

מעבדה במבוא למעגלים דוח 6

CMOS Gates

שמות המגישים + ת"ז:

רותם סילם | 206663437

עדי גרין | 324965946

תאריך הגשה:

1. Please design a schematic of the following gates in CMOS topology (Fig 6.1): $\alpha = (0, \text{NAND2 } 1, \text{NOR2 } 6.1)$ where the "2" means the number of inputs to the gates, i.e. 2 is the minimum. Use the β from the assignment parameters. Show proper operation (transient simulation with all 4 possible inputs).

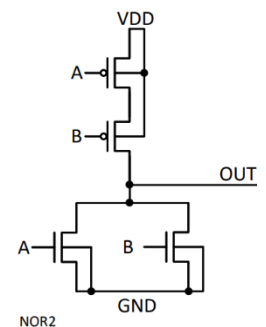
$$\alpha = 13 \bmod 2 = 1$$

$$\beta = 2 + 0.2\alpha = 2.2 \Rightarrow \text{Set } \beta \text{ for the PMOS}$$

מאחר ש- $\alpha=1$, אנו צריכים לייצר את השער NOR2.

פתחנו תיקייה בשם "Lab6", וקובץ בשם "NOR".

יצרנו את NOR2 לפי התמונה המצורפת מהמטלה:



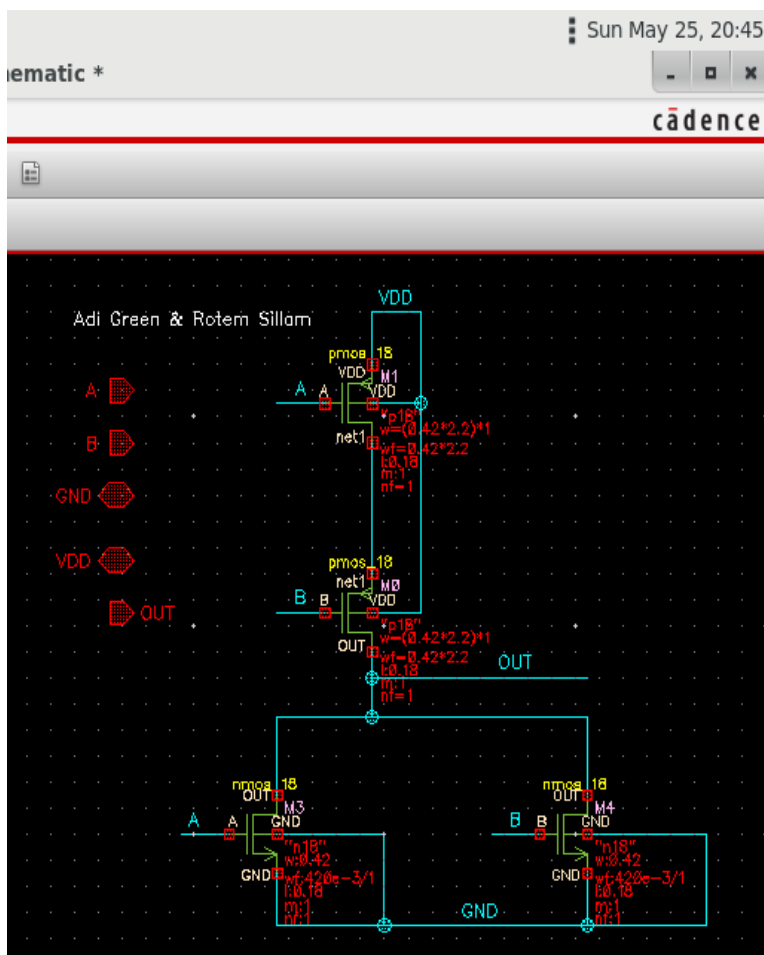
הכפלנו את ה- "Finger width" הדיפולטיבי של PMOS (שהוא 0.42) ב- $\beta=2.2$.

הגדרנו את הפינים:

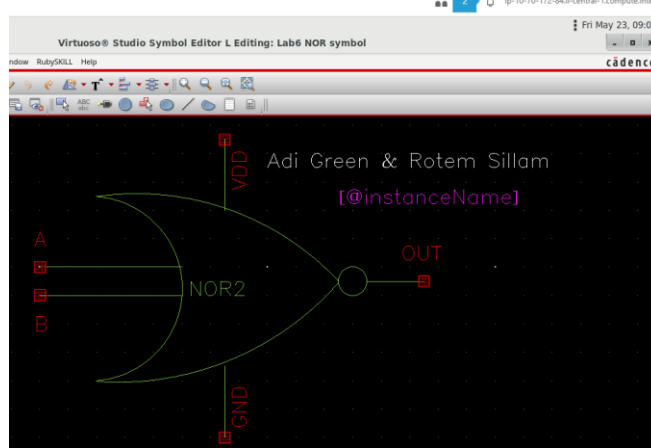
A,B- input

OUT-output

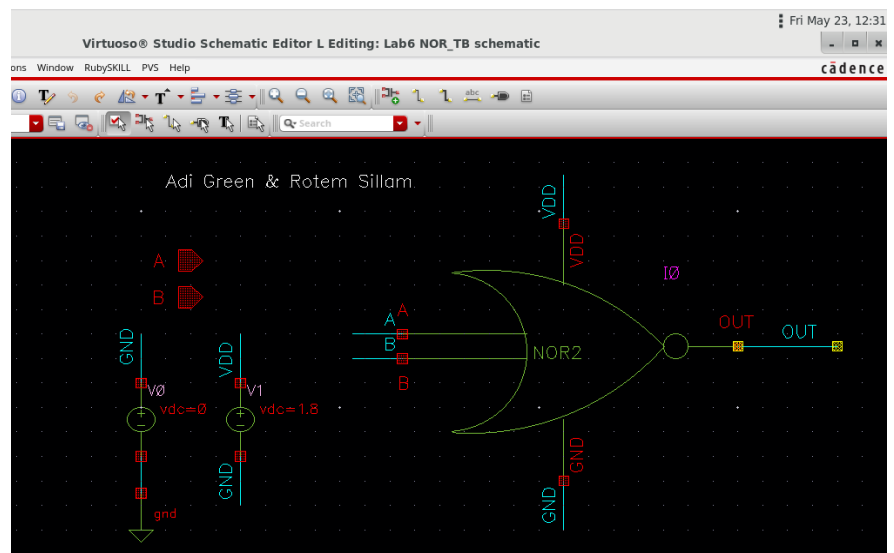
GND,VDD-input and output



יצרנו symbol למעגל:



יצרנו קובץ חדש "NOR_TB", בשביל Test Bench.
 קבענו את VDD להיות 1.8V, מאחר ש: $VDD = 1.5 + \frac{G \bmod 10}{10} = 1.5 + \frac{13 \bmod 10}{10} = 1.8$
 חיברנו פינים ל-A, B מאחר שאלו הכניסות של NOR, וכך נוכל להכניס את הכניסות הלוגיות.



