THE AMERICAN UNIVERSITY IN CAIRO
الجامعة الأمريكية بالقاهرة

Microelectronics Project Report

4-1 Multiplexer Using CMOS Design

Ismaiel Sabet 900221277 ismaielsabet@aucegypt.edu

Habiba Elsayed 900221264 hab20034@aucegypt.edu

Content

Introduction

In this project, we set out to implement a simple inverter circuit and then a 4x1 multiplexer using LTSpice. We accomplished both of these tasks, the results and process of which is discussed in the report.

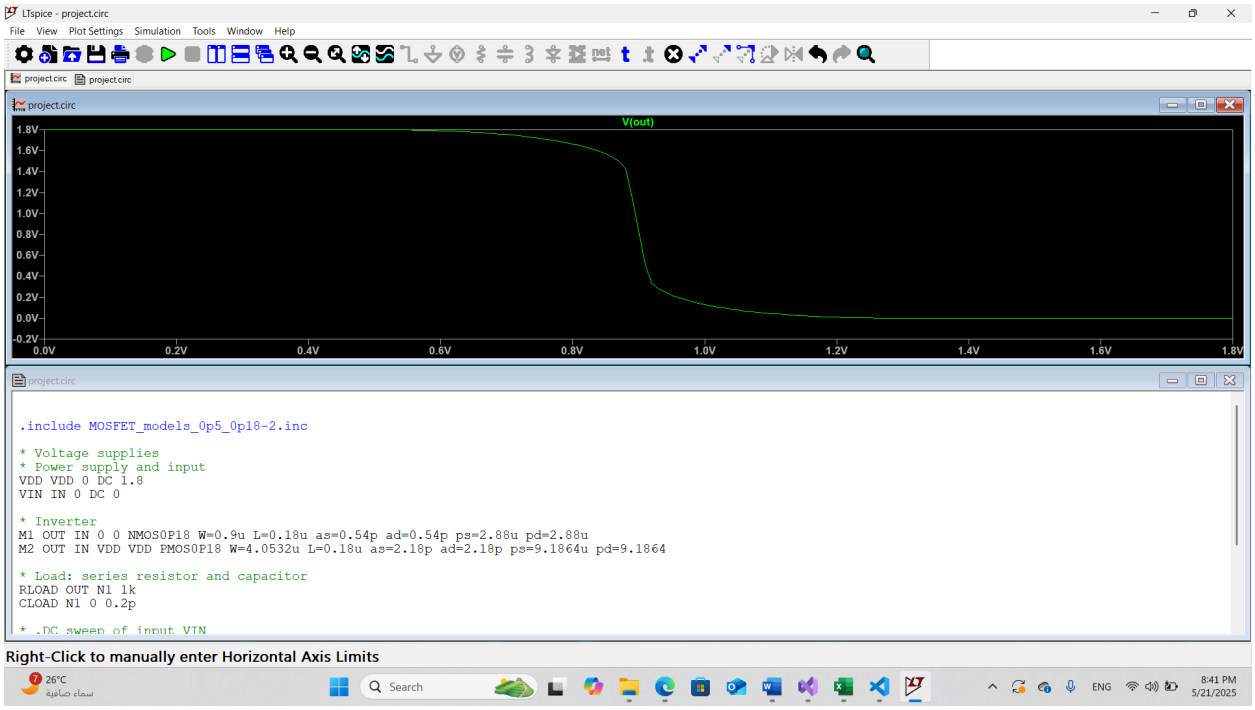
Approaches and Results

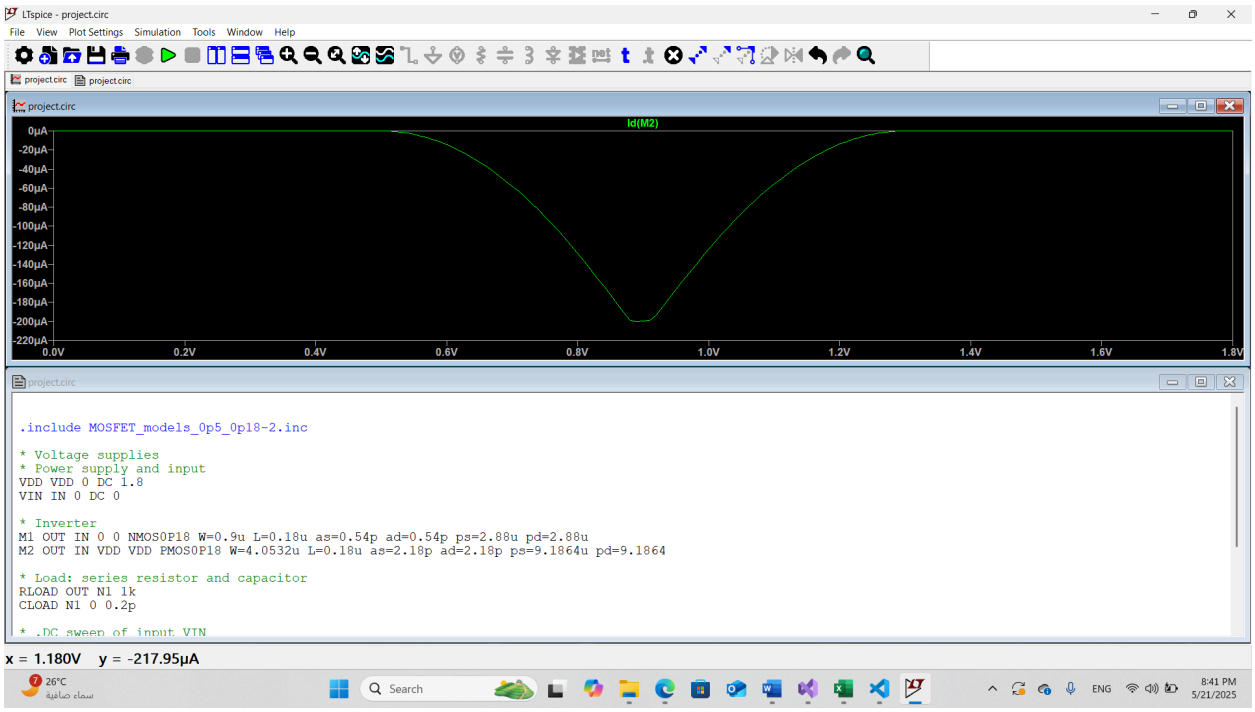
Let's go through each of the questions in the report.

Part A:

1) For both part a and b, we have the width of the NMOS given, in addition to the technology length, which is treated as the length of both the NMOS and PMOS. So, the only unknown quantity is the width of the PMOS. For part a), we will need to solve for it, but for b), it's just the same as that of part a to guarantee the minimum area. X1 is just 3 * Lmin so it's 3 * 0.18 = 0.54, which is also equal to X2. The other parameters will depend on the width values, which we will explore now.

