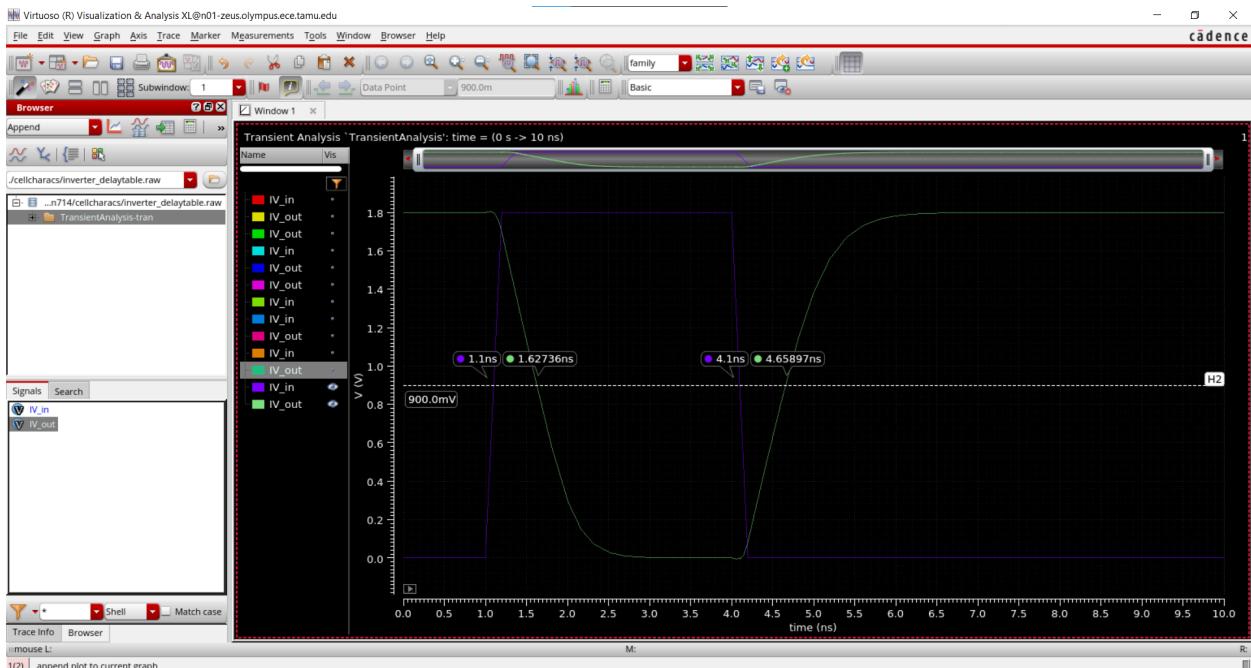


$$\text{Rising delay} = (1.67049 - 1.1) = 0.57$$

$$\text{Falling delay} = (4.70056 - 4.1) = 0.601$$

$$\text{Error \%} = \text{abs}(\text{rising} - \text{falling}) / \text{min}(\text{rising}, \text{falling}) * 100 = (0.601 - 0.57) / 0.57 * 100 = 5.439\%$$

C=78

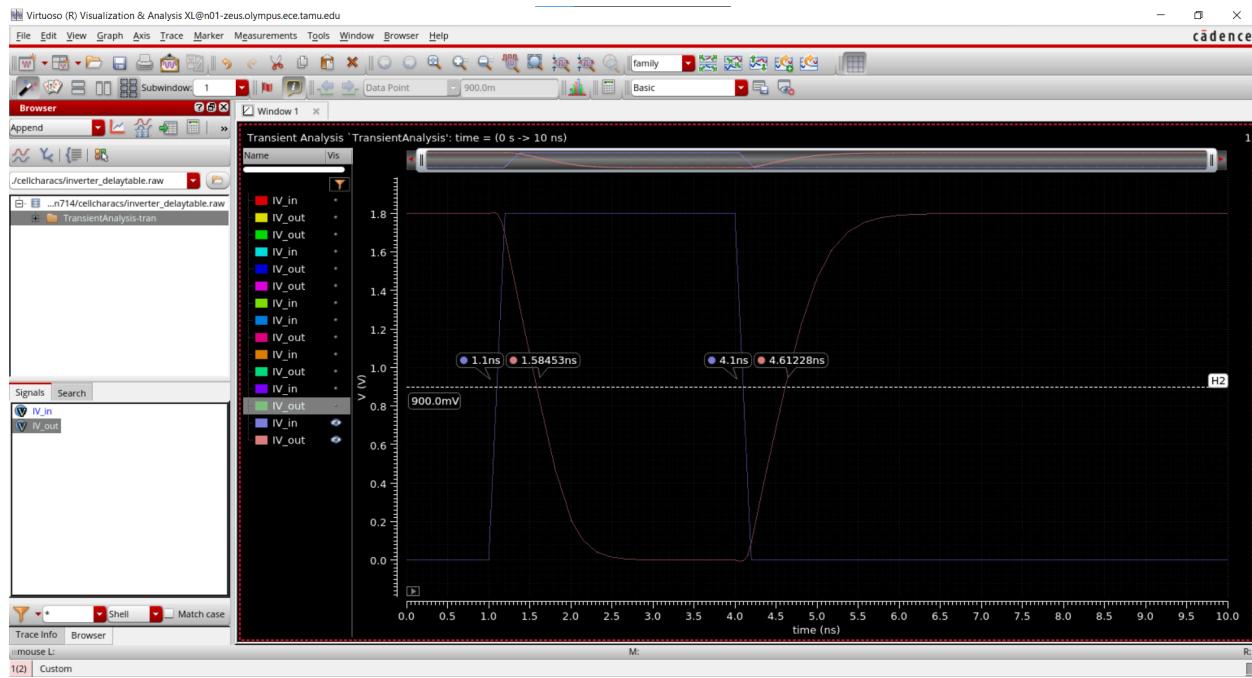


$$\text{Rising delay} = (1.62736 - 1.1) = 0.528$$

$$\text{Falling delay} = (4.65897 - 4.1) = 0.559$$

$$\text{Error \%} = \text{abs}(\text{rising} - \text{falling}) / \text{min}(\text{rising}, \text{falling}) * 100 = (0.559 - 0.528) / 0.528 * 100 = 5.871\%$$

