Luan Vo (30553600)

ECE 558 – VLSI

Lab 1

----------Step 1 – Truth table NAND gate inputs S, and B with output W:--------

S	B	W
0	0	1
0	1	1
1	0	1
1	1	0

----------Step 2 – Create schematics:----------

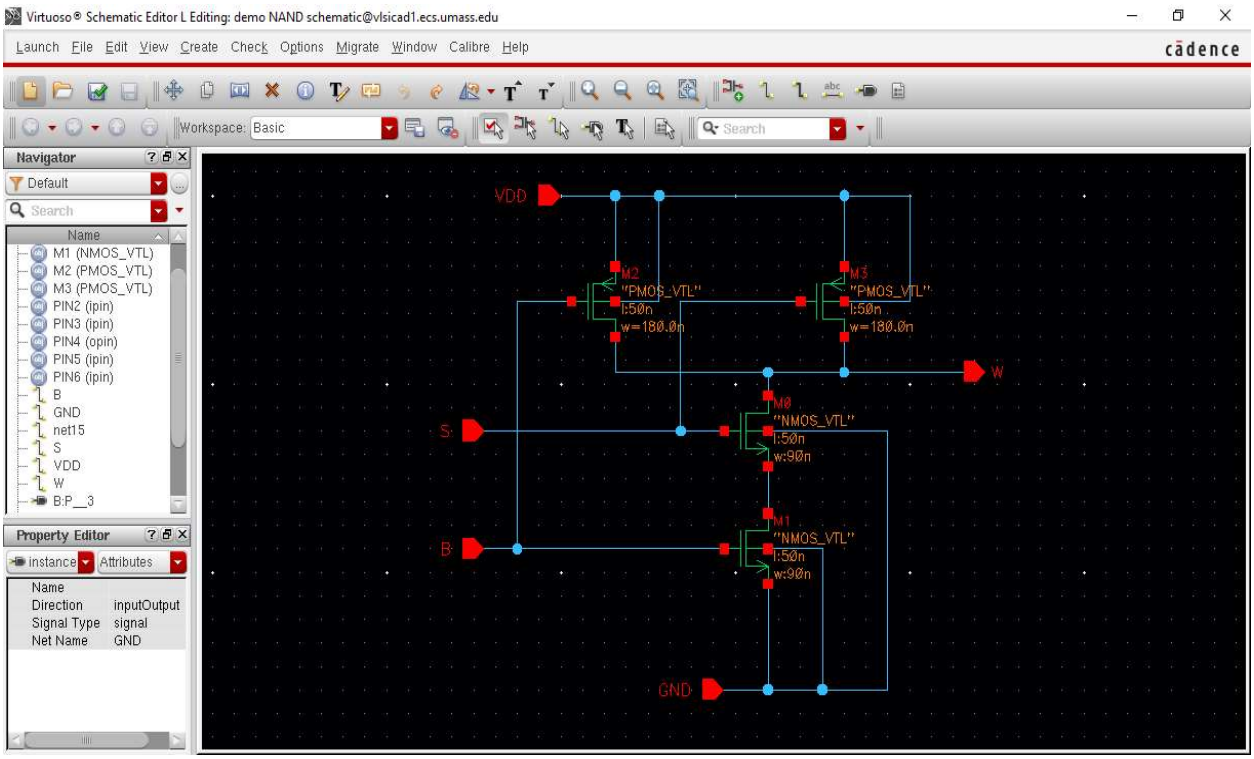
Length = 50nm

Lambda = 25nm

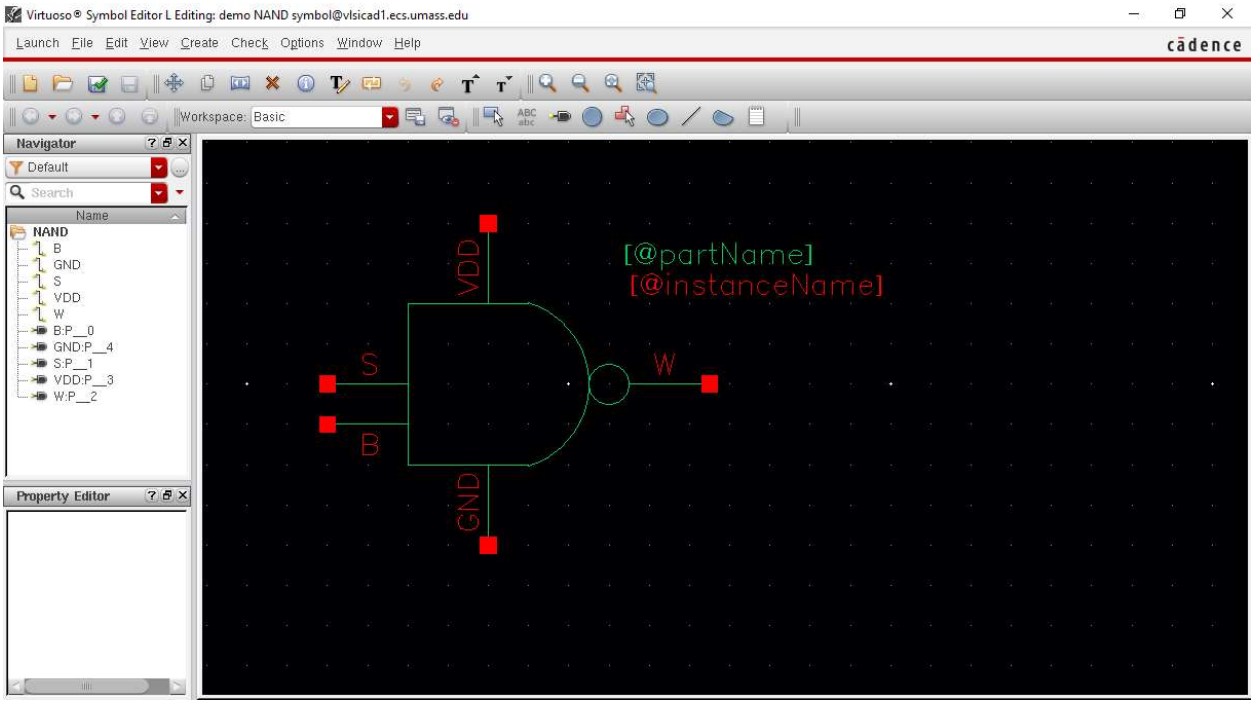
Beta = 2 (P/N width ratio) => PMOS_width = 2*NMOS_width

NMOS_width = 90nm => PMOS_width = 180nm

Transistor Level Schematic:

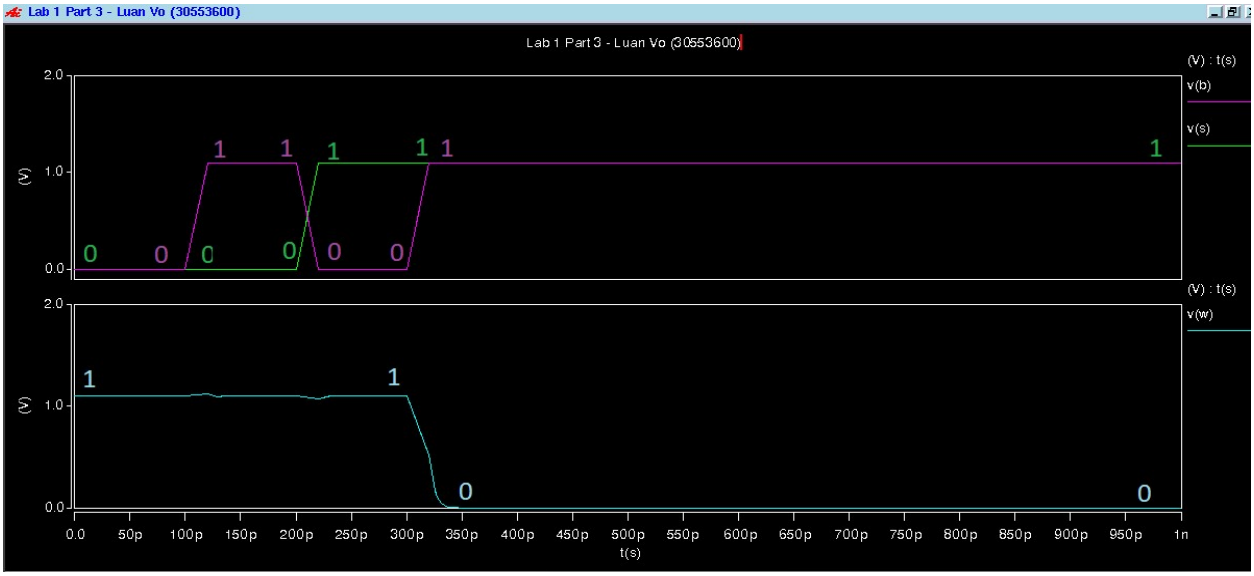


Symbol Level Schematic:



----------Part 3 – Cscope and Hspice:----------

Graph of logic output of NAND gate with no loads nor inverters (1 cycle of 4 inputs):



The vector model was managed so that it will have the follow effects for V(s) and V(b):

Trise = Tfall = 30ps

Tperiod = 100ps

Inputs S, and W in order of: 00, 01, 10, 11

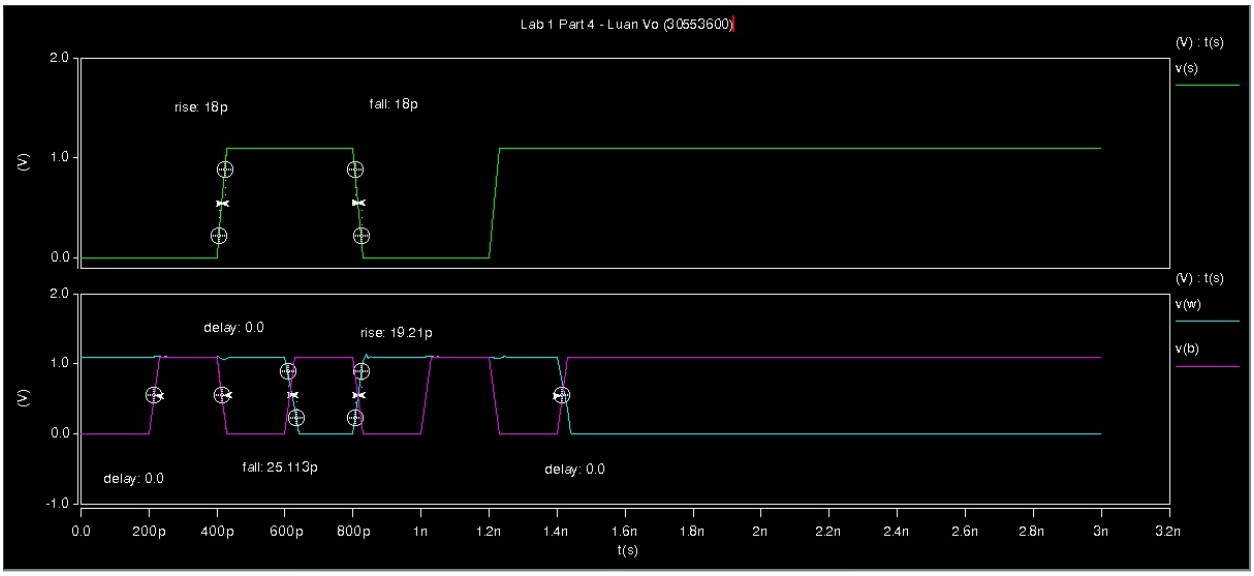
Examine the graph, we can see that, whenever S and B are not at high at the same time, W will output high value. This can be seen from 0 – 300ps, which coincides with the first 3 periods of 00, 01 and 10 that all are supposed to output 1s at W. After 300ps, both S and B are at high value, thus reducing W back to its low value. The truth table of the Cscope can be shown below:

V(s)	V(b)	V(w)
0	0	1
0	1	1
1	0	1
1	1	0

Therefore, the schematic works and it is correct, as this table is equal to that of the actual truth table of NAND gate which is included in part 1.

----------Part 4 – Worst-case Transition:----------

General graph of edges of NAND gate with no loads nor inverters (2 cycles of 8 inputs):



The times where one input from V(s,b) that cause output V(w) to change are:

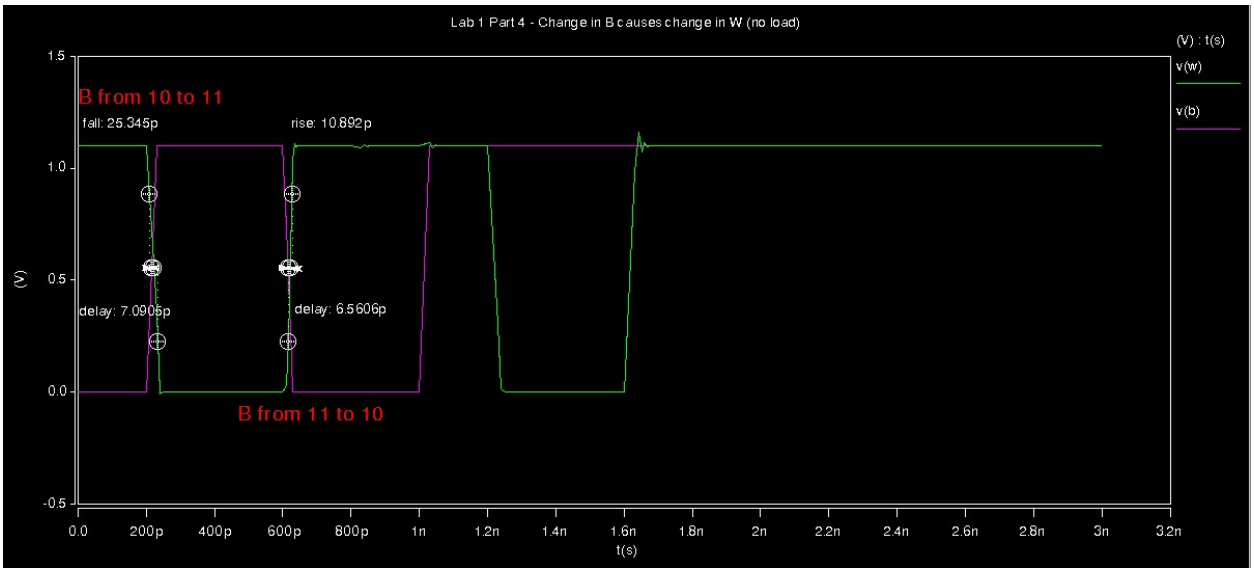
10 to 11 | 01 to 11 | 11 to 10 | 11 to 01

By measuring the output change via one bit input change, we can determine the worst rise/fall.