יצרנו קובץ vec, ע"י כתיבת הפקודה "code vec_file.vec" ב-terminal.

```
vec_file.vec x
project > tower > users > sillamr > ws > vec_file.vec
1 #Adi Green & Rotem Sillam
2 radix 1 1
3
4 io i i
5
6 vname A B
7
8 tunit ps
9
10 trise 5
11
12 tfall 5
13
14 vih 1.8
15 vil 0
16
17 voh 1.8
18 vol 0
19
20 0 0 0
21 200 1 0
22 400 0 0
23 600 0 1
24 800 1 1
--
```

כמות הביטים לכל כניסה, במקרה שלנו לכל כניסה יש ביט
 # 1 1 radix - אחד שהינו
 # io i i - מגדיר שכל הפינים הם קלטים
 # vname A B - מגדיר שמות לפינים (לפי הסדר הזה)
 # tunit ps - מגדיר שיחידת הזמן בקובץ היא פיקו שניות .
 # trise 5 - זמן העליה הינו 5 פיקו שניות
 # tfall 5 - זמן הירידה הינו 5 פיקו שניות

הגדרת המתחים של הכניסה והגבוהה בהתאם:
 vih 1.8, vil 0

הגדרת המתחים של היציאה והגבוהה בהתאם
 voh 1.8, vol 0

קביעת הזמנים שבהם הסיגנלים ישתנו, כאשר העמודה הראשונה משמאל מייצגת את זמני השינוי בפיקו לשניות, העמודה השנייה מייצגת את הערכים הלוגים ב-A, והעמודה השלישית את הערכים הלוגיים ב-B.

עשינו סימולציה של tran עם זמן עצירה של 100ns.

ייבאנו את הקובץ vec, ע"י: `vec_file.vec` -> vector files -> simulation files -> setup

ונוסיף לסימולציה את הסיגנלים A,B,OUT.



זהו הגרף שקיבלנו.

ניתן לראות כי אכן קיבלנו את שער NOR כי שרצינו לקבל.

עבור שער זה, אנו מקבלים מתח גבוה ביציאה, כלומר $V_{out}=V_{DD}=1.8V$, רק עבור $A=B=0$.

העליות והירידות אינן חדות ומיידיות, מאחר שמדובר ברכיבים פיזיים ולכן יש זמן השהיה, (נסביר בפירוט בהמשך).

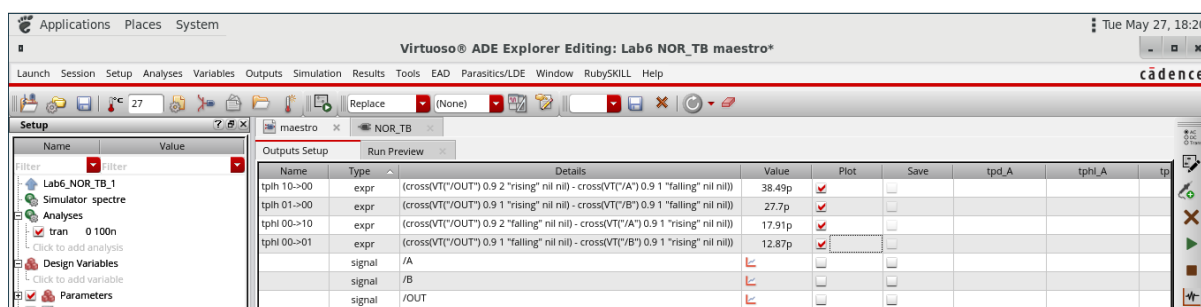
A	B	A B
0	0	1
0	1	0
1	0	0
1	1	0

2. Using simulations, calculate the t_{phl} and t_{plh} of the gate for all possible transitions (use VEC file, you can ignore 00 to 11 and vice versa), and show them in a table in descending order (highest value first). For each value, write the corresponding transitions. Explain the differences.

שינינו את קובץ vec, כך שיהיו כל המעברים האפשריים של עליה לירידה ולהיפך Vout (חוץ מ- 00 < 11, ולהיפך), וזאת על מנת לחשב את הזמנים t_{phl}, t_{plh} כפי שנדרשו.

```
vec_file.vec
project > tower > users > sillamr > ws > vec_file.vec
1 #Adi Green & Rotem Sillam
2 radix 1 1
3
4 io i i
5
6 vname A B
7
8 tunit ps
9
10 trise 5
11
12 tfall 5
13
14 vih 1.8
15 vil 0
16
17 voh 1.8
18 vol 0
19
20 0 0 0
21 200 0 0
22 400 0 1
23 600 0 0
24 800 1 0
25 1000 00
--
```

ייבאנו את הנוסחאות מה-calculator:



כפי שהסברנו במעבדה הקודמת:

הזמן **t_{plh}** - זהו הזמן שלוקח ליציאה לעלות ממתח נמוך למתח גבוה. זהו ההפרש בין הזמן שלוקח ל**V_{out}** לעבור את ה-50% כאשר הוא עולה ממתח נמוך לגבוה, לבין הזמן שלוקח ל**V_{in}** לעבור את ה-50% כאשר הוא יורד ממתח גבוה לנמוך. הזמן **t_{phl}** - זהו הזמן שלוקח ליציאה לרדת ממתח גבוה למתח נמוך. זהו ההפרש בין הזמן שלוקח ל**V_{out}** לעבור את ה-50% כאשר הוא יורד ממתח גבוה לנמוך, לבין הזמן שלוקח ל**V_{in}** לעבור את ה-50% כאשר הוא עולה ממתח נמוך לגבוה. את החישוב הזה נעשה בעזרת הפונקציה שהגדרנו עבורו בcalculator, כאשר היא משמשת בפונקציית cross המחזירה את הזמן שבו האות חוצה את הערך שהגדרנו לו (אנחנו הגדרנו $VDD \cdot 0.5 = 0.9V$), ואת כיוונו (כיוון עלייה או ירידה בהתאם לנדרש).

לשים לב כי כאשר כתבנו 2 לפני rising/falling של **V_{out}**, הסיבה היא שזאת הפעם שניה ש**V_{out}** יורד/עולה.

ניתן לראות בגרף את הנקודות של העליה/ הירידה שבהן המתח שווה ל- $V_{DD} \cdot 0.5 = 0.9V$.

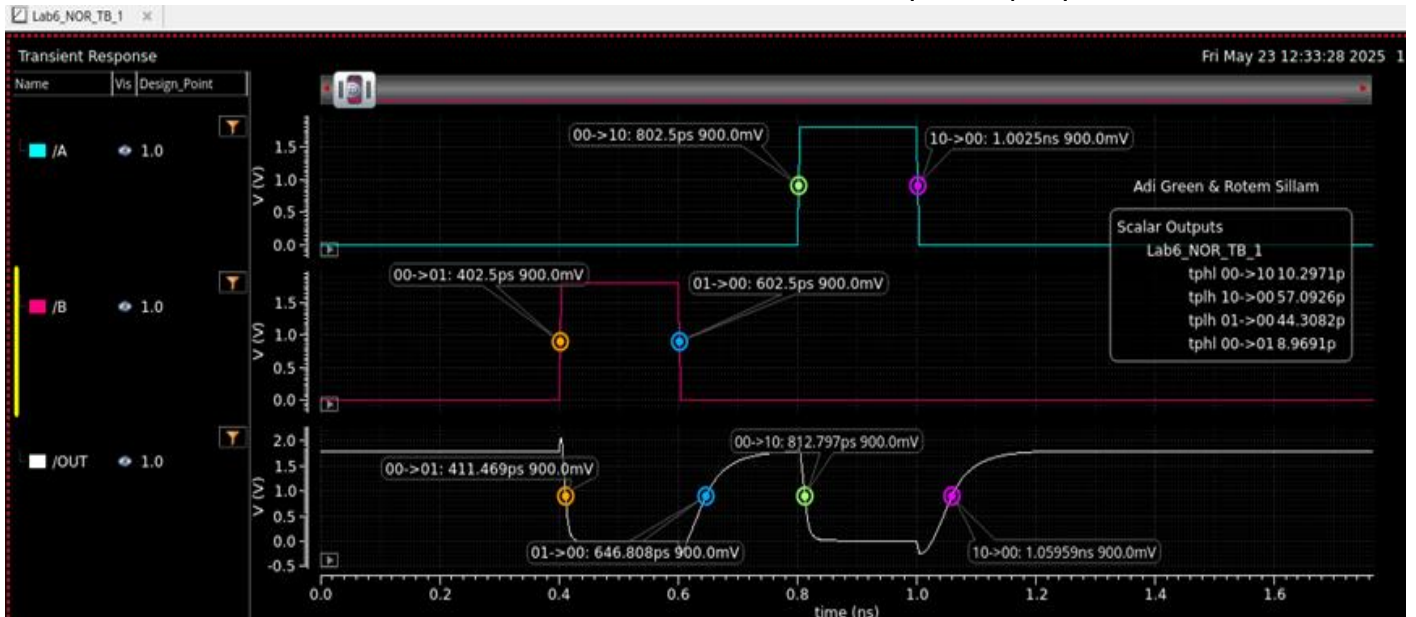
הנקודות הכתומות הן $t_{phl} 00 \rightarrow 01$