C=71

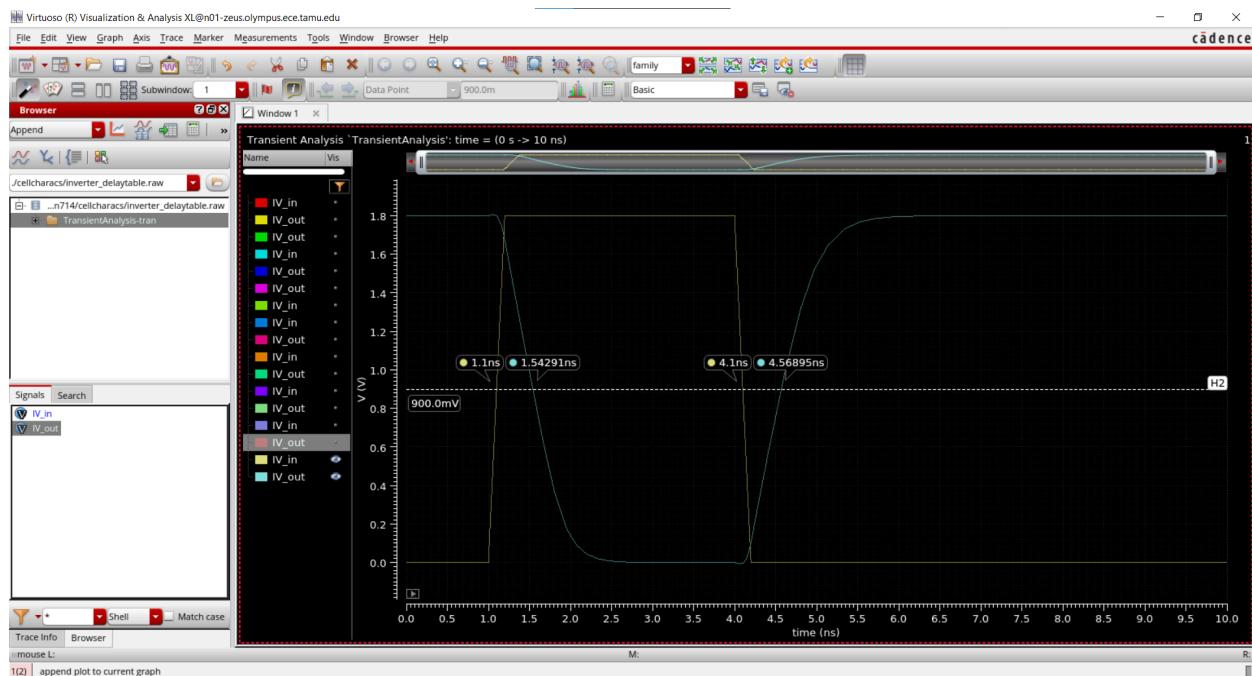


$$\text{Rising delay} = (1.58453 - 1.1) = 0.485$$

$$\text{Falling delay} = (4.61228 - 4.1) = 0.512$$

$$\text{Error \%} = \frac{\text{abs(rising - falling)}}{\min(\text{rising}, \text{falling})} * 100 = \frac{(0.512 - 0.485)}{0.485} * 100 = 5.567\%$$

C=64

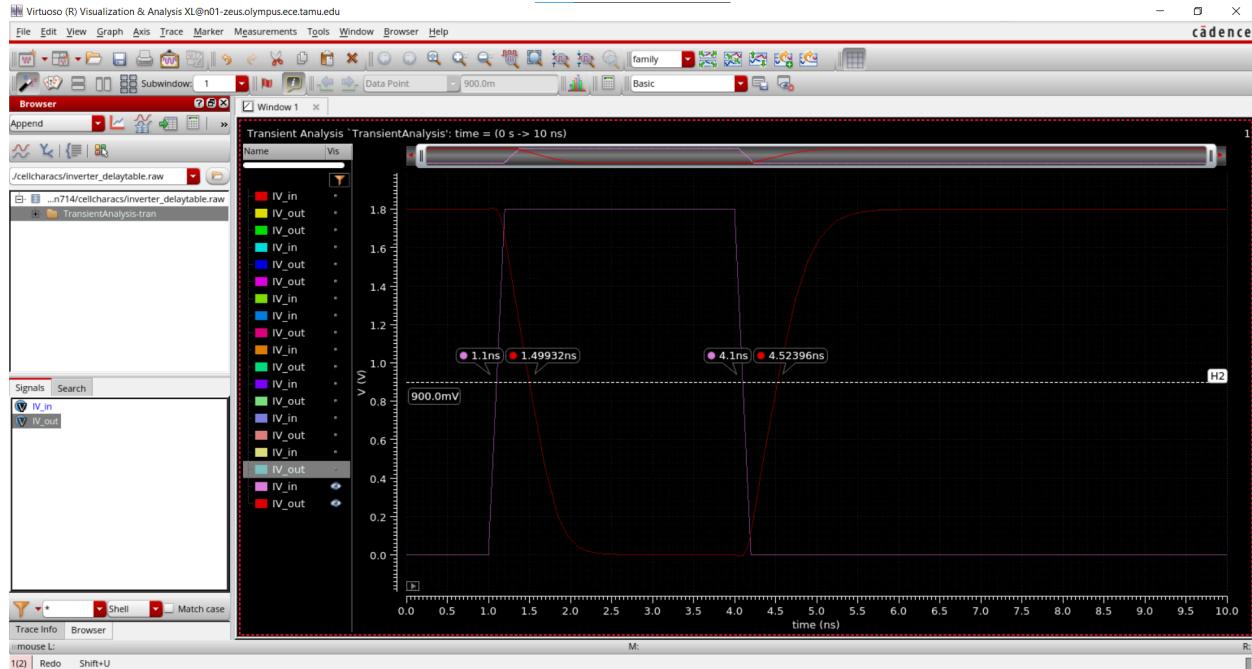


$$\text{Rising delay} = (1.54291 - 1.1) = 0.443$$

$$\text{Falling delay} = (4.56895 - 4.1) = 0.469$$

$$\text{Error \%} = \frac{\text{abs(rising - falling)}}{\min(\text{rising}, \text{falling})} * 100 = \frac{(0.469 - 0.443)}{0.443} * 100 = 5.869\%$$

C=57



$$\text{Rising delay} = (1.49932 - 1.1) = 0.399$$

$$\text{Falling delay} = (4.52396 - 4.1) = 0.424$$

$$\text{Error \%} = \frac{\text{abs(rising - falling)}}{\min(\text{rising}, \text{falling})} * 100 = \frac{(0.424 - 0.399)}{0.399} * 100 = 6.266\%$$