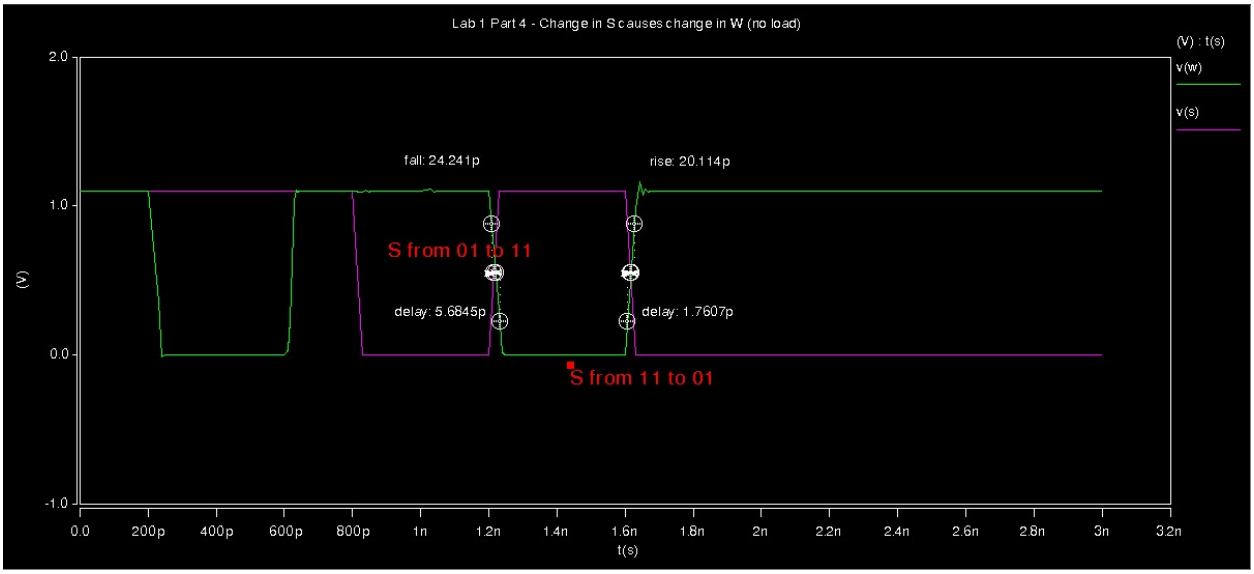
Trise = Tfall = 30 ps

Tperiod = 300 ps (modified to 300ps since edges change from 20-30ps, 300 instead of 150 for better resolution)

Graph of change in B causes W to change (10 to 11 and 11 to 10):



Graph of change in S causes W to change (01 to 11 and 11 to 01):



My observation of the resulting waveform (i.e: V(w)) yielded that the fall and rise measured with how fast it takes for the waveform reach from 80% to 20% and vice versa. This is due to 80%-20% range is defined as the steady state of the transition. Delay measured at edges 50% crossing. The time where delay could be measured are when the FET performs the switching operation, because 3 out of 4 states are when the FET is holding its value.

This table shows the worst-case transition delay of each input bits. This input sequence will cover all cases: 10,11,11,10,00, 01,11,11,01 (where 00 is used reset the delay exists in 10-01-11). This is to make sure neither S, B nor W are changing to quickly or at the same time.

Edge	Input (SB) or V(s,b)	Output (W) or V(w)	Time of output (ps)	Output delay vs inputs	Time (ps)
Tfall (S)	01 to 11	1 to 0	24.241	Tpdf	5.685
Tfall (B)	10 to 11	1 to 0	25.345	Tpdf	7.09
Trise (B)	11 to 10	0 to 1	10.892	Tpdr	6.56
Trise (S)	11 to 01	0 to 1	20.114	Tpdr	1.761

Note: Tfall(S) denotes V(w) falling edge triggered by change in S bit, and so on.

----------Part 5 – Performance analysis with loading:----------

In this part, we will load 16 inverter and Cload parallelly with W. The input waveform will be V(s,b) and resulting waveform is V(w). Observation will be made with 80%-20% rise/fall and 50% crossing.

V(s,b) properties: Trise = Tfall = 30ps | Tperiod = 300ps

We will call the 16 inverters and Cload – being connected parallelly to NAND – the buffer sub circuit. The following table will compare the propagation wave at V(s,b) and V(w). however, we will use only V(b) to show the comparison for 2 reasons: once V(b) switches V(w) also switches, this is due to the transition from 10 to 11 and 11 to 00. Both cases require V(b) to change, whereas V(s) only changes once. This makes the step of comparing edge delay easier if we use V(b) as the indicator.

Compare with Part 4: (Similar 1 input change causes output to change)

The delay time is much higher when compared to part 4, this is because the inverters offset the signal by a bit, combined that offset with the loaded capacitance and that will result in the rise in Tdr and Tdf. Due to signal propagates on a longer “road” and every device is not exactly synchronous. In addition, it is worth noting that it is possible to reduce the rise/fall time by making the period in .vec file larger, this will try and match up the operation frequencies to the frequencies of the capacitance and parasitic capacitance.

Even though the input stays the same – for both edges, V(w) does not. This can be observed from the fall time is longer than rise time for V(w). This is likely due to the capacitance from 16 inverters do not have the same charge and discharge rate.

Comparison Table:

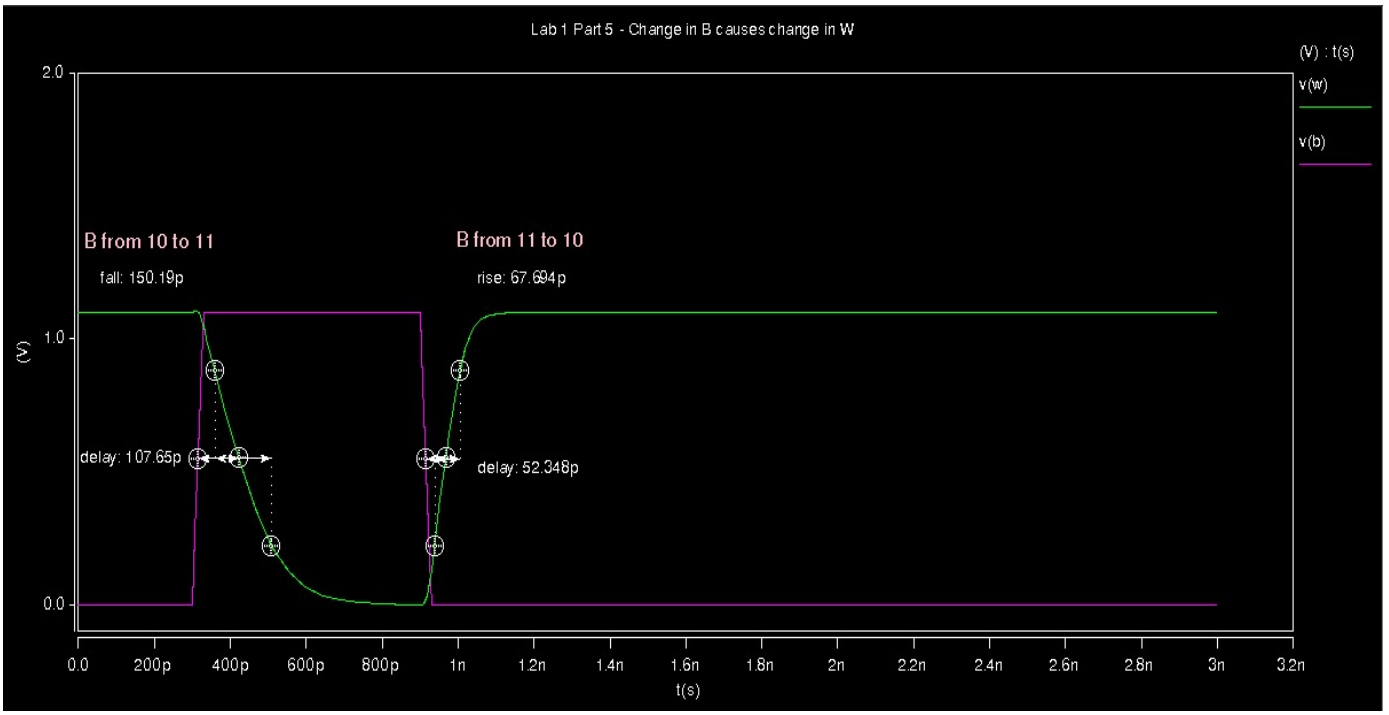
Part 5 – with Load	Input (SB) or V(s,b)	Output (W) or V(w)	Time of output (ps)	Output delay vs inputs	Time (ps)
Tfall (S)	01 to 11	1 to 0	150.13	Tpdf	106.79
Tfall (B)	10 to 11	1 to 0	150.19	Tpdf	107.65
Trise (B)	11 to 10	0 to 1	67.694	Tpdr	52.348
Trise (S)	11 to 01	0 to 1	66.529	Tpdr	51.571
Power	Watt (W) absolute value				
P(static)	5.06572e-07				
P(dyn)	2.888e-03				
Part6 - Resizing	Input (SB) or V(s,b)	Output (W) or V(w)	Time of output (ps)	Output delay vs inputs	Time (ps)
Tfall (S)	01 to 11	1 to 0	108.1	Tpdf	75.989
Tfall (B)	10 to 11	1 to 0	107.19	Tpdf	75.665
Trise (B)	11 to 10	0 to 1	48.568	Tpdr	38.613
Trise (S)	11 to 01	0 to 1	47.048	Tpdr	38.244
Power	Watt (W) absolute value				
P(static)	0.989e-06				
P(dyn)	5.906e-03				
Part8 - Parasitic	Input (SB) or V(s,b)	Output (W) or V(w)	Time of output (ps)	Output delay vs inputs	Time (ps)
Tfall (S)	01 to 11	1 to 0	167.75	Tpdf	115.67
Tfall (B)	10 to 11	1 to 0	165.34	Tpdf	113.69
Trise (B)	11 to 10	0 to 1	74.298	Tpdr	53.698
Trise (S)	11 to 01	0 to 1	76.444	Tpdr	55.201
Power	Watt (W) absolute value				
P(static)	4.613e-07				
P(dyn)	2.893e-03				