הנקודות הכחולות הן $t_{plh} 00 \rightarrow 01$

הנקודות הירוקות הן $t_{phl} 10 \rightarrow 00$

הנקודות הסגולות הן $t_{plh} 00 \rightarrow 10$

חשבנו ידנית את ההפרשים בין הנקודות ואכן יצאו תשובות זהות לתשובות של ה-calculator:



נשווה את הזמנים:

$t_{plh} 10 \rightarrow 00$	expr	$(\text{cross}(VT("/OUT")) 0.9 2 \text{ "rising" nil nil}) - \text{cross}(VT("/A") 0.9 1 \text{ "falling" nil nil})$	38.49p
$t_{plh} 01 \rightarrow 00$	expr	$(\text{cross}(VT("/OUT")) 0.9 1 \text{ "rising" nil nil}) - \text{cross}(VT("/B") 0.9 1 \text{ "falling" nil nil})$	27.7p
$t_{phl} 00 \rightarrow 10$	expr	$(\text{cross}(VT("/OUT")) 0.9 2 \text{ "falling" nil nil}) - \text{cross}(VT("/A") 0.9 1 \text{ "rising" nil nil})$	17.91p
$t_{phl} 00 \rightarrow 01$	expr	$(\text{cross}(VT("/OUT")) 0.9 1 \text{ "falling" nil nil}) - \text{cross}(VT("/B") 0.9 1 \text{ "rising" nil nil})$	12.87p

כפי שניתן לראות, הזמנים של t_{plh} יותר גדולים מהזמנים של t_{phl} . ישנן 2 סיבות:

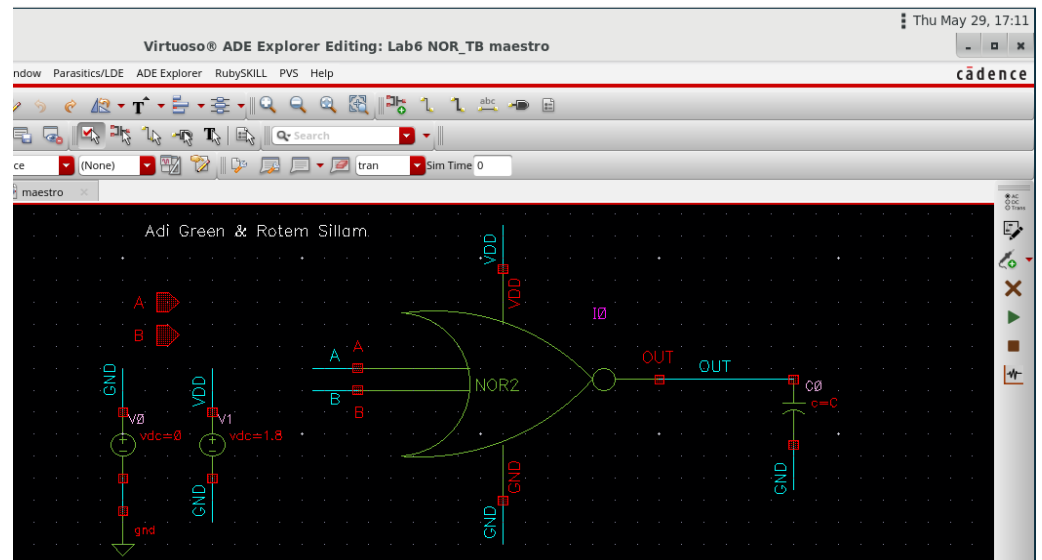
1. כפי שאמרנו, t_{phl} זהו הזמן שבו האות יורד מ-High ל-Low, ולכן NMOS פורק את הקבל לקרקע. הזמן t_{plh} זהו הזמן שבו האות עולה מ-Low ל-High, ולכן PMOS טוען את הקבל מה- V_{DD} . כפי שראינו במעבדה הקודמת, הזרמים ב-NMOS ו-PMOS מאוזנים כאשר β שווה בקירוב ל-3.7. במקרה שלנו, $\beta = 2$, ולכן NMOS חזק יותר (הראנו במעבדה הקודמת שכל שהערך של β קטנה יותר מערך ה- β של המצב המאוזן-NMOS חזק יותר). מאחר ש-NMOS חזק יותר, הזרם שלו חזק יותר ולכן הוא פורק יותר מהר מהטעינה של PMOS.
2. ניתן לראות שבשער NOR, רכיבי NMOS מחוברים במקביל ורכיבי PMOS מחוברים בטור. בטור יש צורך לבדוק כי שני הרכיבים דולקים והתעלה יותר ארוכה (מחוברים ביחד), ולכן הזמן יותר איטי. במקביל, מספיק שאחד מהרכיבים דלוק ויש יותר מסלולים לעבור דרכם, ולכן הזמן יותר מהיר. ולכן רכיבי ה-NMOS יותר מהירים מרכיבי ה-PMOS.

בנוסף, כפי שניתן לראות, הזמנים שכאשר הערך של A משתנה ($00 \rightarrow 10$ ולהיפך), יותר גדולים מהזמנים ש-B משתנה ($01 \rightarrow 00$ ולהיפך).

הסיבה היא שברכיבי ה-PMOS, כפי שניתן לראות, רכיב הכניסה של A יותר רחוק פיזית מה-out מאשר רכיב הכניסה של B, ולכן יקח לו יותר זמן להגיע ל-out.

ברכיבי ה-NMOS, אין הבדל במרחק הפיזי מה-OUT, ולכן לא אמור להיות הבדל בזמנים משום שהם מחוברים במקביל. ואכן קיבלנו הבדל של כ-5 פיקו, בגלל שגיאות זניחות. (מבדיקה עם מתרגל, זה אכן תקין).