C=50

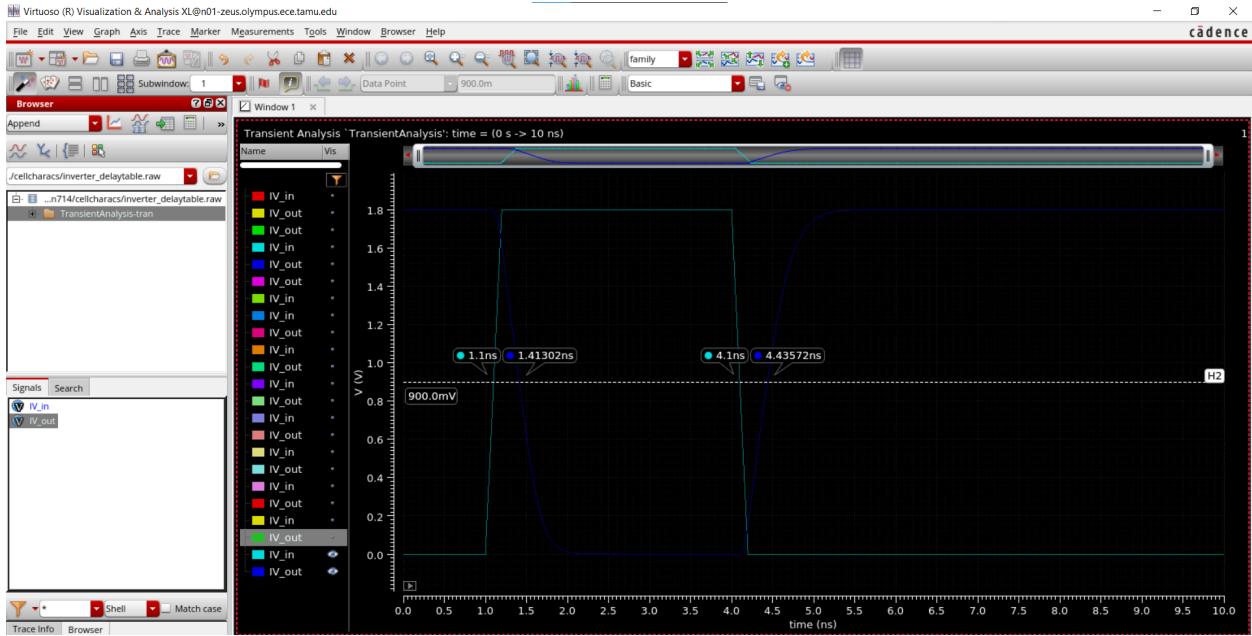


$$\text{Rising delay} = (1.45841 - 1.1) = 0.358$$

$$\text{Falling delay} = (4.48169 - 4.1) = 0.382$$

$$\text{Error \%} = \frac{\text{abs(rising - falling)}}{\min(\text{rising}, \text{falling})} * 100 = \frac{(0.382 - 0.358)}{0.358} * 100 = 6.704\%$$

C=43

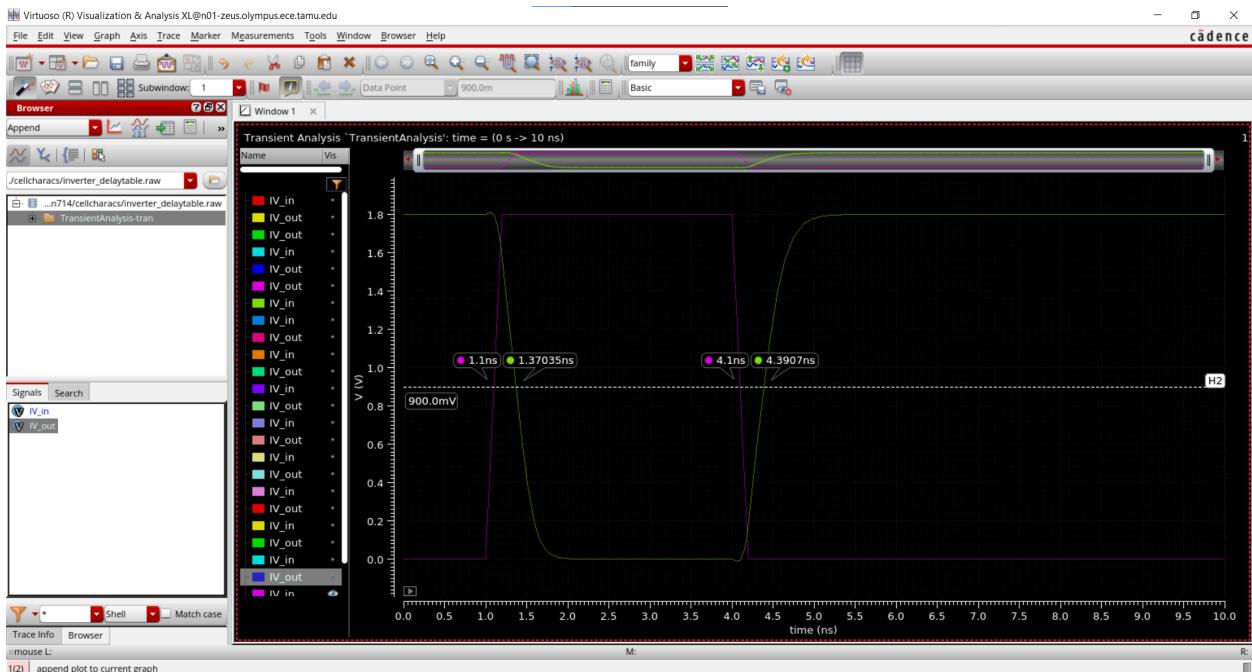


$$\text{Rising delay} = (1.41302 - 1.1) = 0.313$$

$$\text{Falling delay} = (4.43572 - 4.1) = 0.336$$

$$\text{Error \%} = \frac{\text{abs(rising - falling)}}{\min(\text{rising}, \text{falling})} * 100 = (0.336 - 0.313) / 0.313 * 100 = 7.348\%$$

C=36

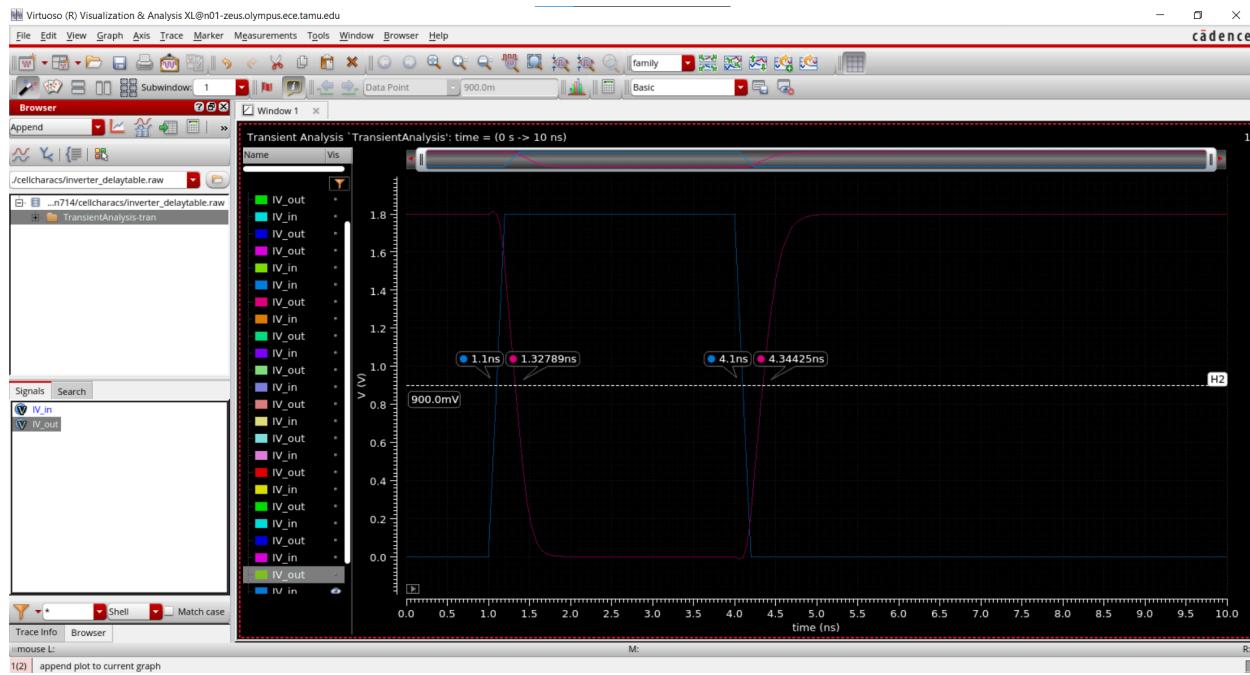


$$\text{Rising delay} = (1.37035 - 1.1) = 0.27$$

$$\text{Falling delay} = (4.3907 - 4.1) = 0.291$$

$$\text{Error \%} = \frac{\text{abs(rising - falling)}}{\min(\text{rising}, \text{falling})} * 100 = (0.291 - 0.27) / 0.27 * 100 = 7.778\%$$