Note: Tfall(S) denotes V(w) falling edge triggered by change in S bit, and so on.

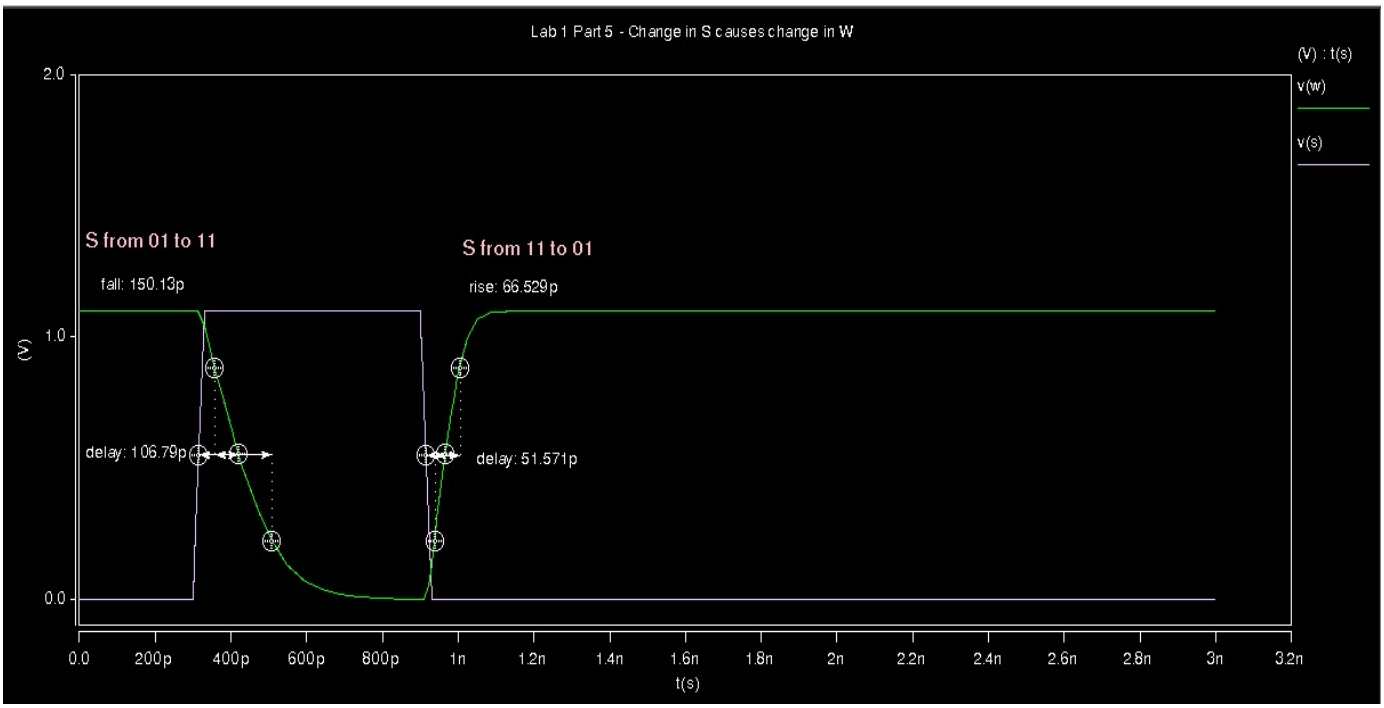
Part 5 graphs of Single Input Change Cause Output Change with Load

Input pattern: for B – 10,11,11,10 | for S – 01,11,11,01

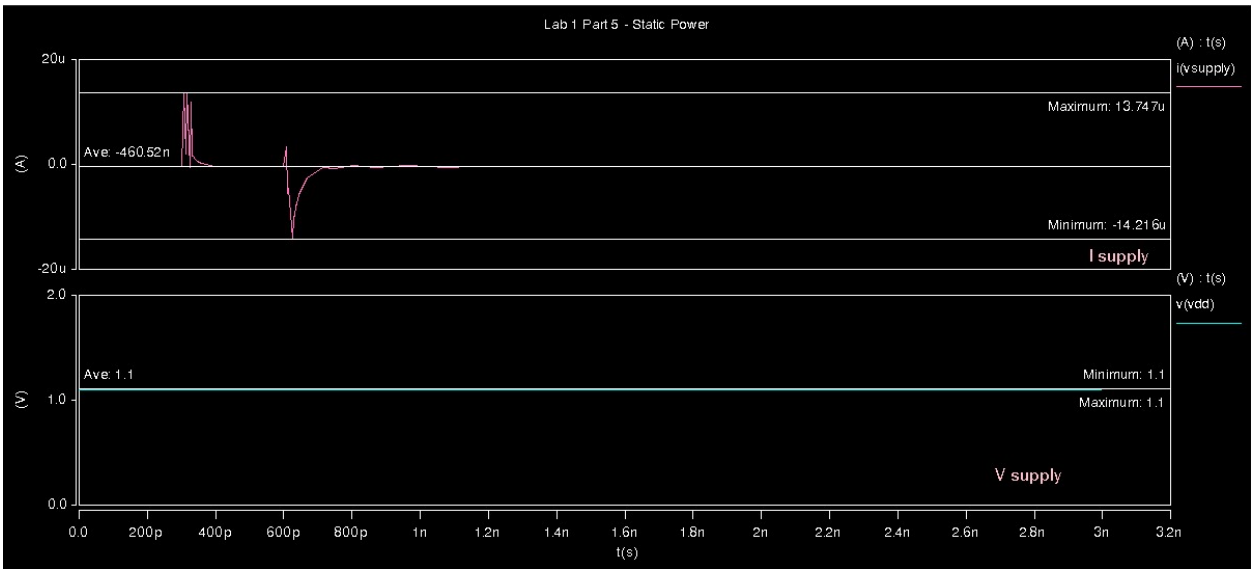
Graph of change in B causes W to change (10 to 11 and 11 to 10) with load:



Graph of change in S causes W to change (01 to 11 and 11 to 01) with load:



Power Graph of circuit when inputs states are 00, 01 and 10 (static values):