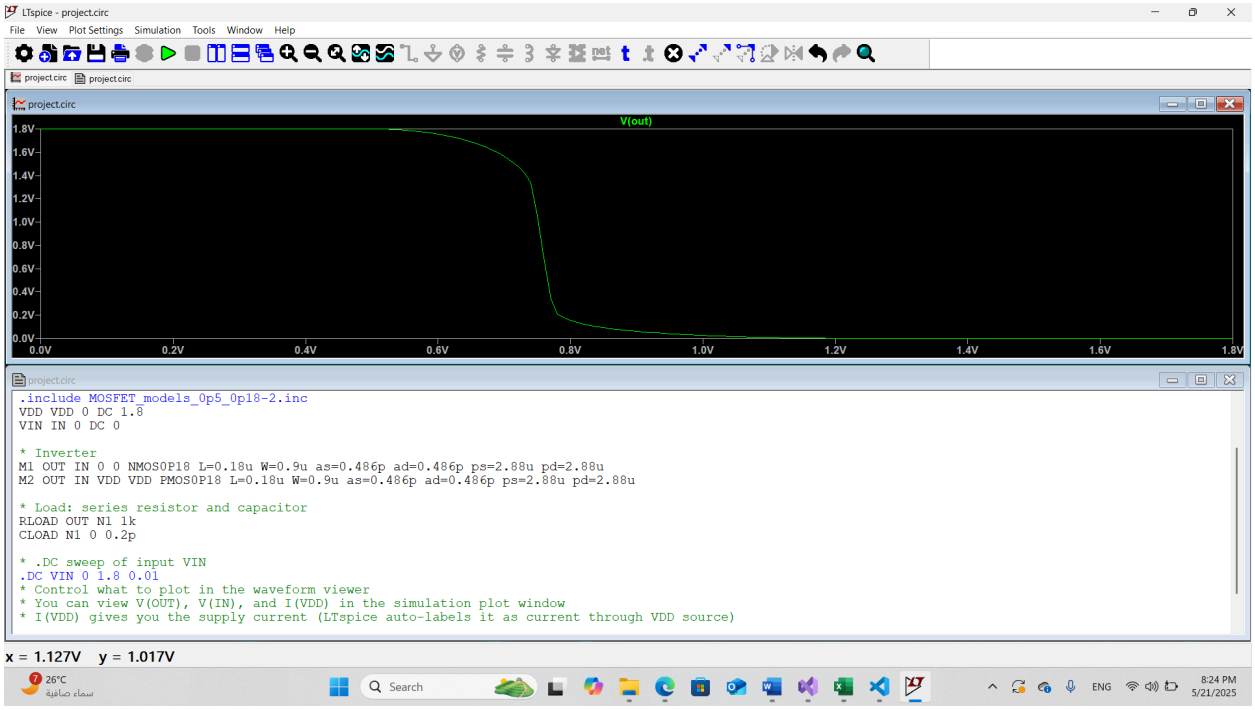
a) Since we want a matched design, this means that kn = kp. This is the same as saying that we want k'n * (W/L)n = k'p * (W/L)p. We have provided above that the width and length of the NMOS are given, so is the length of the PMOS. Further, we have k'n and k'p as well as k'n = Cox * mu_n = 3.81 * 10^(-10) A/V^2 and k'p = Cox * mu_p = 8.46 * 10^(-11) A/V^2. Now, it's just a matter of substitution and solving for the width of the PMOS. This gets us the width of the PMOS as 4.0532 micrometers. From this, we can compute As, Ap, Ps, and Pd for both the NMOS and PMOS. As is just X1 * W, once using the W of the NMOS (0.9) and the other time that of the PMOS (4.0532). Ad is just the same as As. For Ps, we do 2X1 + 2W once again one time substituting 0.9 for W (NMOS) and the other 4.0532 (PMOS). After that, we just implement the netlist that does this, which is attached below. We create both the VDD and Vin based on the appropriate syntax. Then, we appropriately assign both the NMOS and the PMOS. We connect the resistor and capacitor in series, and then do a sweep for the Vin to see the inversion effect (at the start the output is high but as VIN increases the output decreases). Attached below as well is the simulation graph for both Vout vs Vin (the VTC characteristic) and I out vs Vin; in our case I out is called id (M2), which corresponds to the current in the drain of the PMOS, which is also the same as Id (VDD).

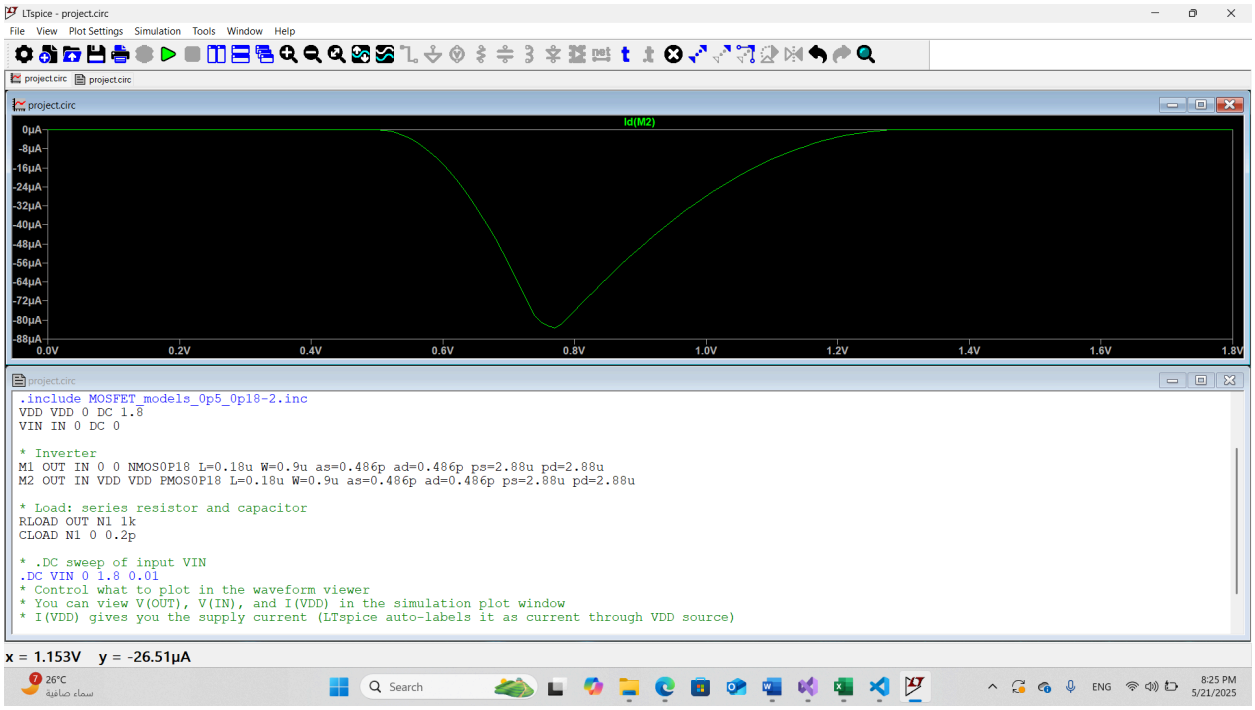




b) In regards to working with minimal area assumption, it will be much simpler as the width and length of both transistors will just be 0.9 and 0.18 respectively, thus all the other design parameters (As, Ad, Ps, Pd)

will be exactly like those of the NMOS in the previous part. As we can see, almost everything is the same as part a, but the difference is only in the PMOS, which just takes the exact same values of the NMOS. Attached below are screenshots of the VTC and Output current. The VTC and Output current curves are almost identical to those produced in part a.