C=29

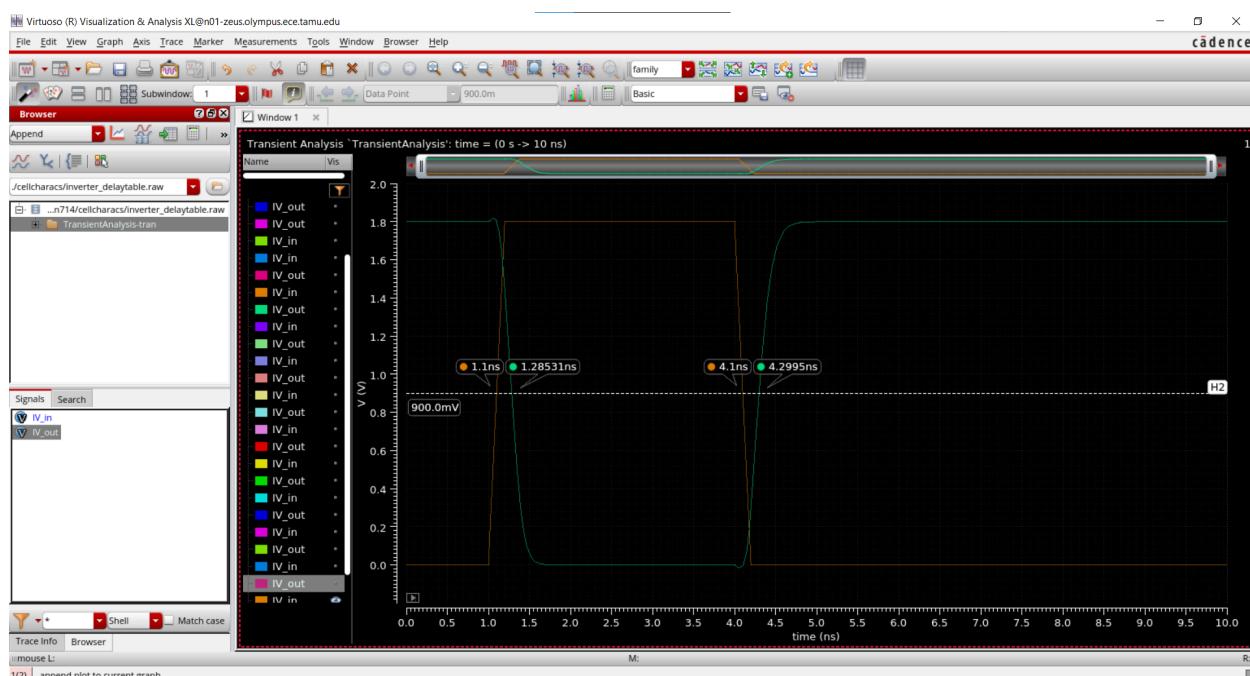


$$\text{Rising delay} = (1.32789 - 1.1) = 0.228$$

$$\text{Falling delay} = (4.34425 - 4.1) = 0.244$$

$$\text{Error \%} = \frac{\text{abs(rising - falling)}}{\min(\text{rising}, \text{falling})} * 100 = (0.244 - 0.228) / 0.228 * 100 = 7.018\%$$

C=22

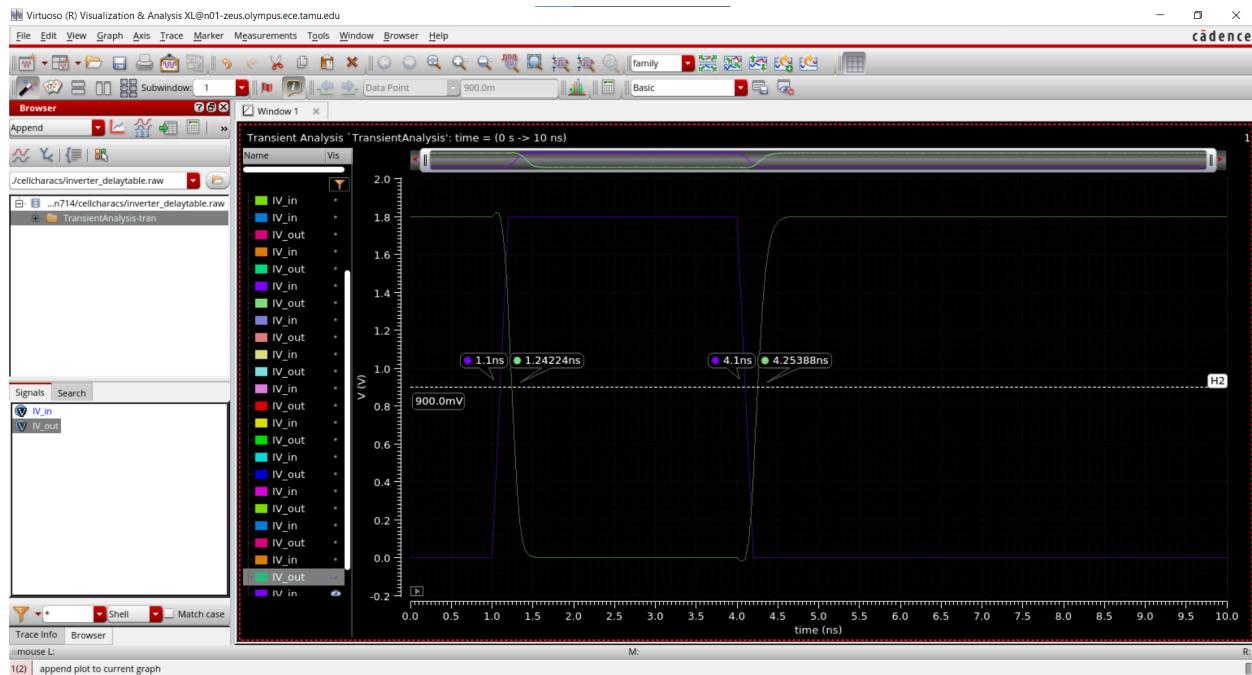


$$\text{Rising delay} = (1.28531 - 1.1) = 0.185$$

$$\text{Falling delay} = (4.2995 - 4.1) = 0.2$$

$$\text{Error \%} = \frac{\text{abs(rising - falling)}}{\min(\text{rising}, \text{falling})} * 100 = (0.2 - 0.185) / 0.185 * 100 = 8.108\%$$