3. Add a capacitor at the output of the gate. Calculate the dynamic energy for each transition from before that changed the output for 10 different values of capacitance from 1[fF] to 100[fF], and find the worst one. Make sure to set period long enough so the biggest capacitor will be fully charged. Also, calculate the static power for each input state. Explain. In the lecture we saw $E_{dynamic} = CL * VDD^2$, does it match the results? Proof.



```

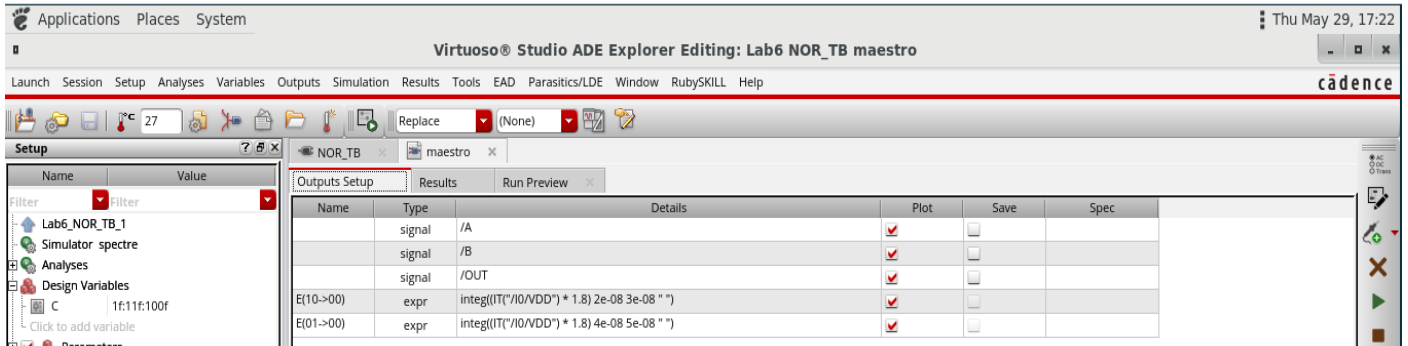
vec_file.vec
project > tower > users > sillamr > ws > vec_file.vec
1  #Adi Green & Rotem Sillam
2  radix 1 1
3
4  io i i
5
6  vname A B
7
8  tunit ps
9
10 trise 5
11
12 tfall 5
13
14 vih 1.8
15 vil 0
16
17 voh 1.8
18 vol 0
19
20 0 0 0
21 10000 1 0
22 20000 0 0
23 30000 0 1
24 40000 0 0
25

```

הגדרנו קבל שקופץ עבור 10 ערכים שונים בין 1f ל100f כנדרש.
בפריקה הזרם בVDD הינו אפסי (זליגה) ולכן אנו בודקים את האנרגיה רק בעליות של Vout, כלומר בטעינת הקבל.

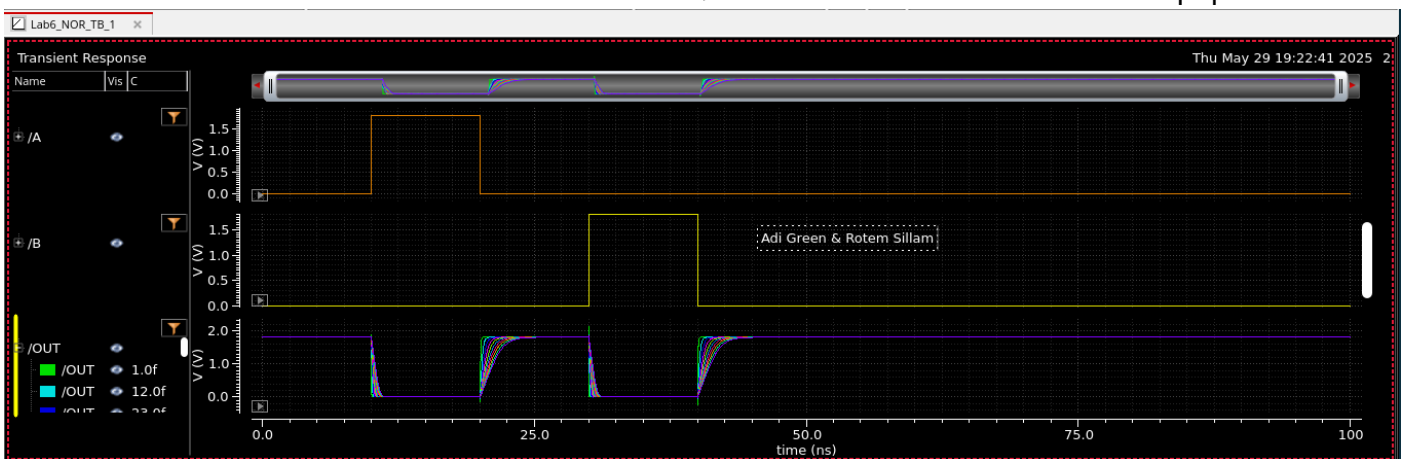
Input change (AB->AB) כניסות מתח לוגי	Output change (volt)	זמן התחלה (sec)	זמן סיום (sec)
10->00	0->1	20n	30n
01->00	0->1	40n	50n

הגדרנו 2 פונקציות לחישוב האנרגיה הדינמית בטעינה (בחרנו זמנים בהתאם), כפי שלמדנו בהרצאה: $\int_{t_0}^{t_1} I_{VDD} * VDD$



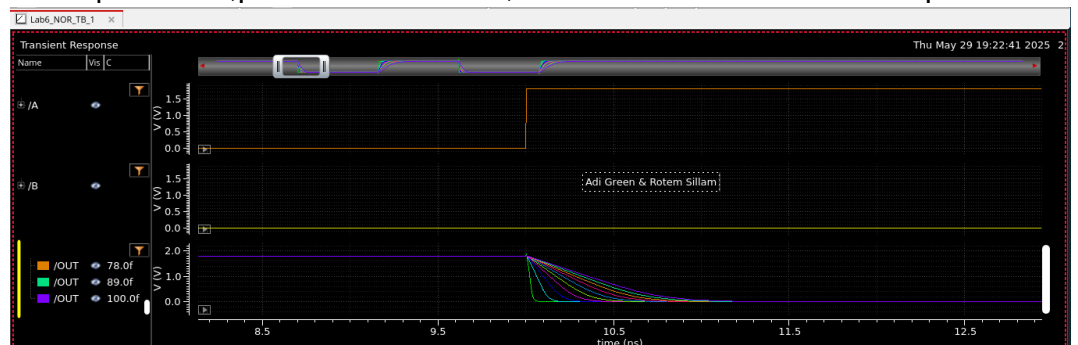
קיבלנו את הגרף הבא:

- בהתחלה ראינו שעבור הזמנים ששמנו בסעיף הקודם, לא קיבלנו עבור כל קיבול הגעה ל VDD (התמלאות קבל), לכן שינו בקובץ vec לזמנים יותר גדולים כדי שיוכלו להגיע למתח זה.