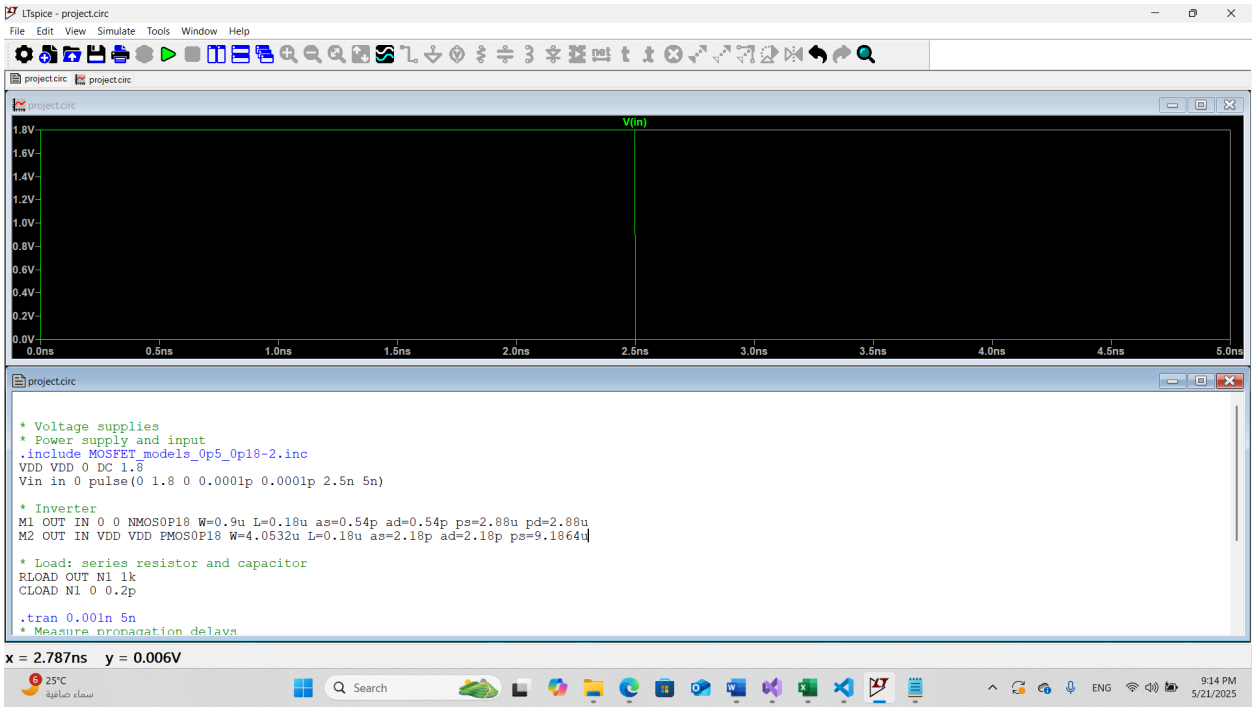




2) For this problem, we decided to divide it into two parts, solving it once for the case of matched design and the other for minimal area.

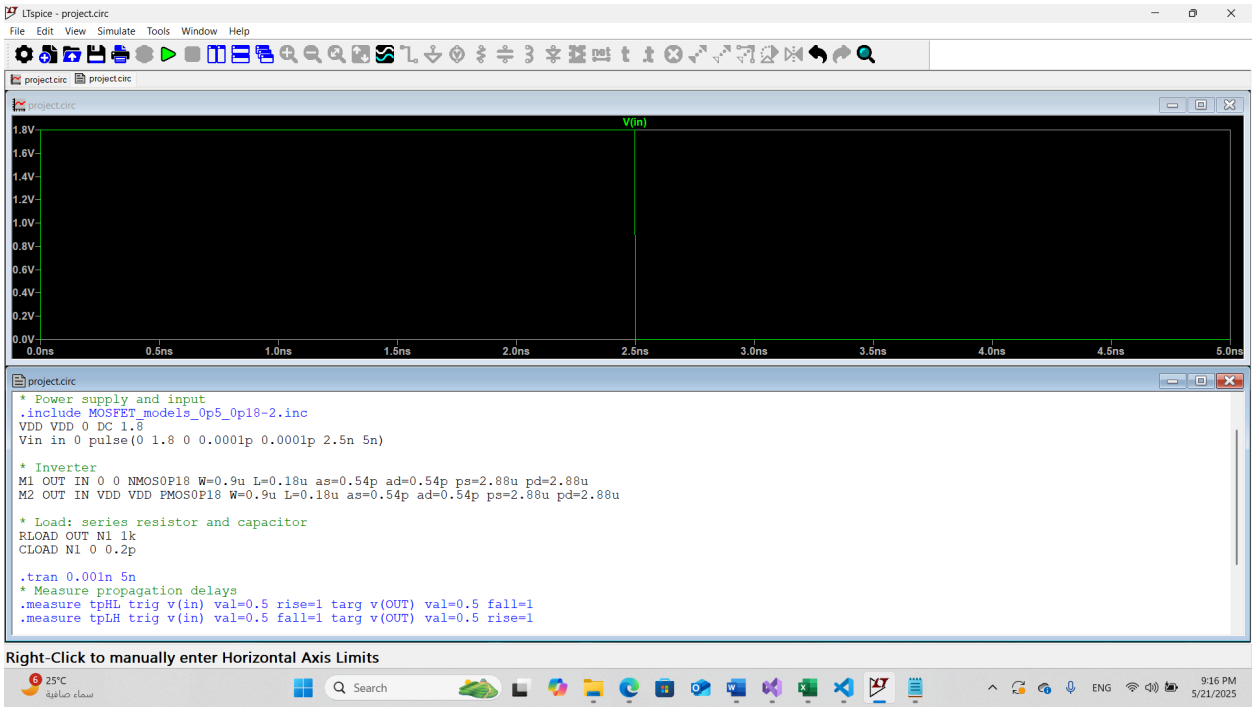
   a) In the case of matched design, we just took the netlist of 1a, and added the pulse update, transient simulation, and propagation delay calculation sections. As we can see, the Vin is now a pulse signal instead of a DC signal. In regards to the values of its parameters, the 0 and the 1.8 are just the range of values we want Vin to take, which is the exact same as the previous question, just in pulse format now. As for the other parameters, we selected them in such a way that we could attain an input square wave without getting invalid propagation delays. In regards to the latter, sometimes when the Tdel and Tr were exactly 0, the propagation delay wouldn't be calculated. So we tried to make them as small as possible in order to register a number, though in theory, one could continue to add more and more zeros after the decimal point. We

decided to let the fall time be 2.5n, while the time it stays on, Tw, is left at 5n, which we believe are the smallest possible values. The transient simulation goes from 0.001n to 5n, and after that we measure both the tpHL and tpLH. Attached below is the output of the Vin and the propagation delays. As we can see, Vin is almost a pure square wave. The propagation delays are both almost zero.



Propagation Delay:
tphl=1.53444217314e-11 FROM 2.77777789347e-17 TO 1.53444495092e-11
tplh=2.47399665544e-12 FROM 2.50000017222e-09 TO 2.50247416888e-09

b) For the minimum area case, we will do the same thing as part a, but with the minimum area code. As we can see it is just the code of 1a with the necessary additions. Also attached is the input waveform shape and the propagation delays. Both are almost the same as part a (matched design).