C=15

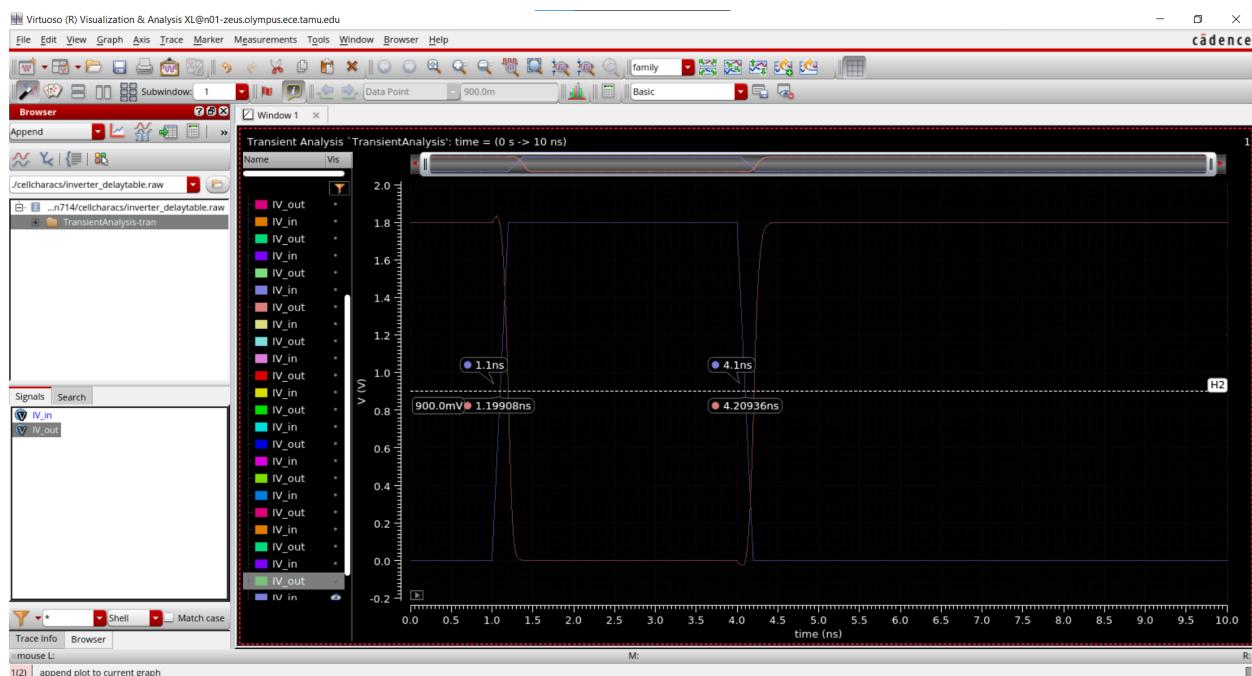


$$\text{Rising delay} = (1.24224 - 1.1) = 0.142$$

$$\text{Falling delay} = (4.25388 - 4.1) = 0.154$$

$$\text{Error \%} = \frac{\text{abs(rising - falling)}}{\min(\text{rising}, \text{falling})} * 100 = \frac{(0.154 - 0.142)}{0.142} * 100 = 8.451\%$$

C=8

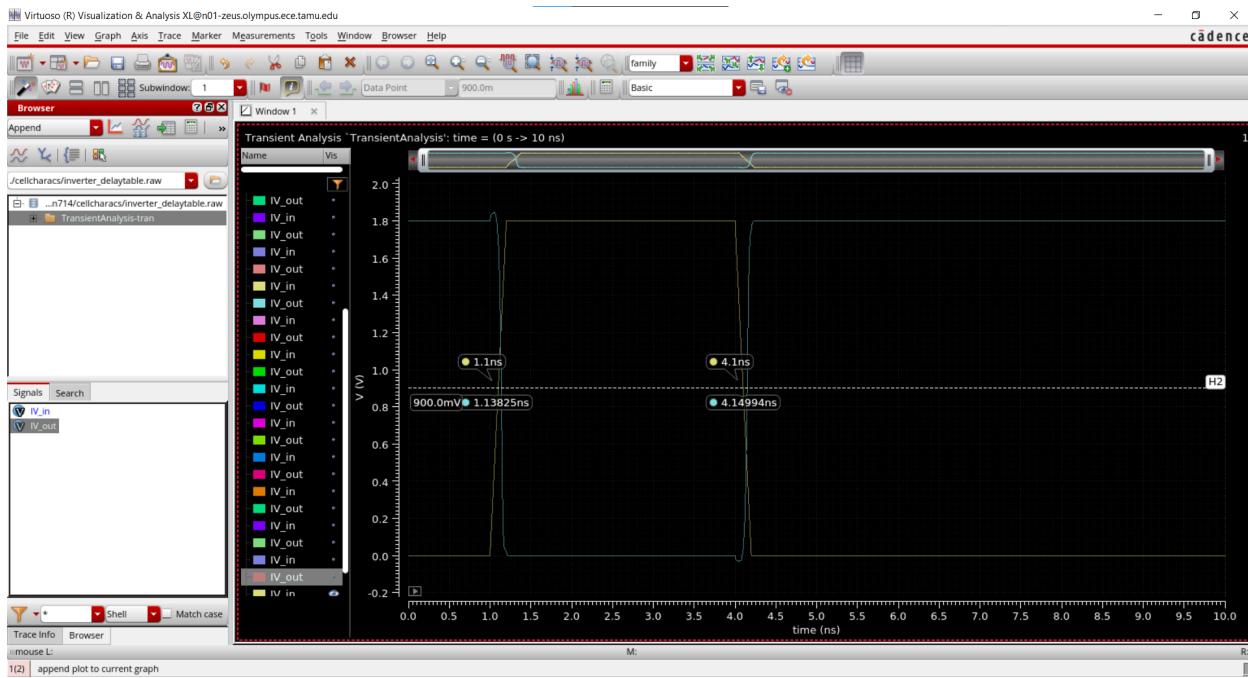


$$\text{Rising delay} = (1.19908 - 1.1) = 0.099$$

$$\text{Falling delay} = (4.20936 - 4.1) = 0.109$$

$$\text{Error \%} = \frac{\text{abs(rising - falling)}}{\min(\text{rising}, \text{falling})} * 100 = \frac{(0.109 - 0.099)}{0.099} * 100 = 10.101\%$$

C=1



$$\text{Rising delay} = (1.13825 - 1.1) = 0.038$$

$$\text{Falling delay} = (4.14994 - 4.1) = 0.05$$

$$\text{Error \%} = \frac{\text{abs(rising - falling)}}{\min(\text{rising}, \text{falling})} * 100 = \frac{(0.05 - 0.038)}{0.038} * 100 = 31.579\%$$

A	B	C	D
Capacitance	Rising Delay (ns)	Falling Delay (ns)	Error %
100	0.859	0.6	5.29%
92	0.613	0.647	5.55%
85	0.57	0.601	5.44%
78	0.528	0.559	5.87%
71	0.485	0.512	5.57%
64	0.443	0.469	5.87%
57	0.399	0.424	6.27%
50	0.358	0.382	6.70%
43	0.313	0.336	7.35%
36	0.27	0.291	7.78%
29	0.228	0.244	7.02%
22	0.185	0.2	8.11%
15	0.142	0.154	8.45%
8	0.099	0.109	10.10%
1	0.038	0.05	31.58%