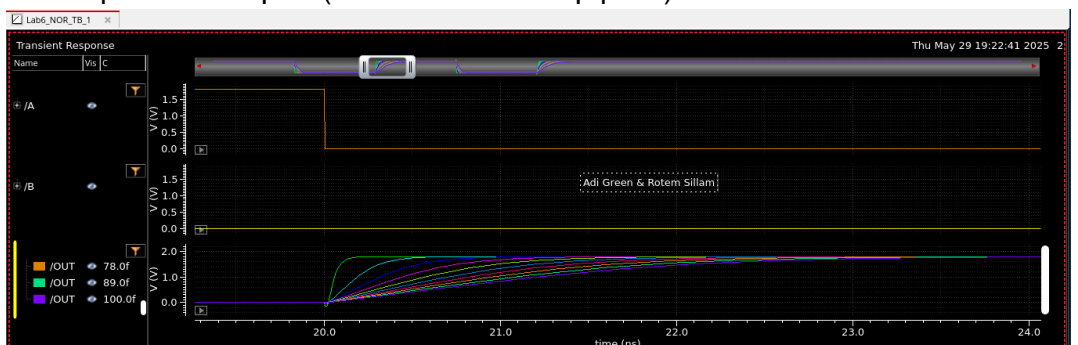


ניתן לראות שיש התנהגות שונה עבור פריקה וטעינה (יוצא שלא בהכרח סימטריים). זה נובע מכך שמסמך אחראי על טעינה, וסמך על פריקה, ויש שוני בתנאים שלהם.

בירידה - ניתן לראות שכל שאנחנו מגדילים את C, השיפוע נהיה יותר מתון, כלומר הדק גדל.



בעליה - ככל שאנחנו מגדילים את C, השיפוע נהיה יותר חד, כלומר הדק קטן. כלומר נצפה לקבל את האנרגיה הדינמית הגדולה ביותר עבור C הגדול ביותר (C הקטן ביותר הוא ההכי גרוע). ניתן לראות שמתקיים לפי הנתונים בטבלה הבאה.



קיבלנו עבור 2 העליות שלנו את הנתונים הבאים:

נציג בטבלה על מנת להציג את התוצאות שקיבלנו מהוירטואוז, ביחס לתוצאות שקיבלנו עבור חישוב ידני של E לפי הנוסחה
 הנתונה: $E_{dynamic} = C * VDD^2$
 הנתונים שיצאו עבור E(10->00):

VDD[V]	C[F]	E(10->00) virtuoso [J]	Edaynamic- חישוב ידני [J]	delta = Evirtuoso - ידני
1.8	1.00E-15	1.53E-14	3.24E-15	1.20E-14
1.8	1.20E-14	5.10E-14	3.89E-14	1.21E-14
1.8	2.30E-14	8.66E-14	7.45E-14	1.21E-14
1.8	3.40E-14	1.22E-13	1.10E-13	1.21E-14
1.8	4.50E-14	1.58E-13	1.46E-13	1.21E-14
1.8	5.60E-14	1.94E-13	1.81E-13	1.22E-14
1.8	6.70E-14	2.29E-13	2.17E-13	1.21E-14
1.8	7.80E-14	2.65E-13	2.53E-13	1.21E-14
1.8	8.90E-14	3.01E-13	2.88E-13	1.21E-14
1.8	1.00E-13	3.36E-13	3.24E-13	1.21E-14

הנתונים שיצאו עבור E(01->00):

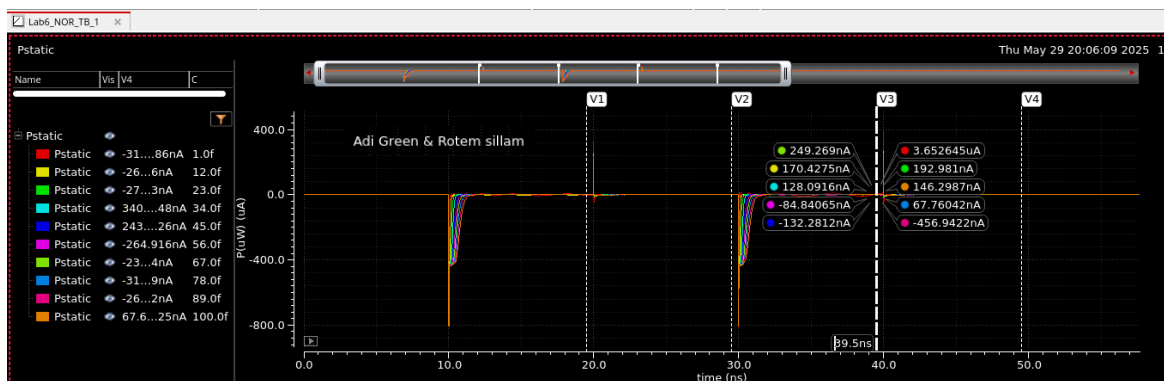
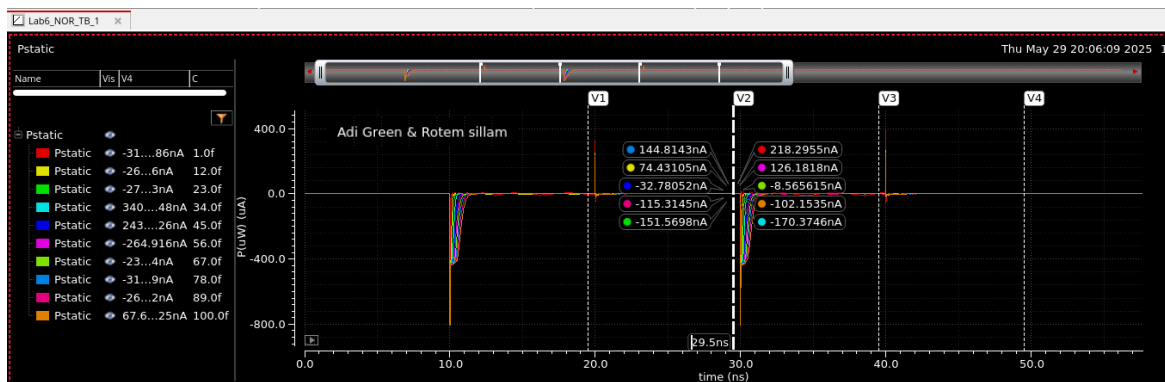
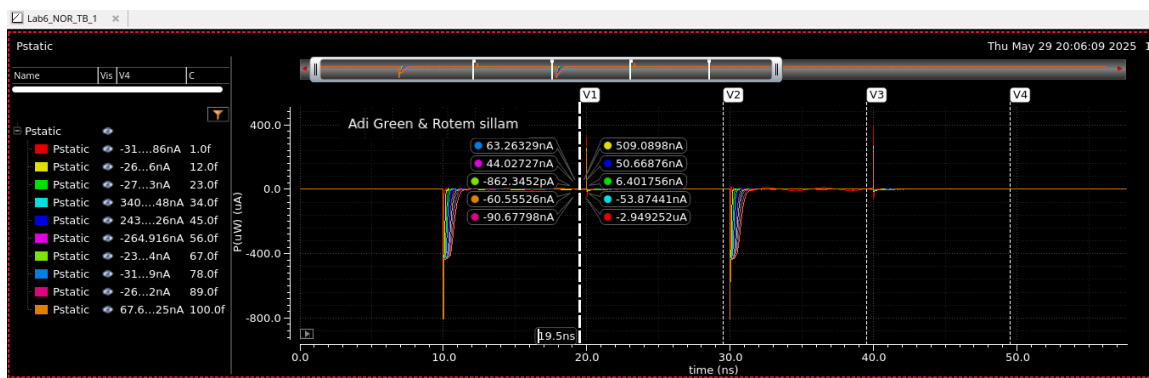
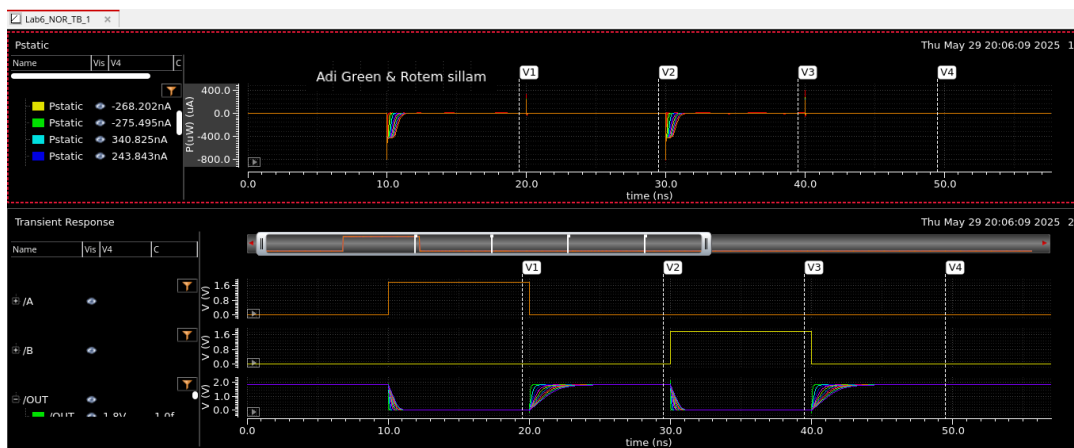
VDD[V]	C[F]	E(01->00) virtuoso [J]	Edaynamic- חישוב ידני [J]	delta = Evirtuoso - ידני
1.8	1.00E-15	1.34E-14	3.24E-15	1.01E-14
1.8	1.20E-14	4.79E-14	3.89E-14	8.97E-15
1.8	2.30E-14	8.37E-14	7.45E-14	9.21E-15
1.8	3.40E-14	1.19E-13	1.10E-13	9.14E-15
1.8	4.50E-14	1.55E-13	1.46E-13	9.20E-15
1.8	5.60E-14	1.91E-13	1.81E-13	9.06E-15
1.8	6.70E-14	2.26E-13	2.17E-13	9.22E-15
1.8	7.80E-14	2.62E-13	2.53E-13	9.28E-15
1.8	8.90E-14	2.98E-13	2.88E-13	9.24E-15
1.8	1.00E-13	3.33E-13	3.24E-13	9.10E-15