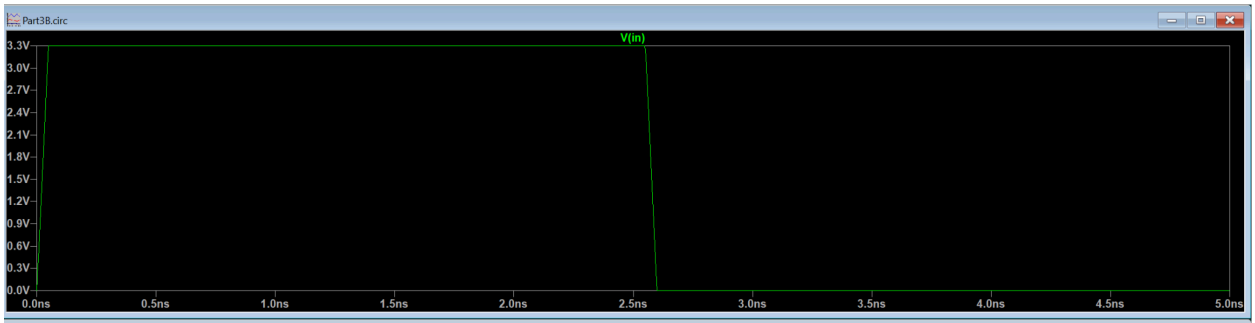
Propagation Delay:
tphl=1.2920325536e-11 FROM 2.77777773717e-17 TO 1.29203533138e-11
tplh=1.34295733673e-11 FROM 2.50000017222e-09 TO 2.51342974559e-09

Comment on propagation delay difference between a and b:
The matched design shows asymmetrical delays, this shows an imbalanced pull-up network (PUN) vs. pull-down network (PDN) unlike the minimum area design which shows more balanced delays. The matched design additionally showed faster low to high switching but slower high to low switching. This might be due to the imbalance between the PUN and the PDN where the PUN is stronger.

3) For this problem, we decided to divide it into 2 parts, one time solving it under the assumption that the transistor has matched design and the other time under the assumption that the transistor has minimum area. We swap the mosfet model from the 0.18 one to the 0.5 one. Similarly we do the same computations that we did in 1. So in this case, X1= 3*Lmin = 3* 0.5 = 1.5. Similarly, we

compute the other values. For this problem in general, we decided to just get the output of the input wave and the delay rather than getting the VTC and Iout vs Vin.

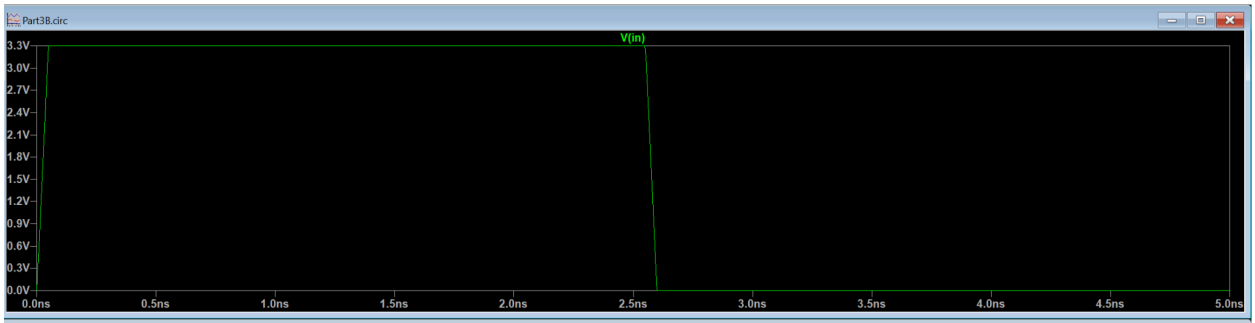
a) For the case of matched design, we first start by computing the the width of the PMOS. We will use the same values for k'n and k'p that were computed in question 1. We will use 0.5 as the new length for both the Nmos and Pmos and 1.25 as the new width for the Nmos. Solving the equations gives us the width of the pmos as 5.6294u. We will then compute the As = Ad according using As = Ad = X1 * W once for the NMOS and once for the PMOS. After that, we will then compute the Ps = Pd = 2X1 * 2W once for the NMOS and once for the PMOS. We update the respective values of Vdd and CLOAD. We decided to increase the Tdel and Tr a bit in this case to see the difference in the visualization as compared to part a, so as we'll see in the output, it will no longer look as square as before in question 2. Attached below is the input waveform and the propagation delays. As we can see the wave is not as square as before, but the propagation delay is still almost very small.




```
Part B number 3 (minimum area case) :
* Voltage supplies
* Power supply and input
.include MOSFET_models_0p5_0p18-2.inc
VDD VDD 0 DC 3.3
Vin in 0 PULSE(0 3.3 0 50p 50p 2.5n 5n)


* Inverter
M1 OUT IN 0 0 NMOS0P5 W=1.25u L=0.5u as=1.875p ad=1.875p ps=5.5u pd=5.5u
M2 OUT IN VDD VDD PMOS0P5 W=1.25u L=0.5u as=1.875p ad=1.875p ps=5.5u pd=5.5u

* Load: series resistor and capacitor
RLOAD OUT N1 1k
CLOAD N1 0 0.5p

.tran 0.001n 5n
* Measure propagation delays
.measure tpHL trig v(in) val=0.5 rise=1 targ v(OUT) val=0.5 fall=1
.measure tpLH trig v(in) val=0.5 fall=1 targ v(OUT) val=0.5 rise=1
```

tphl=9.12399958903e-10 FROM 7.57575749813e-12 TO 9.19975716401e-10
tplh=-6.05947508856e-13 FROM 2.5924242425e-09 TO 2.59181829499e-09

b) For the minimum area case, we once again just made the dimensions of the PMOS exactly like those of the NMOS. The logic and values relating to the pulse, transient, and propagation delays remain the same in the netlist. Attached below is input voltage waveform and the propagation delays. Like in the matched design case, the input waveform is not a perfect square and the propagation delays are almost negligible.