$f=101.146 \text{ MHz}$



Below is a graph plotting all current vs frequency, with their corresponding sink capacitance:

A	B	C
freq (MHz)	Current (uA)	CC (pF)
100	2.14364	0.00341170904
200	4.22532	0.003362402838
300	6.35657	0.00337226514
400	8.48782	0.003377196291
500	10.6191	0.00338016453
600	12.7503	0.003382122136
700	14.832	0.003372265898
800	16.9632	0.003374721431
900	19.0945	0.003376648974
1000	21.1762	0.003370296924

Then, we solve the average: 0.00337797932 pF

Nand2 waveforms



$$\text{Rising delay} = (1.7564 - 1.1) = 0.656$$

$$\text{Falling delay} = (4.75689 - 4.1) = 0.657$$

$$\text{Error \%} = \text{abs(rising - falling)} / \min(\text{rising}, \text{falling}) * 100 = (0.657 - 0.656) / 0.656 * 100 = 0.152\%$$

Nand2 code

```

2. olympus.ece.tamu.edu      File: nand.spi
GNU nano 2.3.1

;Spice netlist for an inverter and a capacitor
simulator lang=spectre

include "../../../../ecen714/cellcharacs/model18.spi"
include "../../../../ecen714/cellcharacs/cell18.spl"

vgnd (gnd 0) vsource dc=0
vdd (vdd 0) vsource dc=1.8

vpwl (NAND2_in 0) vsource type=pwl wave=[0n 0 1n 0 1.2n 1.8 4n 1.8 4.2n 0]
// A   B   output   VDD GND
X1 (NAND2_in vdd NAND2_out vdd gnd) NAND2 wp=0.65u lp=0.2u wn=0.3u ln=0.2u

R1 (NAND2_out 1) resistor r=1
C1 (1 0) capacitor c=100f

TransientAnalysis tran start=0 stop=10ns step=1ps
save NAND2_in NAND2_out

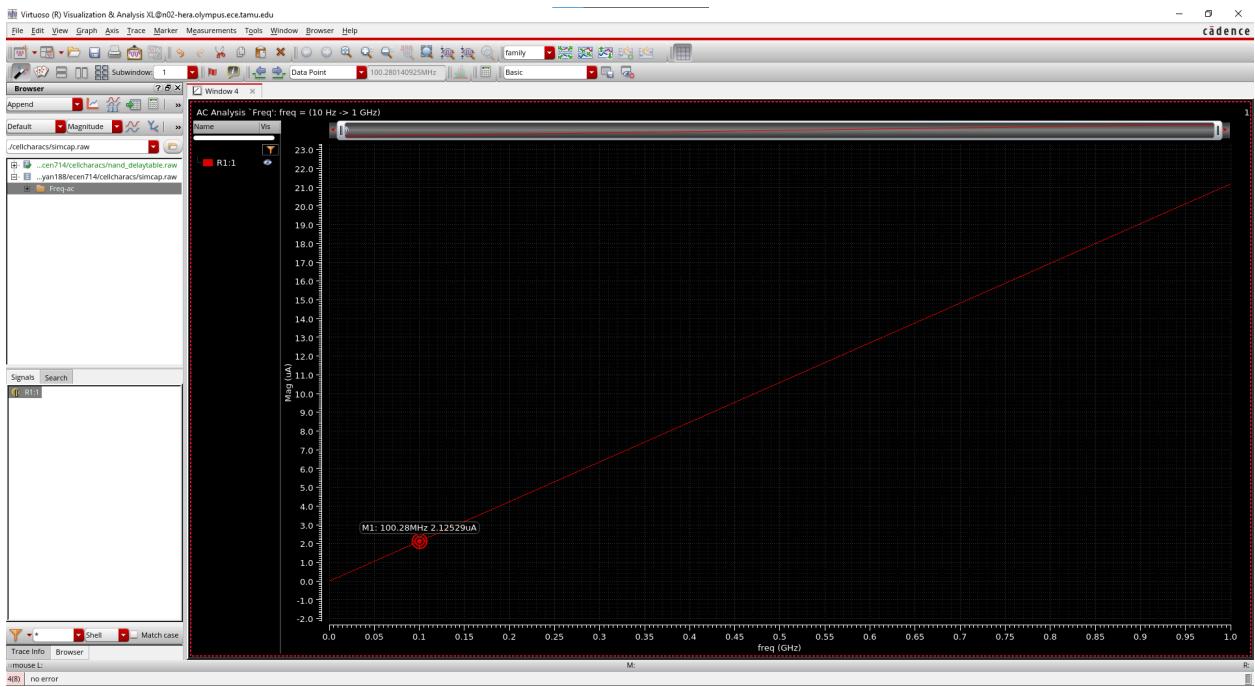
[ Read 26 lines (Converted from DOS format) ]
^G Get Help          ^O WriteOut        ^R Read File        ^Y Prev Page
^X Exit             ^J Justify         ^W Where Is        ^V Next Page
^A Cut Text          ^U UnCut Text      ^C Cur Pos          ^T To Spell
^S MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```

Repeating the above process in the inverter gate, we get the following:

A	B	C	D
Capacitance	Rising Delay	Falling Delay	Error %
100	0.564	0.504	11.91%
92	0.522	0.467	11.78%
85	0.487	0.437	11.44%
78	0.452	0.406	11.33%
71	0.417	0.376	10.90%
64	0.381	0.345	10.44%
57	0.343	0.315	8.89%
50	0.307	0.282	8.87%
43	0.271	0.251	7.97%
36	0.236	0.22	7.27%
29	0.199	0.189	5.29%
22	0.163	0.158	3.17%
15	0.127	0.127	0%
8	0.09	0.096	6.67%
1	0.04	0.052	30%

To solve for the sink capacitance at each frequency, We plot the graph and adjust the slider to get the current at 100MHz to 1000MHz in 100MHz increments:

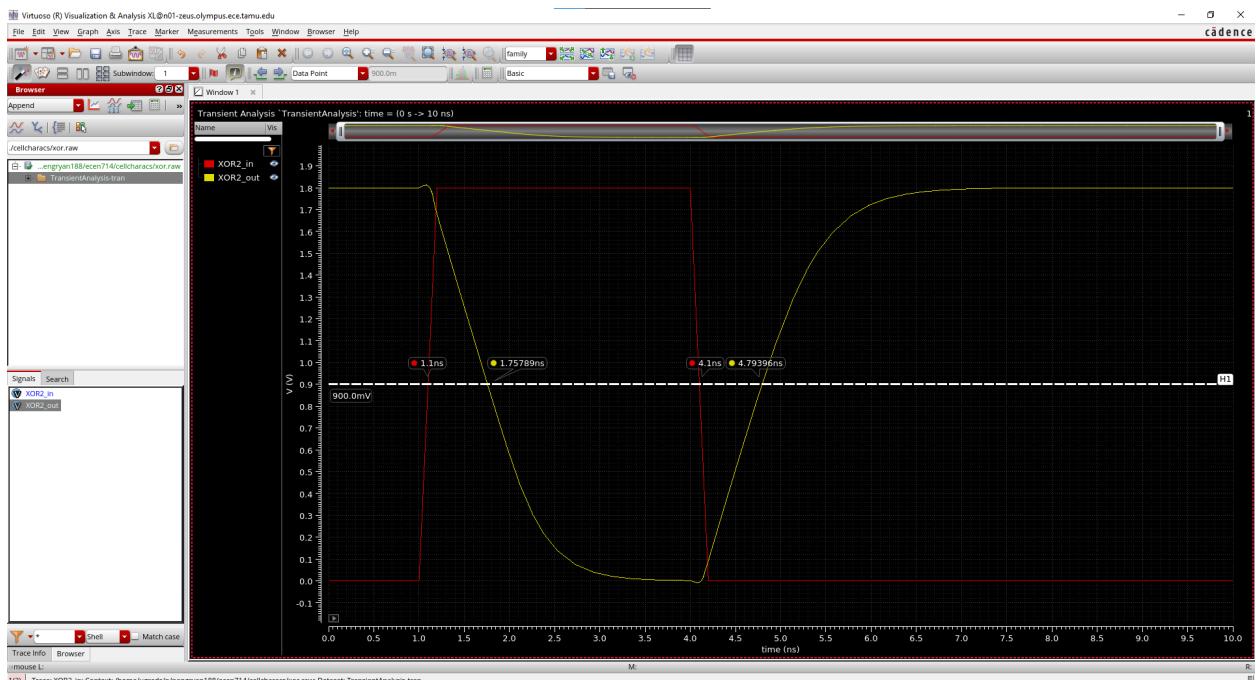


Then, we get the following results:

A	B	C
freq (MHz)	Current (uA)	CC (pF)
100	2.12529	0.003382504108
200	4.25054	0.003382472277
300	6.36289	0.003375618004
400	8.47979	0.003374001255
500	10.5975	0.003373289037
600	12.7196	0.003373978708
700	14.8379	0.003373607347
800	16.9618	0.00337444291
900	19.0706	0.003372422526
1000	21.1858	0.003371824811

Average: 0.003375416098 pF

XOR2



$$\text{Rising delay} = (1.75789 - 1.1) = 0.658$$

$$\text{Falling delay} = (4.79396 - 4.1) = 0.694$$

$$\text{Error \%} = \frac{\text{abs(rising - falling)}}{\text{min(rising, falling)}} * 100 = \frac{(0.694 - 0.658)}{0.658} * 100 = 5.471\%$$

XOR2 code:



The screenshot shows a terminal window titled "2. olympus.ece.tamu.edu" with the file "xor.spi" open. The content of the file is a Spice netlist for a XOR2 gate. It includes model definitions for an inverter and a capacitor, component declarations for the XOR2 gate (X1), and a transient analysis command. The code is as follows:

```
GNU nano 2.3.1
File: xor.spi

;Spice netlist for an inverter and a capacitor
$simulator lang=spectre

INCLUDE "/~/ecen714/cellcharacs/model18.spi"
INCLUDE "/~/ecen714/cellcharacs/cell18.spi"

vnd (gnd 0) vsource dc=0
vdd (vdd 0) vsource dc=1.8

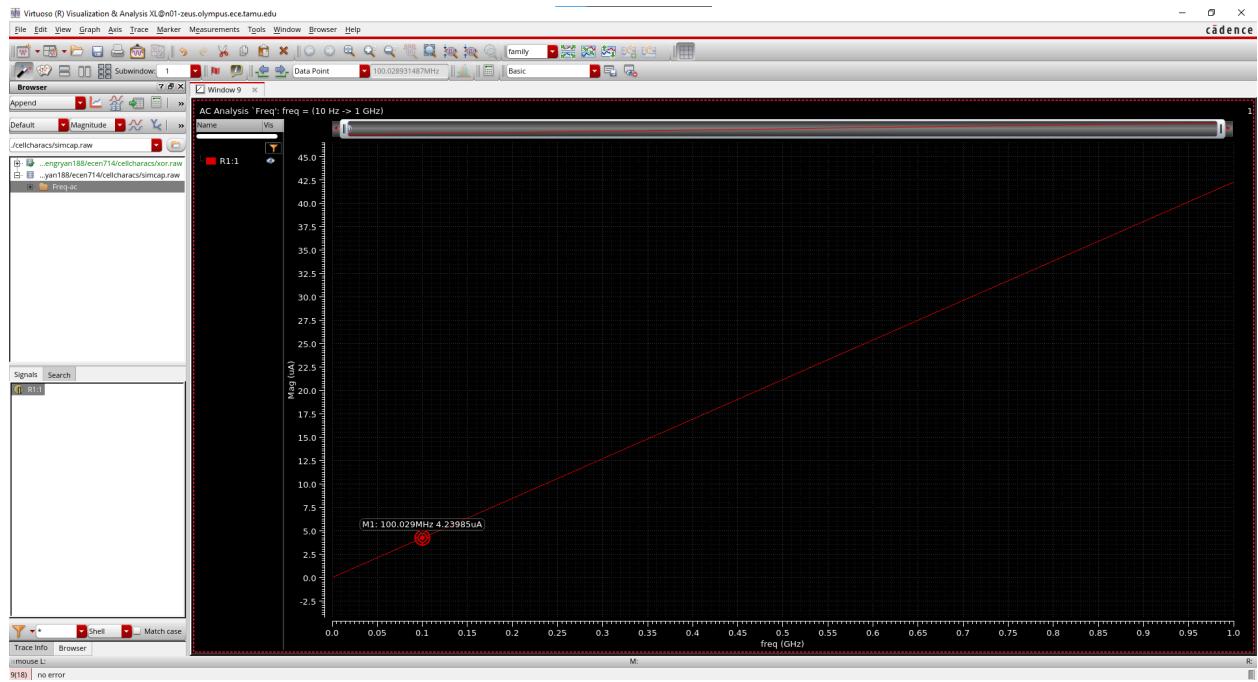
vpwl (XOR2_in 0) vsource type=pwl wave=[0n 0 in 0 1.2n 1.8 4n 1.8 4.2n 0]
X1 (XOR2_in vdd XOR2_out vdd gnd) XOR2 wp=0.9u lp=0.2u wn=0.4u ln=0.2u
R1 (XOR2_out 1) resistor r=1
C1 (1 0) capacitor c=100f