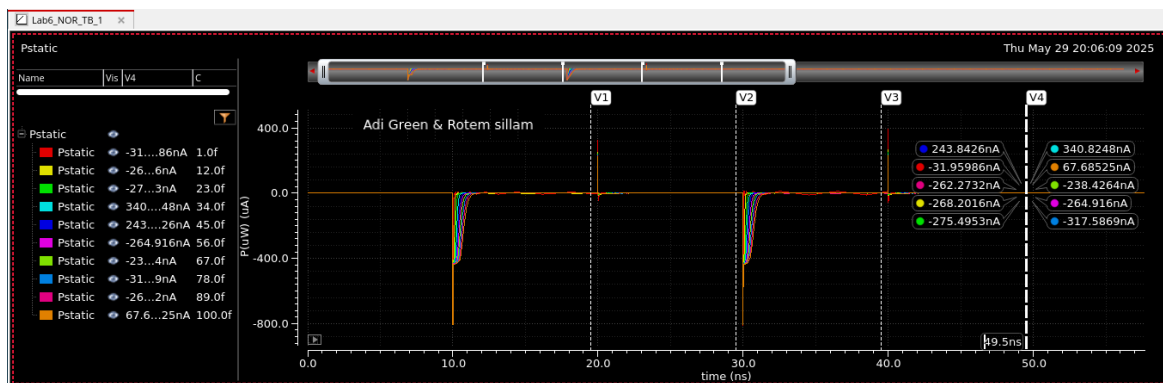
ניתן לראות כי ישנו הבדל בין החישוב הידני לתוצאות של המאסטר בקירוב של 9f[J] עבור E(01->00), וקירוב של 12.1f[J] עבור E(10->00). הסיבה לכך שאנחנו לא מקבלים בדיוק אותה אנרגיה בשניהם, היא שישנם זרמי זליגה וקיבולים פרזיטים המשפיעים על האנרגיה הכוללת של המערכת, שהנוסחה $E_{dynamic} = C * VDD^2$ לא לוקחת בחשבון.

הספק סטטי:

- הספק זה קיים משום שיש זרם זליגה (לא נקבל זרם 0 גם כשהרכיב לא עובד). הספק זה מחושב באמצעות הנוסחה:

$$P_{static} = I_{GND} * VDD = I_{GND} * 1.8$$





Inputs(AB)	max value(W)
00->10	509.0898 n
10->00	218.2955 n
00->01	3652.645 n
01->00	340.8248 n

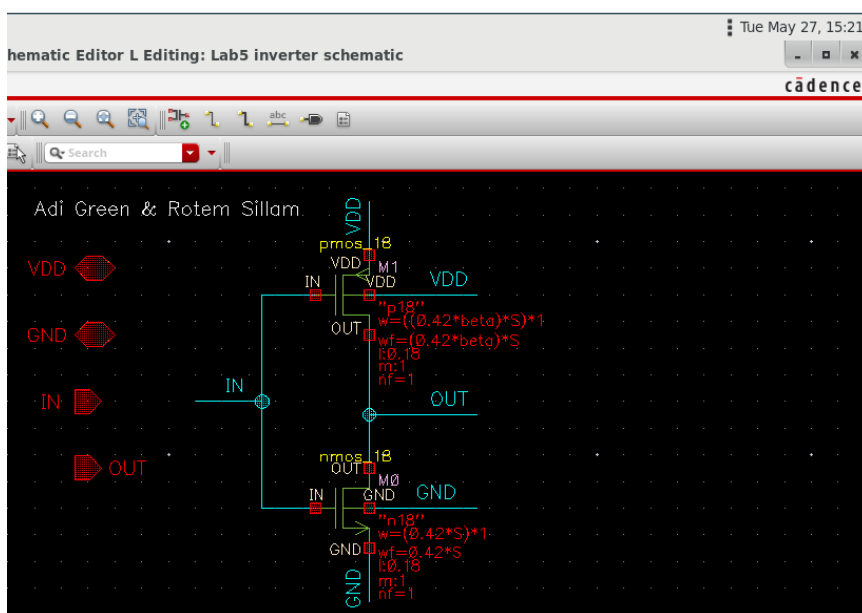
מהטבלה ניתן לראות שערך הזרם המרבי (שהוא מקור ההספק הדינמי) משתנה בהתאם לשינוי במצב הכניסות.

הסיבה לפערים הללו נובעת מהשונות בכמות הטרנזיסטורים הפעילים ובנתיב הזרם בעת כל מעבר.