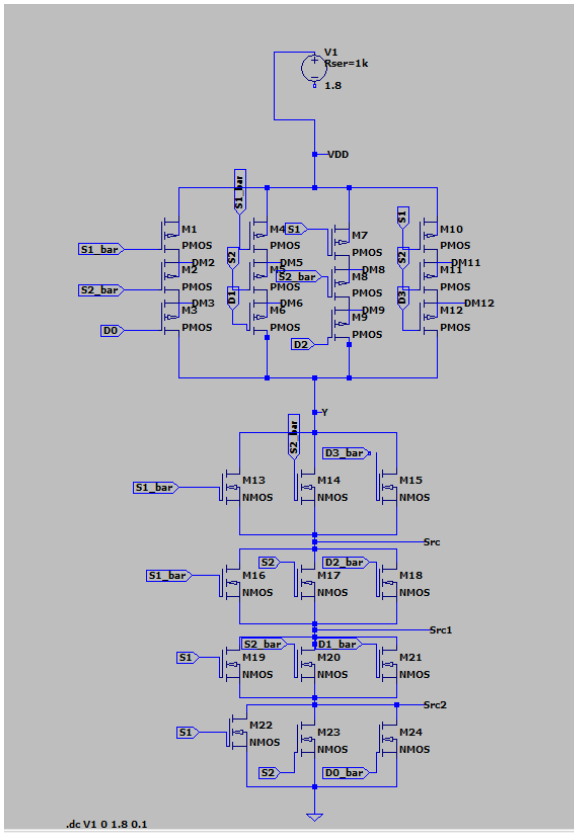
Part3B.circ

```
Part B number 3 (minimum area case) :
* Voltage supplies
* Power supply and input
.include MOSFET_models_0p5_0p18-2.inc
VDD VDD 0 DC 3.3
Vin in 0 PULSE(0 3.3 0 50p 50p 2.5n 5n)


* Inverter
M1 OUT IN 0 0 NMOS0P5 W=1.25u L=0.5u as=1.875p ad=1.875p ps=5.5u pd=5.5u
M2 OUT IN VDD VDD PMOS0P5 W=1.25u L=0.5u as=1.875p ad=1.875p ps=5.5u pd=5.5u

* Load: series resistor and capacitor
RLOAD OUT N1 1k
CLOAD N1 0 0.5p

.tran 0.001n 5n
* Measure propagation delays
.measure tpHL trig v(in) val=0.5 rise=1 targ v(OUT) val=0.5 fall=1
.measure tpLH trig v(in) val=0.5 fall=1 targ v(OUT) val=0.5 rise=1
```

tphl=9.12399958903e-10 FROM 7.57575749813e-12 TO 9.19975716401e-10
tplh=-6.05947508856e-13 FROM 2.5924242425e-09 TO 2.59181829499e-09

Part B:

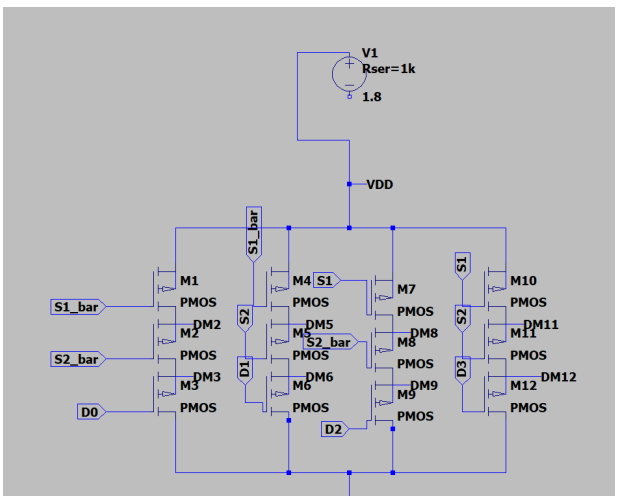
4) Overall Schematic:



The CMOS circuit was designed by splitting it into a Pull-Up Network and a Pull-Down Network. The PUN boolean expression was first obtained from the truth table of a 4-1 MUX. The complement of Y was then obtained using De Morgan's Law to construct the boolean expression of the PDN.

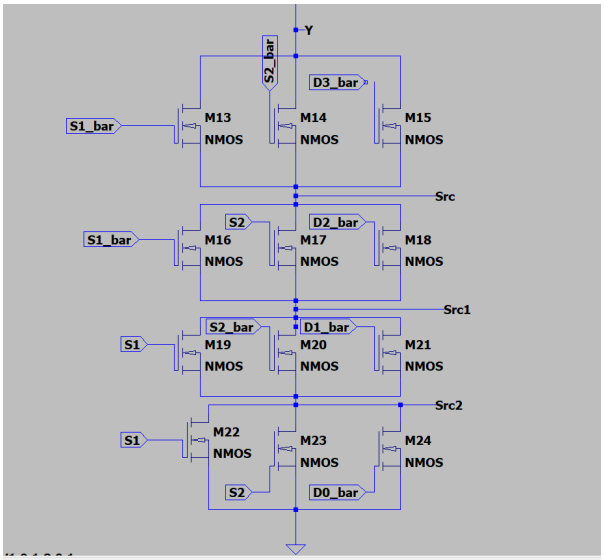
Pull-Up Network:

$Y = (D0 \cdot \bar{S}1 \cdot \bar{S}2) + (D1 \cdot \bar{S}1 \cdot S2) + (D2 \cdot S1 \cdot \bar{S}2) + (D3 \cdot S1 \cdot S2)$



Pull-Down Network

$\bar{Y} = [\bar{D}0 + S1 + S2] \cdot [\bar{D}1 + S1 + \bar{S}2] \cdot [\bar{D}2 + \bar{S}1 + S2] \cdot [\bar{D}3 + \bar{S}1 + \bar{S}2]$



The number of transistors required to build this circuit was 24 (excluding the inverters). Since each expression has 12 inputs, and as each variable is assumed to be connected to a distinct gate, 24 transistors were used. As we also used inverted inputs and connect them to some gates, 6 additional transistors were used. This time, we assumed inverters made of a single NMOS gate with a pull up resistor for simplicity.

5) To implement the netlist of the schematic, inverter circuits were first added to obtain the complement of all the inputs which are part of the boolean expression. The PUN and PDN were then described using 0.18u technology assuming a matched design with p=2n and n=5. Thus, the width of the NMOS is 5*0.18 and the width of the PMOS is 5*0.18*2.

   DC inputs were additionally used to test the full truth table (4 test cases) of the 4-1 MUX. The test cases demonstrated that when the selection lines inputted correspond to an input (D0-D3), the output (Y) follows that input. For example, if S1 and S2 are at voltage low, the output will follow D0 meaning it will output VDD if D0 is voltage high and 0 if D0 is voltage low.

The test cases were as follows:

Test Case 1:

VDD VDD 0 1.8
Vin_S1 S1 0 0
Vin_S2 S2 0 0
Vin_D0 D0 0 1
Vin_D1 D1 0 0
Vin_D2 D2 0 0
Vin_D3 D3 0 0

Test Case 2:

VDD VDD 0 1.8
Vin_S1 S1 0 0
Vin_S2 S2 0 1.8
Vin_D0 D0 0 0
Vin_D1 D1 0 1.2
Vin_D2 D2 0 0
Vin_D3 D3 0 0

Test Case 3:

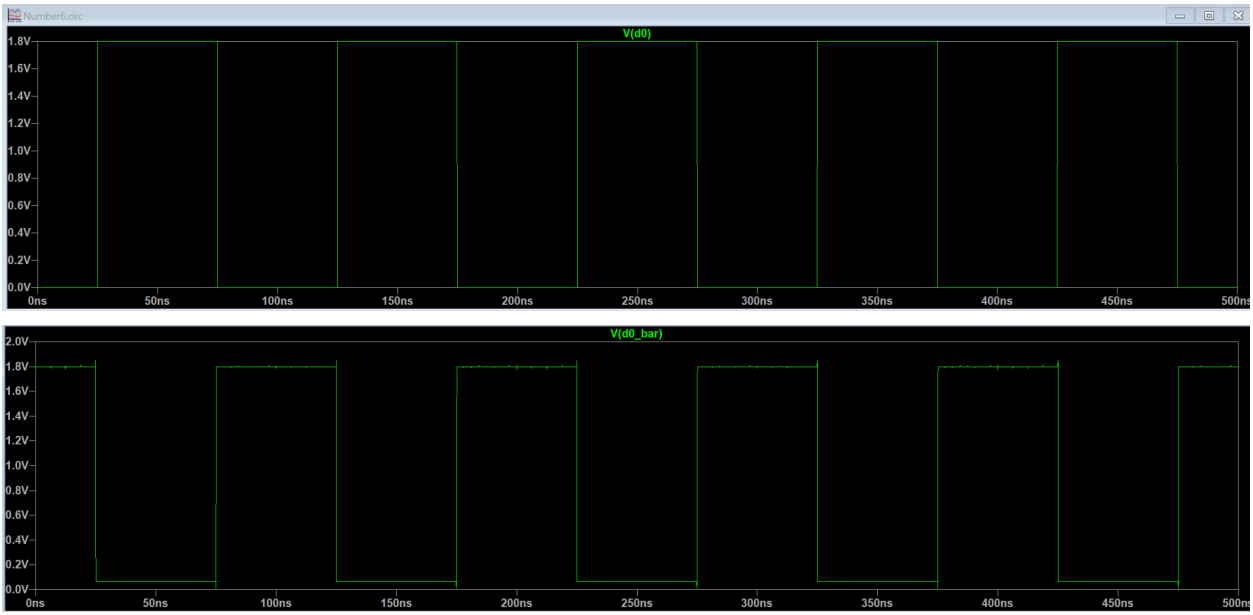
VDD VDD 0 1.8
Vin_S1 S1 0 1.8
Vin_S2 S2 0 0
Vin_D0 D0 0 0
Vin_D1 D1 0 0
Vin_D2 D2 0 1
Vin_D3 D3 0 0

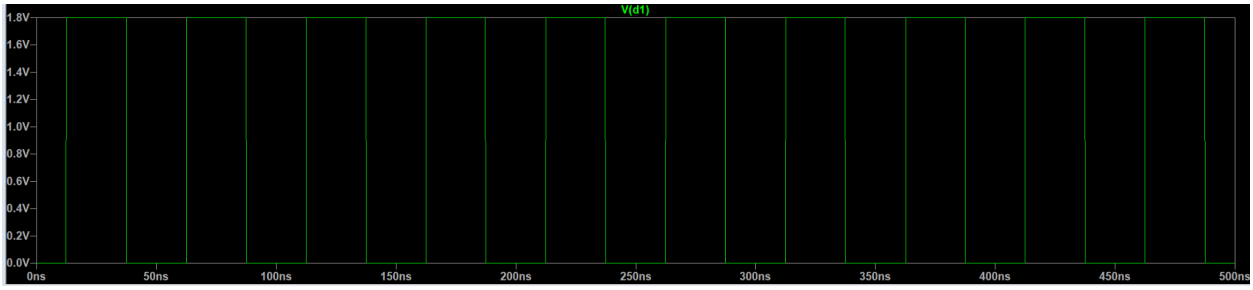
Test Case 4:

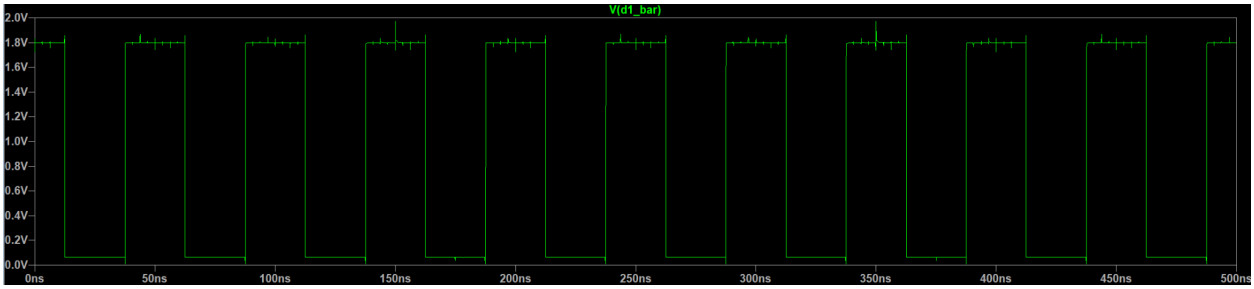
VDD VDD 0 1.8
Vin_S1 S1 0 1.8
Vin_S2 S2 0 1.8
Vin_D0 D0 0 0
Vin_D1 D1 0 0
Vin_D2 D2 0 0
Vin_D3 D3 0 1

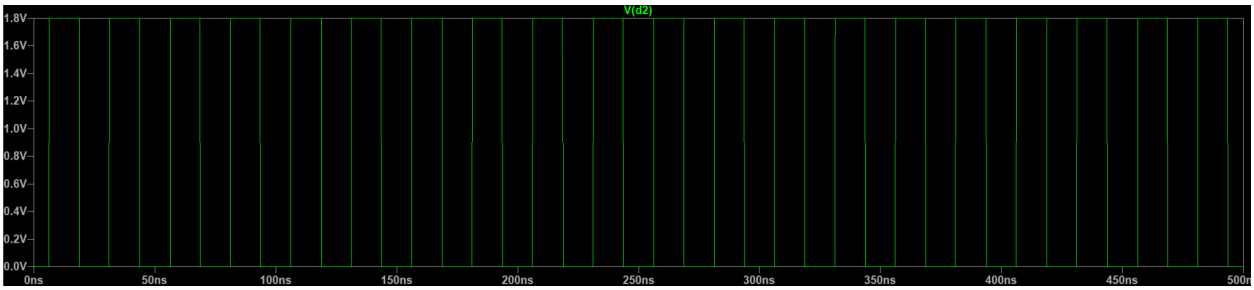
The output of all the above test cases was 1.8V as expected. Each test case was repeated one more time with the selected input being at voltage low.

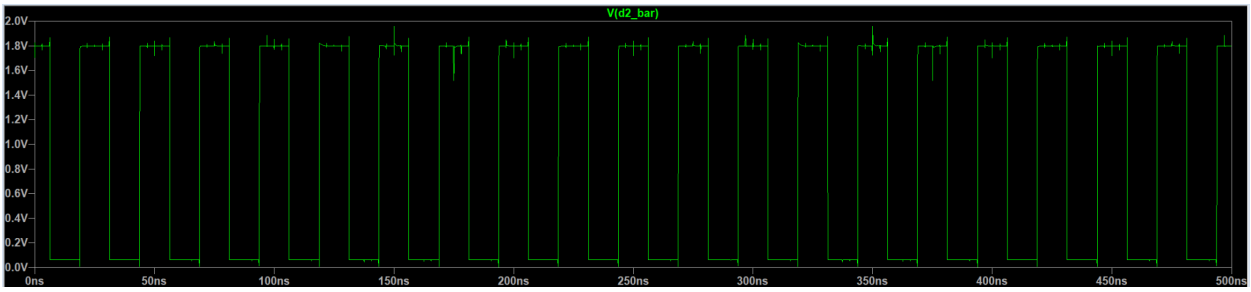
6) For this question, the netlist from number 5 was taken and the pulse and transient parts were added, similar to how Part A question 3 was conducted. Attached below are all images of all the input and output values. The input output values presented are D0,D1,D2,D3, S1, S2, and their inverses. In regards to D0, it's just a normal pulse signal and D0_bar is just its inverse. However as we can see in the images that D0 is a very slow signal while D1 is faster than it. Similarly D2 is even faster than D1, and D3 is even faster than D2. The reason why these signals become faster (faster frequency or even simpler shorter periods) is because of the updated parameter values in the pulse list. For example, D0 has the following parameters "Vin_D0 D0 0 PULSE(0 1.8 25n 10p 10p 50n 100n)" while D1 has the following parameter "Vin_D1 D1 0 PULSE(0 1.8 12.5n 10p 10p 25n 50n)". In fact, the parameters get halved from each data signal to the one after it when it comes to some of the parameters like Tf and Tw, thus causing the frequency to double. The reason why we do that is to create a binary counting pattern, which allows for proper testing of the 4 possible combinations of the 4x1 multiplexer. As for S1 and S2, they are both fixed and oscillate at the same rate so their graphs are almost the same except that S2 has a Tdel of 50n instead of 0 for S1, creating a sort of shift in the graph. As for Vy and Vy(load), we can see that they are almost the same, except that Vy(load) has much less noise compared to Vy. We hypothesize that this is because of the presence of the resistor and capacitor, which isolate some of the noise.