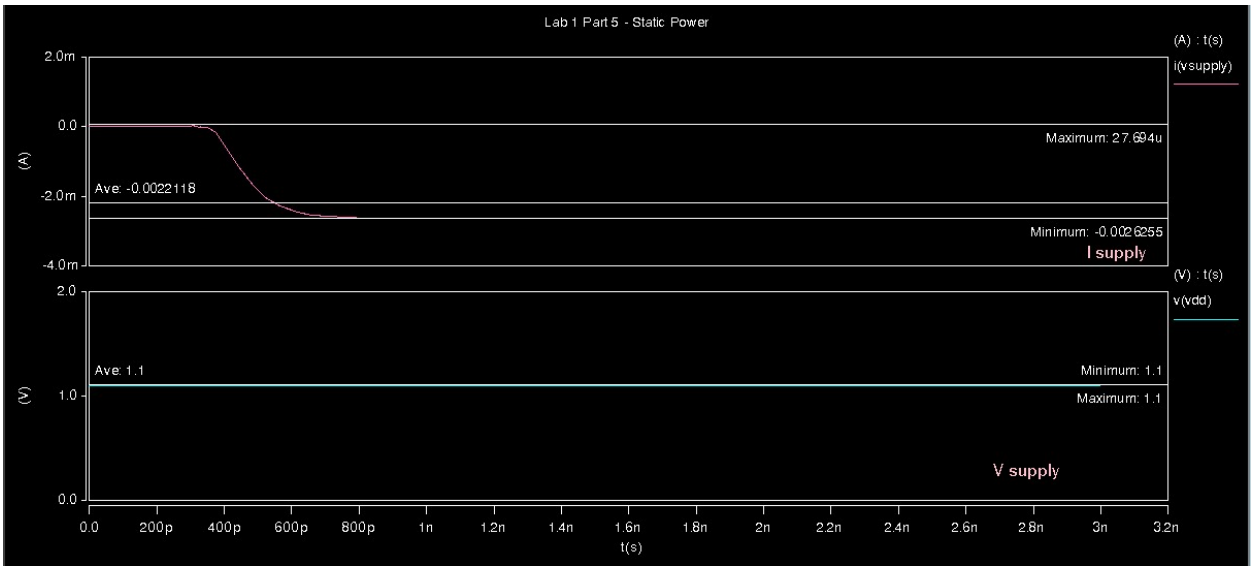



```
$DATA1 SOURCE='HSPICE' VERSION='E-2010.12-SP2 32-BIT'
.TITLE '** generated for: hspiced'
 avgpower          maxpower          temper            alter#
  5.724e-07         2.512e-05           25.0000         1
```

The states that will result in power leakage are 00, 01 and 10, as these states do not change V(out), thus the FET holds the value in causing the result that is P(static). This can be computed by graphing the plots of FET with all the inputs from 3 states and P(stat) = I-avg-leakage * Vdd
This can also be done through hspice coding with:
.measure tran avgpower AVG power from=1ps to=3000ps
Static/leakage power (graphical) = I-avg * VDD = -460.52n * 1.1 = -5.06572e-07 W
Static/leakage power (.measure) = I-avg * VDD = avgpower = 5.724e-07 W
Once again, we see that the graphical is closely equal to the hspice simulation. Therefore, this is the reason why I believe this is the correct method of computing the static power.
This method will include all leakage path as the only time leakage happens are when V(w) holds its output, i.e: during 00 to 01, 01 to 10. Which is why we see that hspice measure gives off a bit more power than our calculation, this is because I simultaneously compute the avg power of all 3 states at the same time. Even though V(out) never changes during these states, the FETs from

earlier parts W and NAND is still being switched, so the power is a little higher than if it were just by themselves.

Power Graph of circuit behavior when inputs change from 00 to 11 (switching):




```
$DATA1 SOURCE='HSPICE' VERSION='E-2010.12-SP2 32-BIT'
.TITLE '** generated for: hspiced'
 avgpower         maxpower         temper           alter#
  2.436e-03        2.888e-03          25.0000        1
```

The simulated power was gotten from coding in the spice script to allow simulator to output peak and avg power. As Peak Pdyn is the peak power when FET is in operation (switching), this is done when the inputs switch from 11 to 00 as it will make V(out) change from 0 to 1 (as shown in graph). This is the only time the V(out) ever switches during the 4 states. So peak power = I_switching-peak*Vdd (note here that peak is negative since it draws out power from supply).

.measure tran maxpower MAX power from=1ps to=3000ps

Peak dynamic power (graphical) = I_switching-peak * VDD = -0.0026251 * 1.1 = -2.888e-03 W

Peak dynamic power (.measure) = I-max * VDD = maxpower = 2.88e-03 W

There are almost no differences between the hspice simulation and cscope graphical method.

----------Part 6 – Resizing ----------

Student's ID: 30553600 => Resizing = 2+0 = 2. Therefore, every transistors' width is doubled.

NMOS: L = 50 nm | W = 180 nm

PMOS: L = 50 nm | W = 360 nm

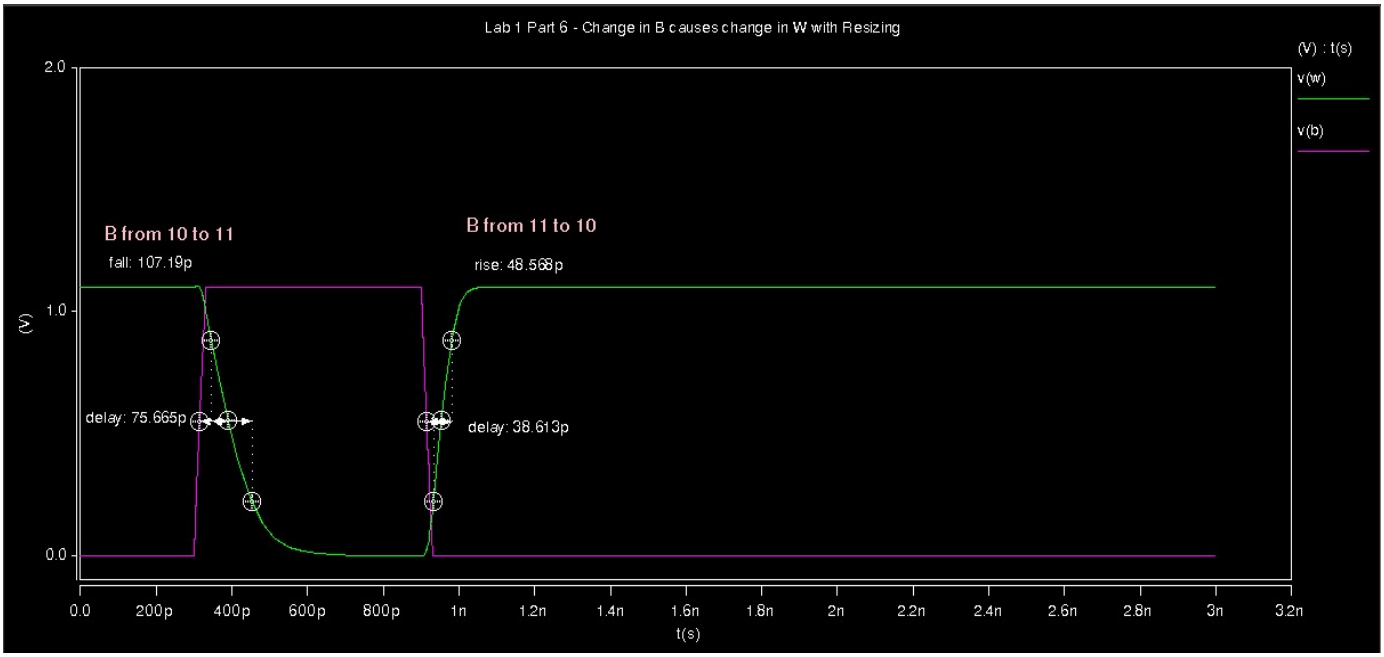
Results are appended to table in Part 5, refer to part 5 for table.