TransientAnalysis tran start=0 stop=10ns step=1ps
save IV_in IV_out
```

Then, we repeat the process from earlier to get the delay for each capacitance:

Capacitance	Rising Delay	Falling Delay	Error %
100	0.743	0.895	20.46%
92	0.687	0.829	20.67%
85	0.638	0.771	20.85%
78	0.589	0.71	20.54%
71	0.54	0.652	20.74%
64	0.491	0.595	21.18%
57	0.442	0.533	20.59%
50	0.393	0.474	20.61%
43	0.344	0.416	20.93%
36	0.294	0.359	22.11%
29	0.245	0.297	21.22%
22	0.196	0.238	21.43%
15	0.146	0.179	22.60%
8	0.096	0.119	23.96%
1	0.038	0.054	42.11%

Then, we plot the simcap.spi and observe the plot:

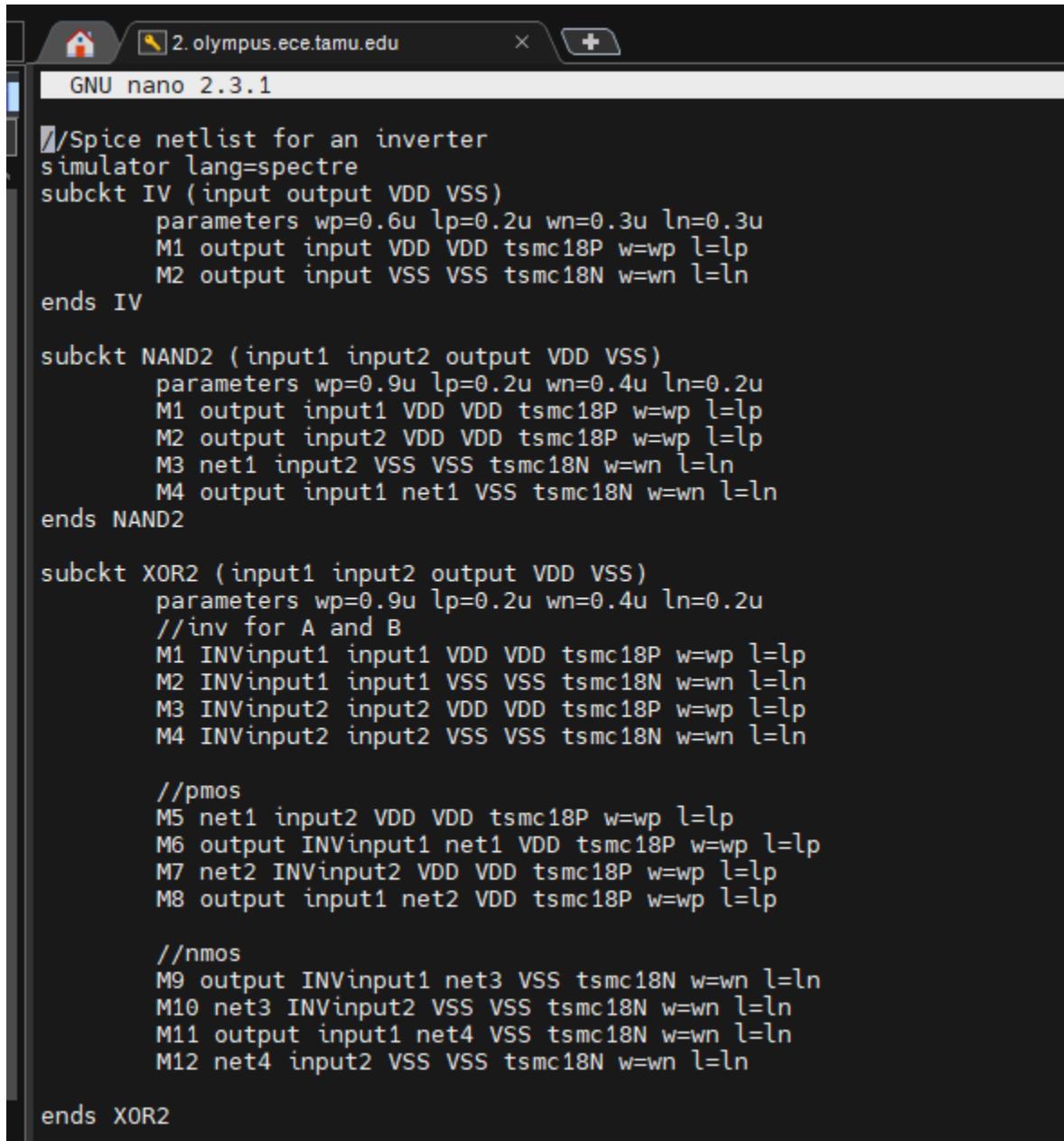


Then, we receive the following results for each marked frequency:

A	B	C
freq (MHz)	Current (uA)	CC (pF)
100	4.23985	0.00674793089
200	8.49272	0.006758291877
300	12.7179	0.006747055538
400	16.9609	0.006748527721
500	21.1849	0.006743363143
600	25.4135	0.006741140279
700	29.6569	0.00674291751
800	33.8763	0.006739475784
900	38.0814	0.006734270091
1000	42.261	0.006726047086

Average: 0.006742901992 pF

Cell18.spi file:



The screenshot shows a terminal window titled "GNU nano 2.3.1" with the URL "2.olympus.ece.tamu.edu" in the address bar. The terminal displays a SPICE netlist for three logic cells: an inverter (IV), a NAND2 gate, and a XOR2 gate. The netlist uses TSMC18P technology with specific dimensions (wp=0.6u, lp=0.2u, wn=0.3u, ln=0.3u for IV; wp=0.9u, lp=0.2u, wn=0.4u, ln=0.2u for NAND2; wp=0.9u, lp=0.2u, wn=0.4u, ln=0.2u for XOR2) and node names (VDD, VSS, M1-M12).

```
//Spice netlist for an inverter
simulator lang=spectre
subckt IV (input output VDD VSS)
    parameters wp=0.6u lp=0.2u wn=0.3u ln=0.3u
    M1 output input VDD VDD tsmc18P w=wp l=lp
    M2 output input VSS VSS tsmc18N w=wn l=ln
ends IV

subckt NAND2 (input1 input2 output VDD VSS)
    parameters wp=0.9u lp=0.2u wn=0.4u ln=0.2u
    M1 output input1 VDD VDD tsmc18P w=wp l=lp
    M2 output input2 VDD VDD tsmc18P w=wp l=lp
    M3 net1 input2 VSS VSS tsmc18N w=wn l=ln
    M4 output input1 net1 VSS tsmc18N w=wn l=ln
ends NAND2

subckt XOR2 (input1 input2 output VDD VSS)
    parameters wp=0.9u lp=0.2u wn=0.4u ln=0.2u
    //inv for A and B
    M1 INVinput1 input1 VDD VDD tsmc18P w=wp l=lp
    M2 INVinput1 input1 VSS VSS tsmc18N w=wn l=ln
    M3 INVinput2 input2 VDD VDD tsmc18P w=wp l=lp
    M4 INVinput2 input2 VSS VSS tsmc18N w=wn l=ln

    //pmos
    M5 net1 input2 VDD VDD tsmc18P w=wp l=lp
    M6 output INVinput1 net1 VDD tsmc18P w=wp l=lp
    M7 net2 INVinput2 VDD VDD tsmc18P w=wp l=lp
    M8 output input1 net2 VDD tsmc18P w=wp l=lp

    //nmos
    M9 output INVinput1 net3 VSS tsmc18N w=wn l=ln
    M10 net3 INVinput2 VSS VSS tsmc18N w=wn l=ln
    M11 output input1 net4 VSS tsmc18N w=wn l=ln
    M12 net4 input2 VSS VSS tsmc18N w=wn l=ln

ends XOR2
```