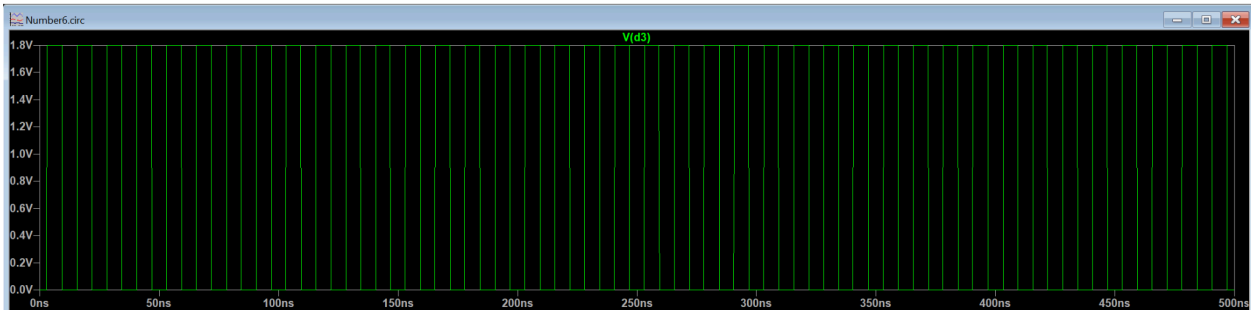


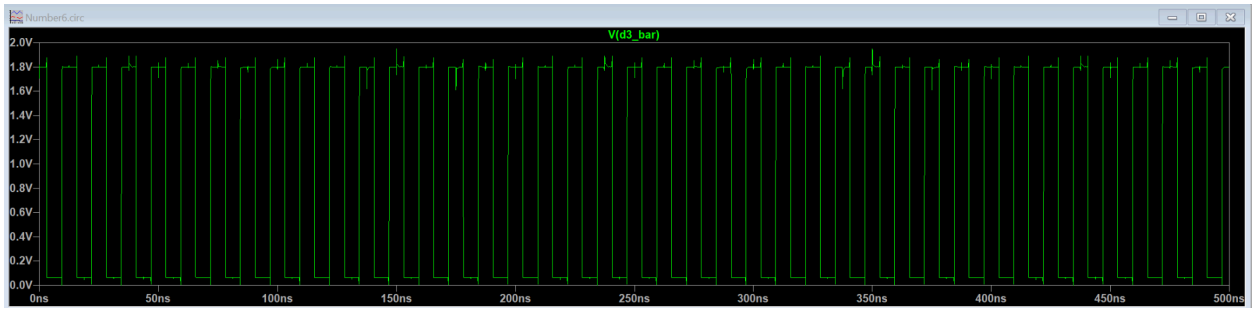

V(d1)
1.8V
1.6V
1.4V
1.2V
1.0V
0.8V
0.6V
0.4V
0.2V
0.0V
0ns
50ns
100ns
150ns
200ns
250ns
300ns
350ns
400ns
450ns
500ns

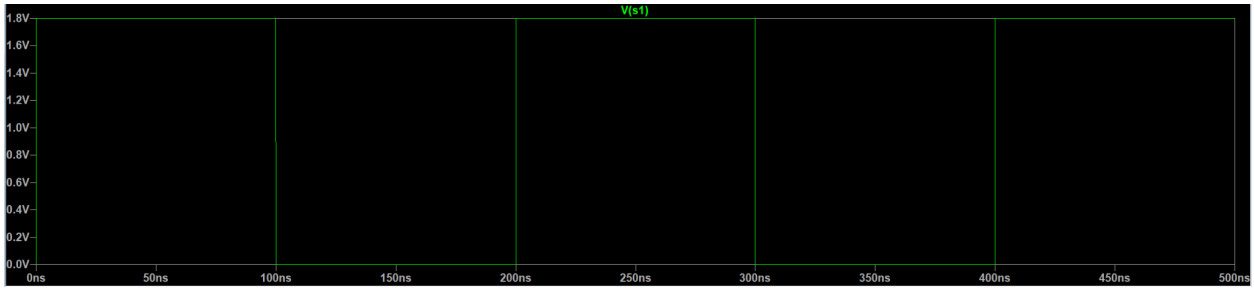

V(d1_bar)
2.0V
1.8V
1.6V
1.4V
1.2V
1.0V
0.8V
0.6V
0.4V
0.2V
0.0V
0ns
50ns
100ns
150ns
200ns
250ns
300ns
350ns
400ns
450ns
500ns

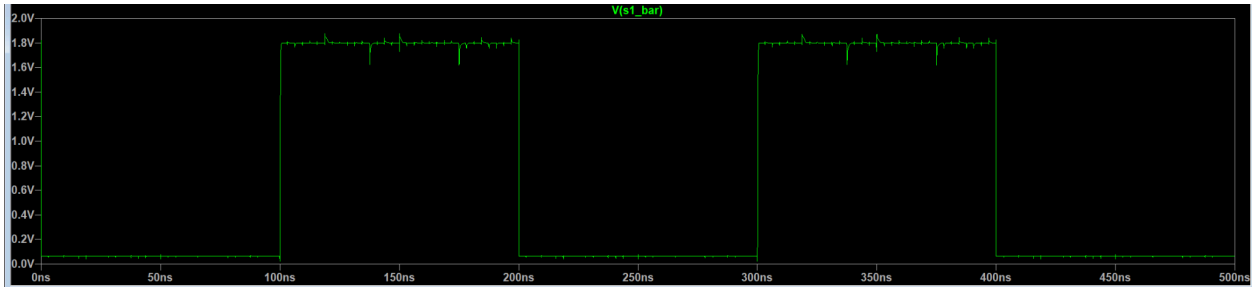

V(d2)
1.8V
1.6V
1.4V
1.2V
1.0V
0.8V
0.6V
0.4V
0.2V
0.0V
0ns
50ns
100ns
150ns
200ns
250ns
300ns
350ns
400ns
450ns
500n