Part 6 Graphs of Single Input Change Cause Output Change with Load and Resizing

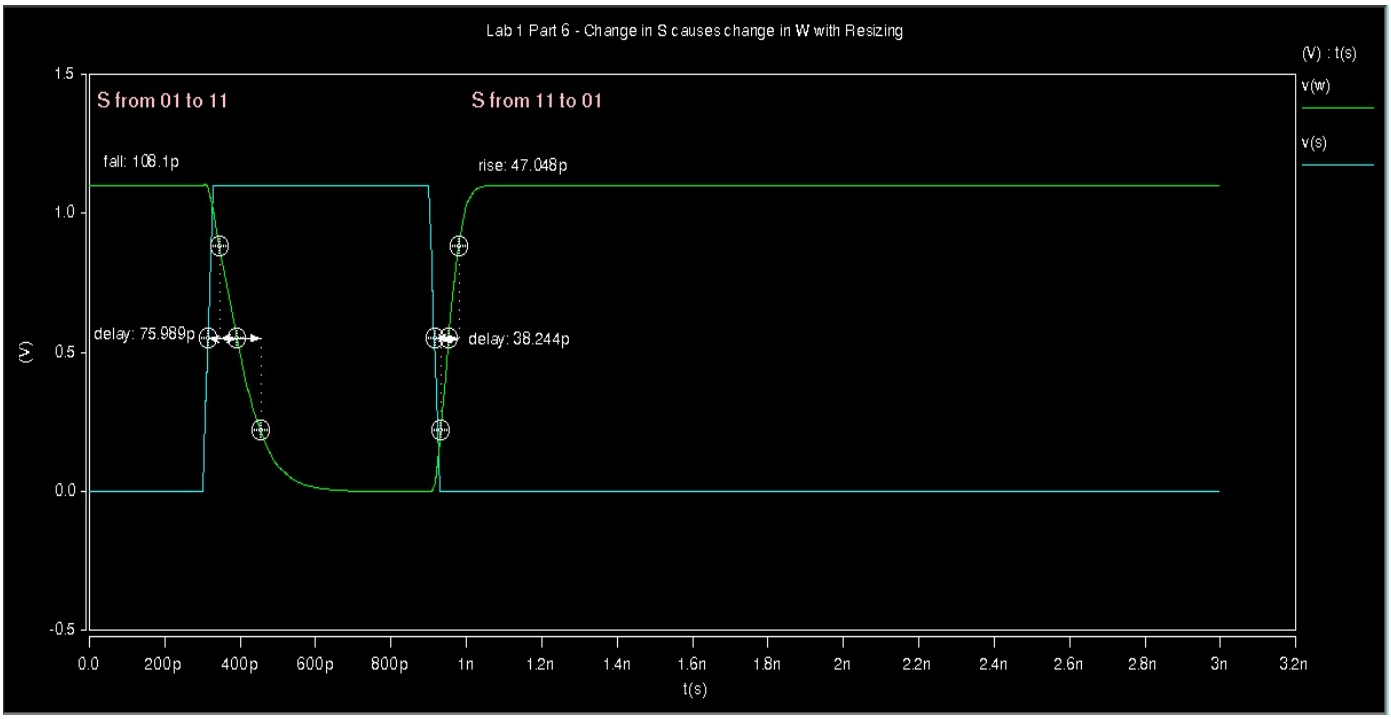
Input pattern: for B – 10,11,11,10 | for S – 01,11,11,01

This pattern is similar to part 4 and 5

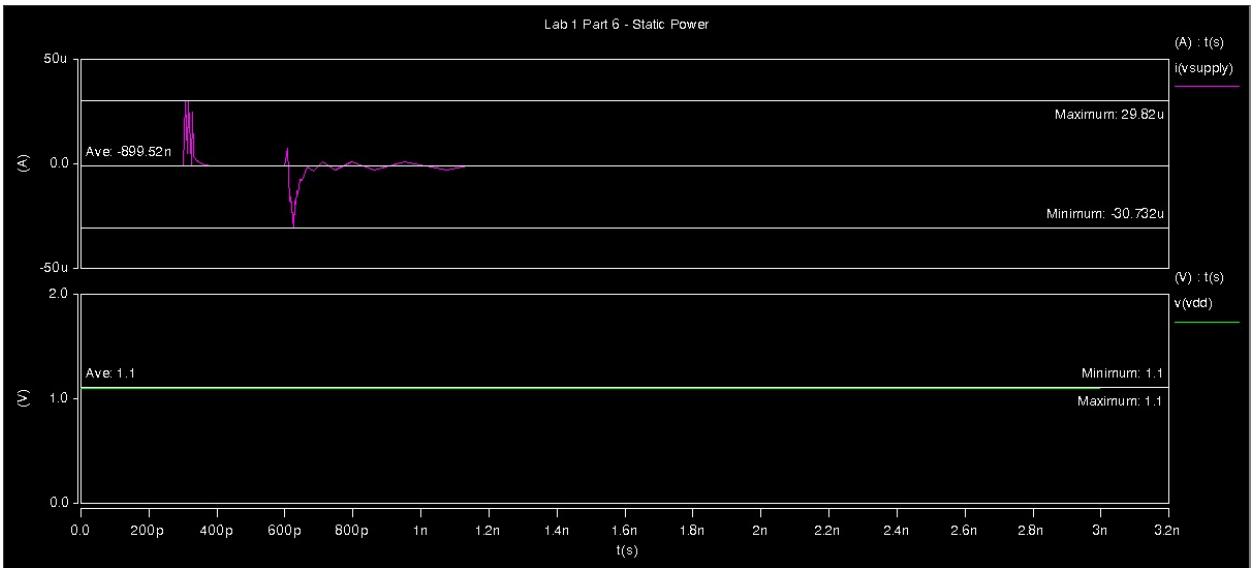
Graph of change in B causes W to change (10 to 11 and 11 to 10) with load and resizing:



Graph of change in S causes W to change (01 to 11 and 11 to 01) with load and resizing:



Power Graph of circuit after resizing and inputs states are 00, 01 and 10 (static values):

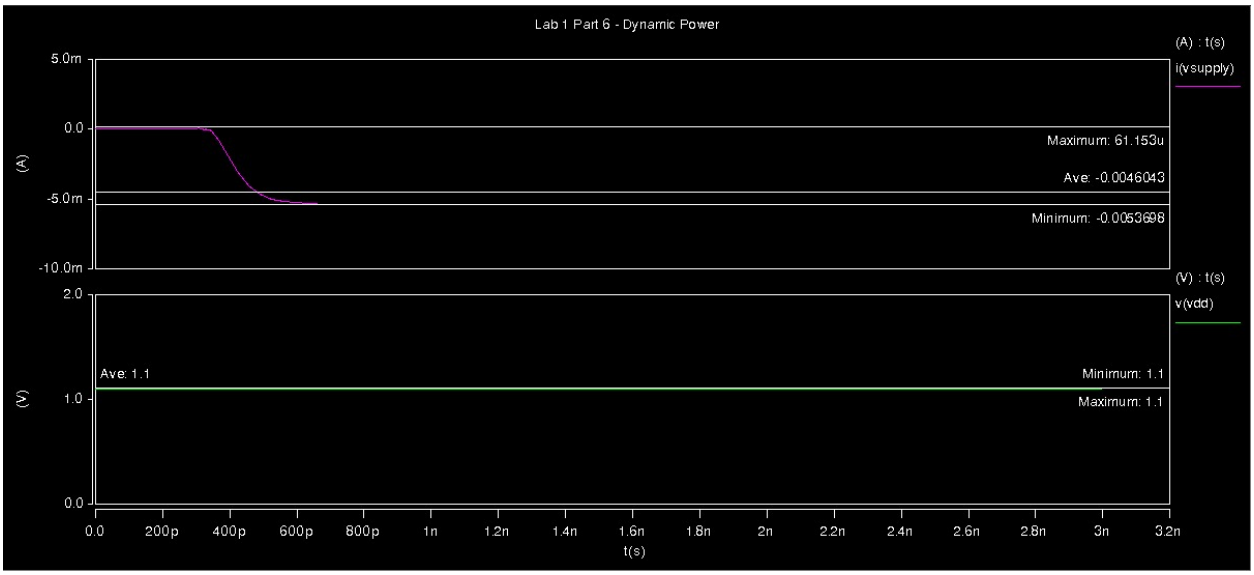



```
$DATA1 SOURCE='HSPICE' VERSION='E-2010.12-SP2 32-BIT'
.TITLE '** generated for: hspiced'
 avgpower         maxpower         temper           alter#
  1.147e-06        4.831e-05          25.0000        1
```