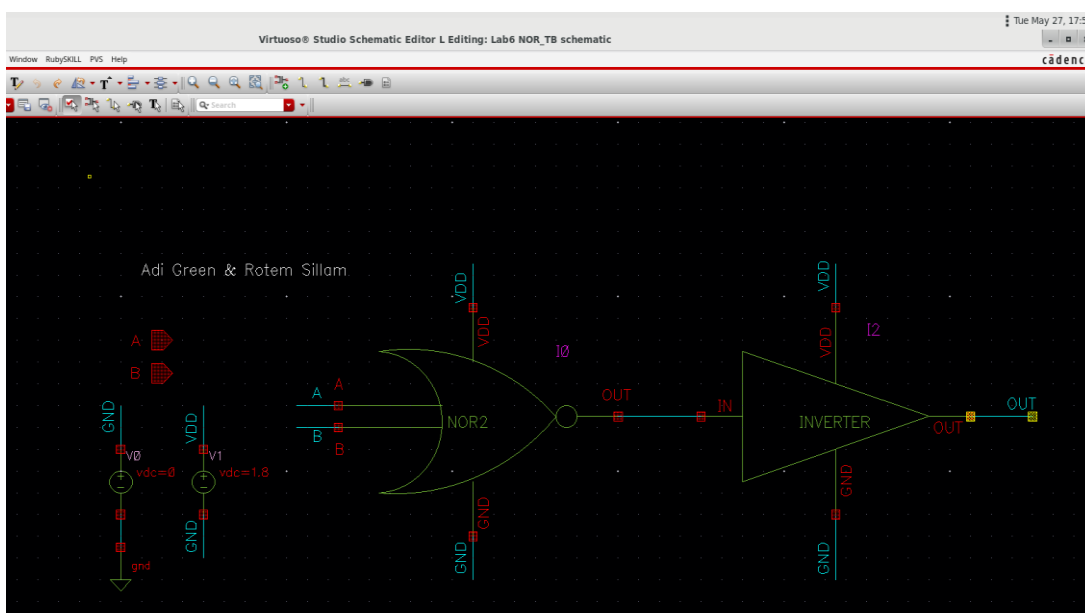
כאשר יותר טרנזיסטורים מסוג NMOS פעילים בו זמנית, הזרם שזורם גדול יותר, ולכן ההספק המחושב באמצעות $p=VI$ יהיה גבוהה יותר.

4. Now remove the capacitor and add an Inverter from Lab 5. What gate did we created? Adjust the Inverter so we have a parameter("S") that multiply the minimum gate width both in the nMOS and pMOS, similarly to what we did in Lab 5. Calculate tpd for your gate(NAND or NOR) for a single transition, for 10 values of S from 1-10. Explain the results.


ניתן לראות שהוספנו פרמטר S למos וlmos כפי שהתבקשנו. כשנריץ את האנליזה, נגדיר שהוא ירוץ מ1 עד 10, בקפיצות של 1 כנדרש.



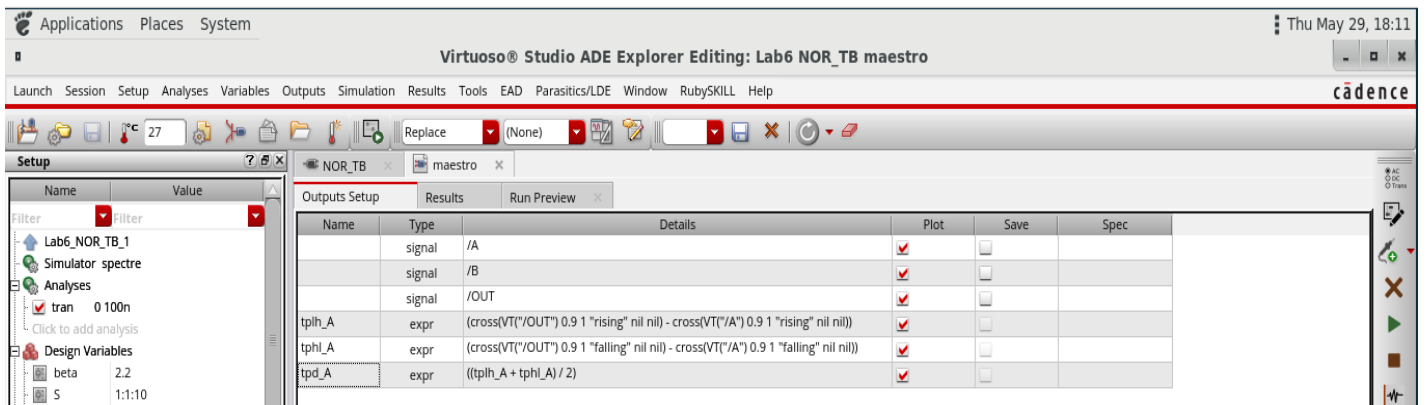
נחבר Inverter למוצא הNOR שבינינו ונמחק את הקבל.



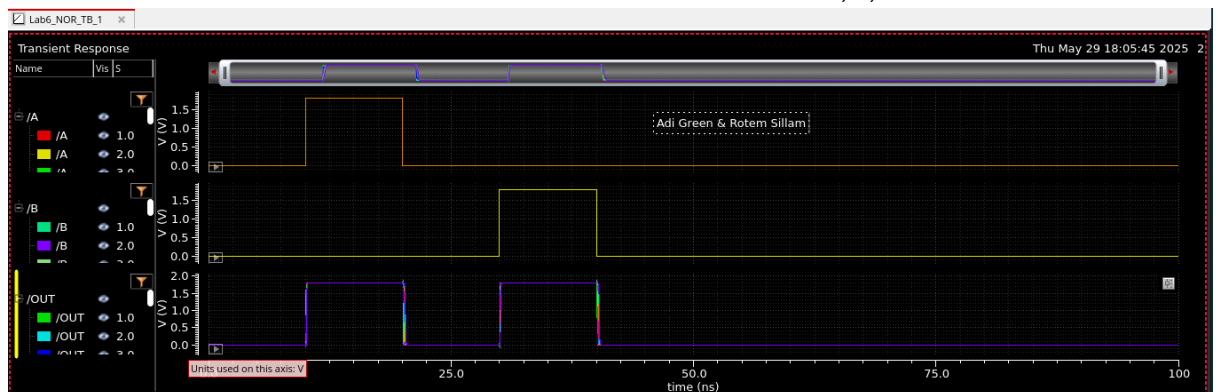
```

project > tower > users > sillamr > ws >  vec_file.vec
1      #Adi Green & Rotem Sillam
2
3      radix 1 1
4
5      io i 1
6
7      vname A B
8
9      tunit ps
10
11     trise 5
12
13     tfall 5
14
15     vih 1.8
16     vil 0
17
18     voh 1.8
19     vol 0
20
21     0 0 0
22     10000 1 0
23     20000 0 0
24     30000 0 1
25     40000 0 0
26

```



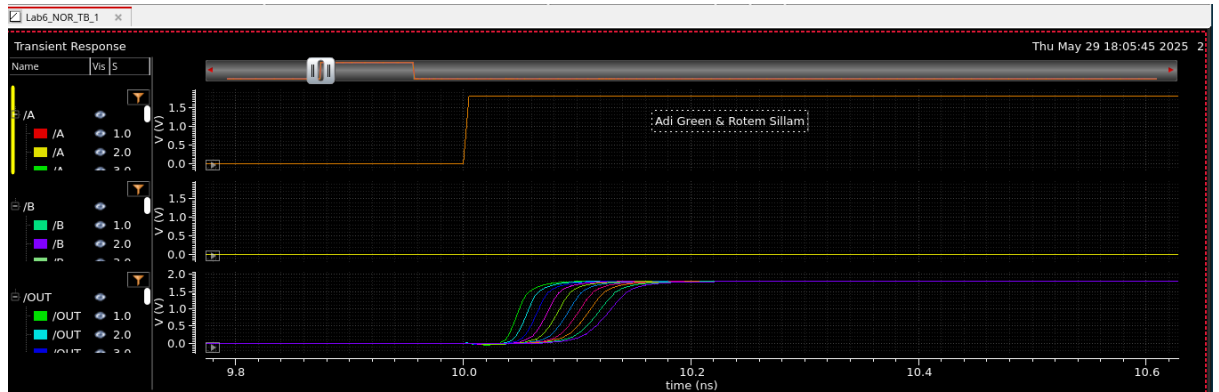
הדפסנו גרפים של הכניסות A, B, והמוצא Vout התלוי בהם:



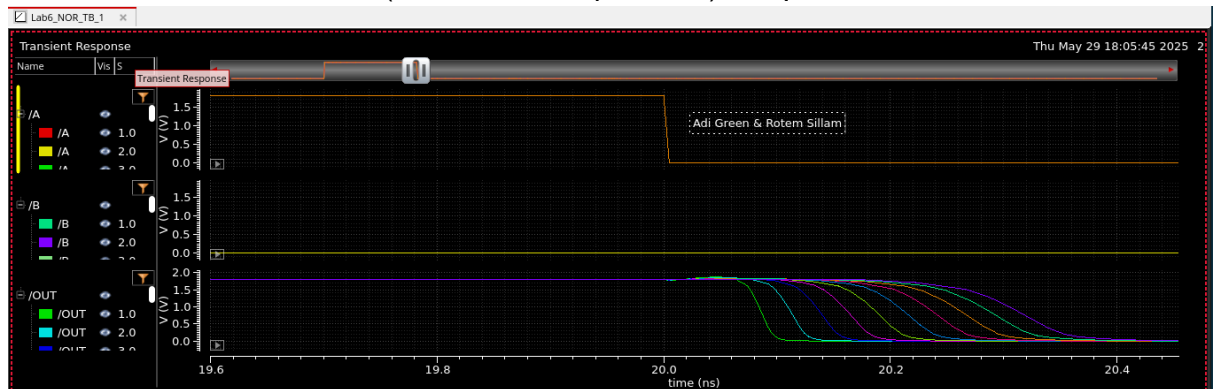
ניתן לראות שקיבלנו שער OR:

A	B	Vout
0	0	0
1	0	1
0	1	1
1	1	1

בנוסף ניתן לראות שבמוצא אנחנו מקבלים מספר שיפועים שונים כתלות בS שהכנסנו.
עליה: ככל שS גדול יותר השיפוע מתון (כלומר t_{phl} גדל ככל שS גדל).



ירידה: ככל שS גדול יותר השיפוע מתון יותר (כלומר t_{phl} גדל ככל שS גדל)



נבצע את המדידות הבאות על מנת לחשב זמנים:

עבור הכניסה A נחשב את t_{phl} , ומתי V_{out} חוצה את 0.9 כלומר את $2/VDD$ בפעם הראשונה כתוצאה מהכניסה A).

לאחר מכן נחשב את t_{pd} :
$$\frac{t_{phl} + t_{plh}}{2}$$

Virtuoso® Studio ADE Assembler Editing: Lab6 NOR_TB maestro

Launch File Create Tools Options Run EAD Parasitics/LDE Window RubySKILL Help

No Parasitics/LDE No Sweeps Single Run, Sweeps and Corners Reference History Interactive

Data View

Filter Name Value

Tests

Lab6_NOR_TB_1

Global Variables

Parameters

Corners

Documents

Setup States

Data History

Run Summary

1 Test

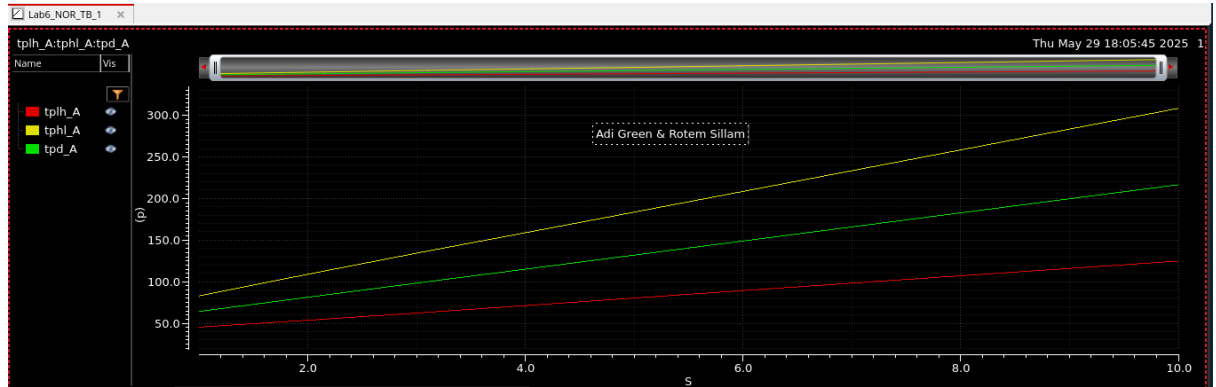
1 Point Sweep (Test Sweep Pc) 0 Corner

Outputs Setup Results Run Preview

Detail - Transpose

Point	Corner	S	/A	/B	/OUT	t_{phl_A}	t_{phl_A}	t_{pd_A}
1	nom	1				45.46p	83.03p	64.25p
2	nom	2				53.72p	109p	81.36p
3	nom	3				62.34p	134.4p	98.37p
4	nom	4				71.32p	158.9p	115.1p
5	nom	5				80.39p	183.6p	132p
6	nom	6				89.35p	208.5p	148.9p
7	nom	7				98.31p	233.4p	165.9p
8	nom	8				107.1p	258.4p	182.8p
9	nom	9				116p	283.4p	199.7p
10	nom	10				124.8p	308.3p	216.5p

ניתן לראות שקיבלנו גרף של t_{plh} , t_{phl} , t_{pd} כתלות ב C (זה הממוצע של השניים האחרים):



מסקנות:

כאשר ערך S עולה, נראה שיש עליה בזמן t_{pd} . תופעה זו נובעת מהרחבת ה-NMOS וה-PMOS (הכפלו ב finger width את הרחב פי S , כלומר ככל ש S גדל, רחב כל טרנזיסטור גדל).

מתקיים trade-off:

- הגדלת רחב הטרנזיסטור מפחיתה את ההתנגדות שלו, כלומר תתאפשר זרימה יותר גדולה\מהירה של זרם, וכתוצאה מכך זמן ההשהיה יתקצר. כלומר ההתנהגות תהיה כך: $R_{on} \propto \frac{1}{W}$
- מצד שני מתקיימת עלייה בקיבול של הטרנזיסטור כתוצאה מהרחבתו. כלומר לפי מה שלמדנו בהרצאה, ההתנהגות תהיה כך: $C \propto W$
- קיבול גבוה יותר מאריך את זמן הטעינה והפריקה של הקבל, ובכך מאט את תגובת המעגל.
- ניתן לראות בשלב זה שהגדלת W תשפיע יותר על הקיבול של הטרנזיסטור (יחס ישר) מאשר על ההתנגדות. נוסף על כך, בשער NOR Pmos-ים מחוברים בטור.

מבחינת התנגדות - ההתנהגות היא כמו $\frac{1}{R}$ ולכן חיבור בטור רק יקטין את ההתנגדות הכללית. כלומר נקבל באמת שהקיבול יותר מושפע מההגדלה של W .

- למדנו בהרצאה את נוסחה זו: $t_{pd} = 0.69R_{eq} (C_{int} + C_{ext})$
- ובמקרה שלנו: $t_{pd} = 0.69R_{NOR2} (C_{NOR2} + S * C_{inverter})$