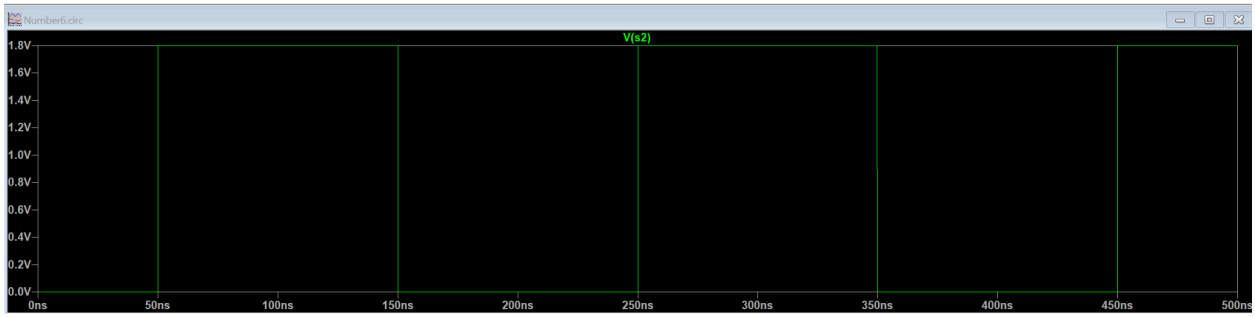

V(d2_bar)
2.0V
1.8V
1.6V
1.4V
1.2V
1.0V
0.8V
0.6V
0.4V
0.2V
0.0V
0ns
50ns
100ns
150ns
200ns
250ns
300ns
350ns
400ns
450ns
500ns


Number6.circ
V(d3)
1.8V
1.6V
1.4V
1.2V
1.0V
0.8V
0.6V
0.4V
0.2V
0.0V
0ns
50ns
100ns
150ns
200ns
250ns
300ns
350ns
400ns
450ns
500ns

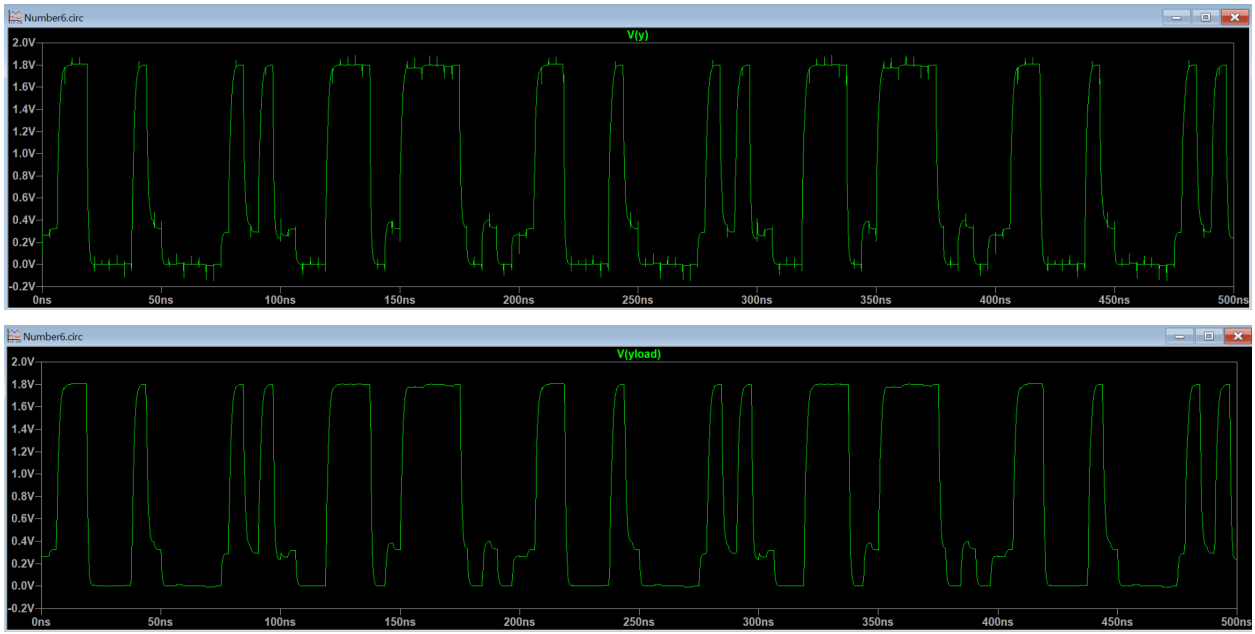

Number6.circ
V(d3_bar)
2.0V
1.8V
1.6V
1.4V
1.2V
1.0V
0.8V
0.6V
0.4V
0.2V
0.0V
0ns
50ns
100ns
150ns
200ns
250ns
300ns
350ns
400ns
450ns
500ns


V(s1)
1.8V
1.6V
1.4V
1.2V
1.0V
0.8V
0.6V
0.4V
0.2V
0.0V
0ns
50ns
100ns
150ns
200ns
250ns
300ns
350ns
400ns
450ns
500ns


V(s1_bar)
2.0V
1.8V
1.6V
1.4V
1.2V
1.0V
0.8V
0.6V
0.4V
0.2V
0.0V
0ns
50ns
100ns
150ns
200ns
250ns
300ns
350ns
400ns
450ns
500ns


Number6.circ
V(s2)
1.8V
1.6V
1.4V
1.2V
1.0V
0.8V
0.6V
0.4V
0.2V
0.0V
0ns
50ns
100ns
150ns
200ns
250ns
300ns
350ns
400ns
450ns
500ns


V(s2_bar)
2.0V
1.8V
1.6V
1.4V
1.2V
1.0V
0.8V
0.6V
0.4V
0.2V
0.0V
0ns
50ns
100ns
150ns
200ns
250ns
300ns
350ns
400ns
450ns
500ns



7) The average power dissipated and the propagation delays were all logged using the .measure command. Please note that the propagation delays were measured whenever a different input was selected. This means that when D0 was selected and the output value changed, the propagation delays were measured and so on for D0-D3.

   The following values were recorded:

   iavg: AVG(I(VDD))=-0.000572615448991 FROM 1e-07 TO 4e-07
   pavg: -Iavg*1.8=0.00103070780818
   tplh_d0=1.30637910789e-08 FROM 2.50049999999e-08 TO 3.80687910788e-08
   tphl_d0=-3.079293013e-08 FROM 7.5015e-08 TO 4.422206987e-08
   tplh_d1=-2.15251596426e-08 FROM 1.12505e-07 TO 9.09798403574e-08
   tphl_d1=-4.04081599359e-08 FROM 1.37515e-07 TO 9.71068400641e-08
   tplh_d2=2.5304893661e-08 FROM 1.81255e-07 TO 2.06559893661e-07
   tphl_d2=-1.83351307106e-08 FROM 1.93765e-07 TO 1.75429869289e-07
   tplh_d3=3.78554832598e-08 FROM 2.4063e-07 TO 2.7848548326e-07
   tphl_d3=5.04231183816e-08 FROM 2.3439e-07 TO 2.84813118382e-07
   tplh_s1=7.84804834364e-08 FROM 5.00000010973e-12 TO 7.84854834365e-08

tphl_s1=-1.52018816171e-08 FROM 1.00015e-07 TO 8.4813118383e-08

Contribution and Conclusion

In this project, the individual contributions of each person were as follows:

- Together: Question 1 and 2
- Ismaiel: Question 3, Question 6, Report part A section entirely
- Habiba: Question 4, Question 5, Question 7, Report part B section entirely

In this project, we set out to implement a simple inverter circuit and then a 4x1 multiplexer using LTSpice as hopefully revealed in this report.