P static graphical = I_static-avg * VDD = -899.52n * 1.1 = -0.989e-06 W = -0.989 uW

P static simulated = I-avg * VDD = avgpower = 1.147e-06 W

Power Graph of circuit behavior after resizing and inputs change from 00 to 11 (switching):



```
$DATA1 SOURCE='HSPICE' VERSION='E-2010.12-SP2 32-BIT'
.TITLE '** generated for: hspiced'
 avgpower         maxpower         temper           alter#
  5.069e-03        5.906e-03          25.0000        1
```

P dynamic = I_switching-peak * VDD = -0.0053692 * 1.1 = -5.9061e-03 W = -5.906 mW

P dynamic = I-peak * VDD = maxpower = 5.906e-03 W = 5.906 mW

Comparing results to part 5:

Interesting observations are made when observing the time between part 5 and 6, we see that all Trise, Tfall, Tdr and Tdf of part 6 are lower than that of part 5. Which means that after resizing MOSFETs – specifically by increasing the width by 2 – we have decrease edge transition and delay time. This would make sense because Id = uCox * (W/2L) * (Veff)^2, it is evident that increasing width while keeping other parameters the same will increase Id. Since Id is inversely proportional to delay, this means that more current will result in less delay. This would explain why the delays in part 6 are less than that of part 5, and why the change is also easily visible between part 6 and part 5.

The yielded power results exhibit similar behaviors to part 5, where dynamic power between graphical and simulation are very closely equal. In addition, we can see that Part 6's static power has more error between graphical and simulation when compared to part 5 – where static power of part 5 is almost exactly equal between the 2 methods. This could be because Id has increased with the increase with width, thus leaking more current during its static state. This would explain very well the relationships between current vs parasitic capacitance vs leakage.

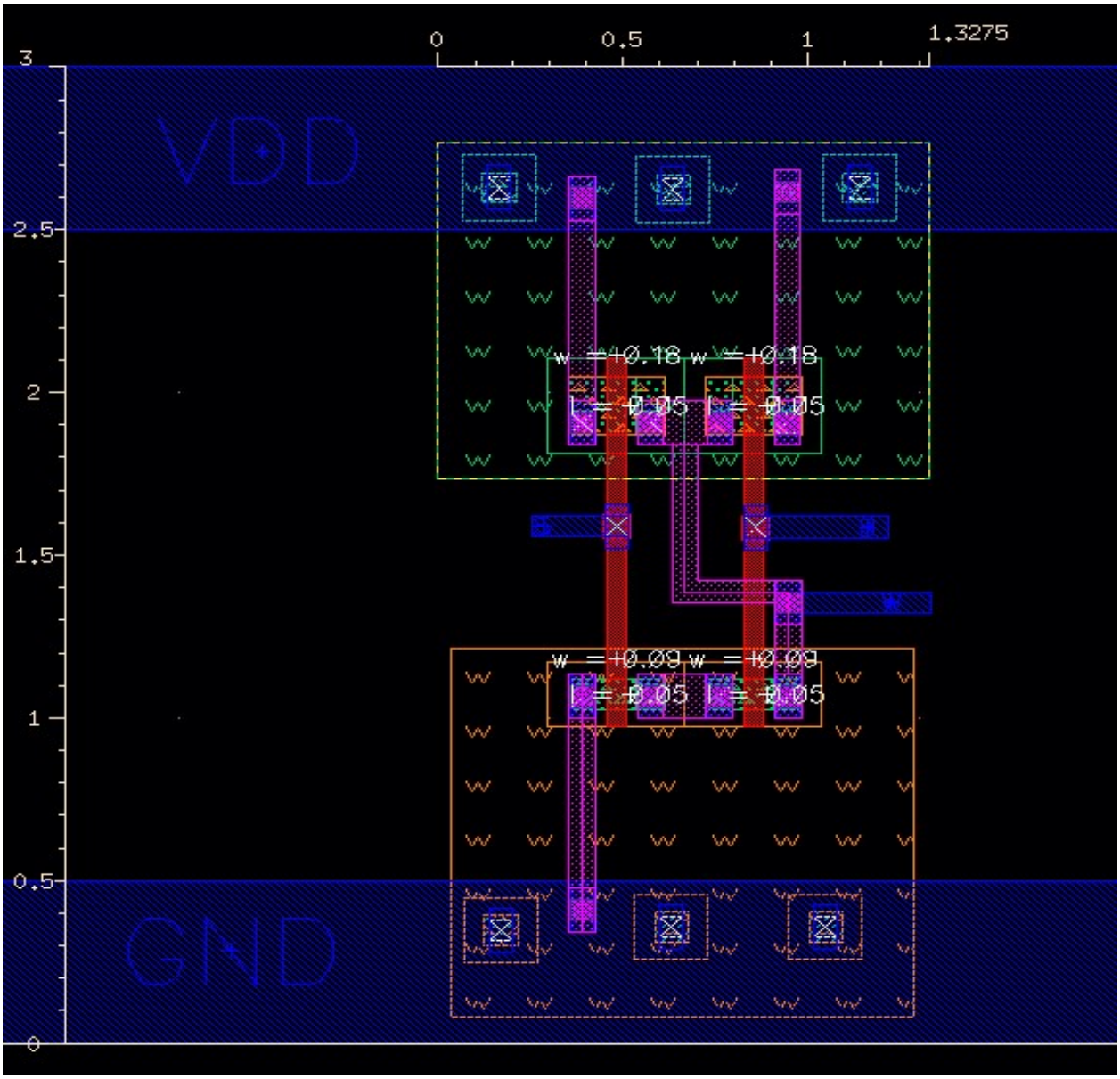
(Part 7 next page)

----------Part 7 – Layout using NAND gate:----------

Layout:

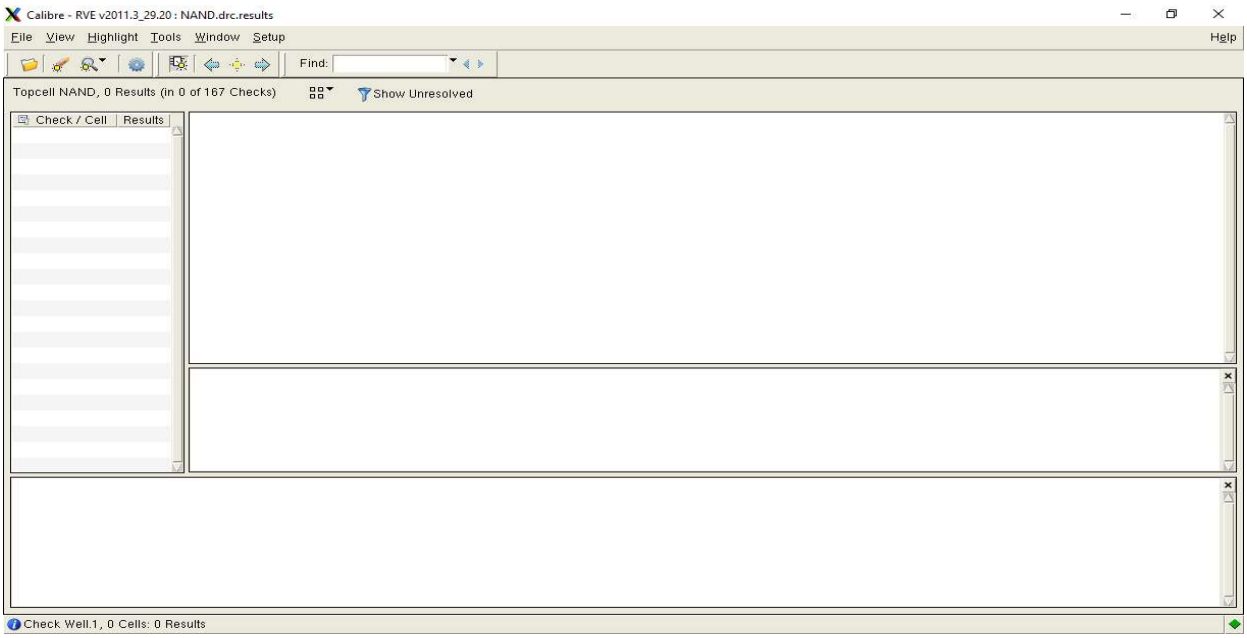
Lambda = 25nm

Height = 128*lambda = 3200nm = 3.2um

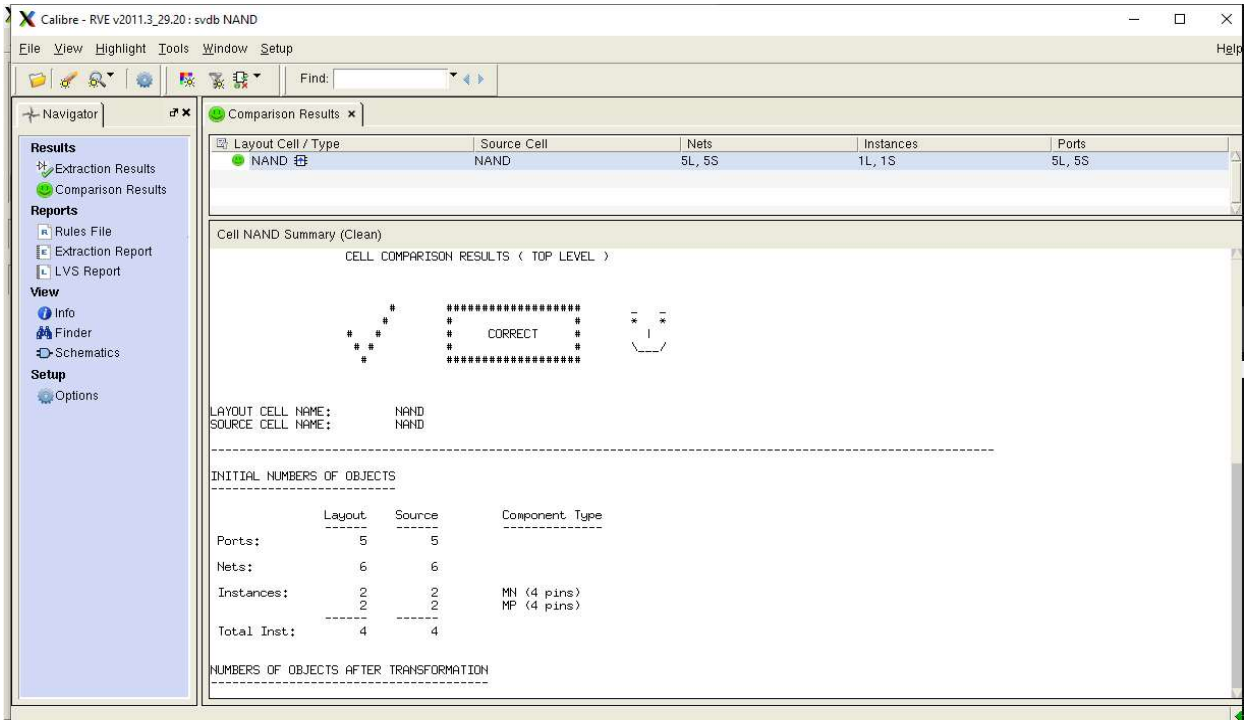


The total cell height is 3um which adheres to the design constraint, which was 3.2um cell height.

DRC check:



LVS:



----------Part 8 – Performing Parasitic Analysis----------

Parameters are exactly the same with part 5 while taken parasitic capacitance into account.

Results appended to Table in Part 5

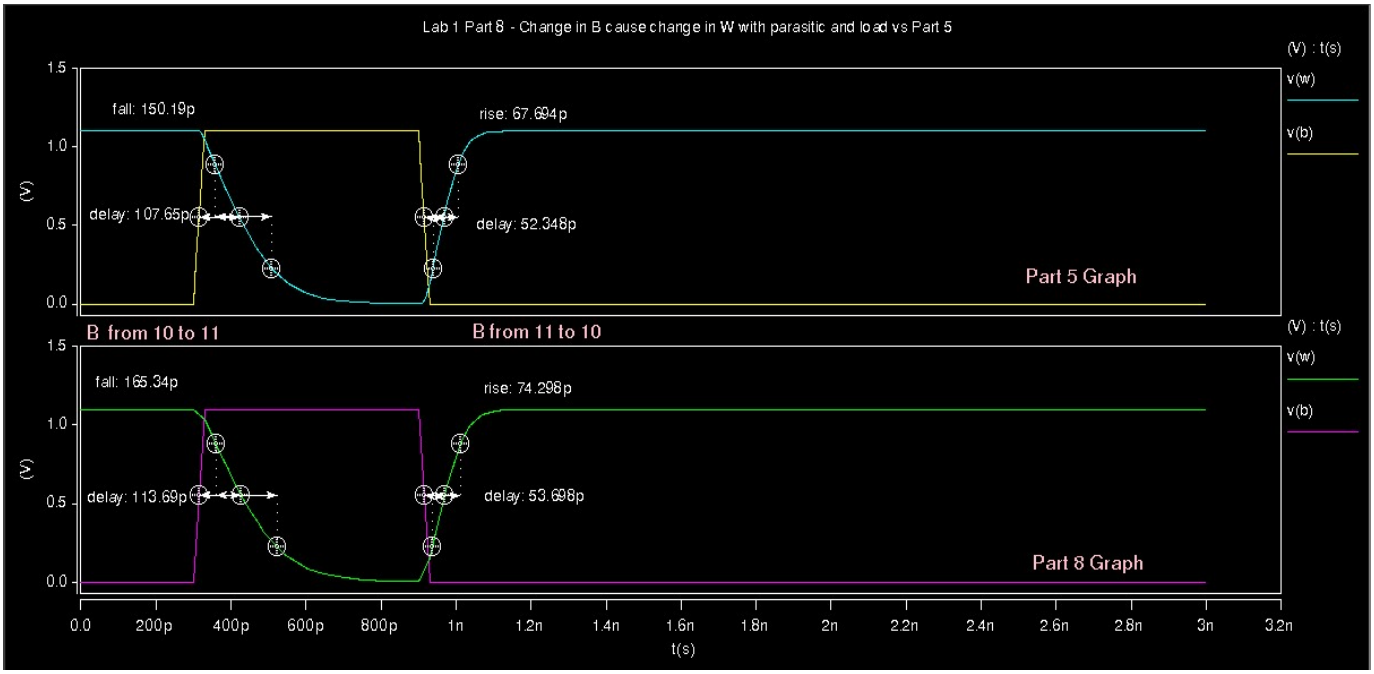
Using similar methods and reasons for part 4, 5 and 6, we have:

Part 8 Graphs of Single Input Change Cause Output Change with Load and Parasitic vs Part 5

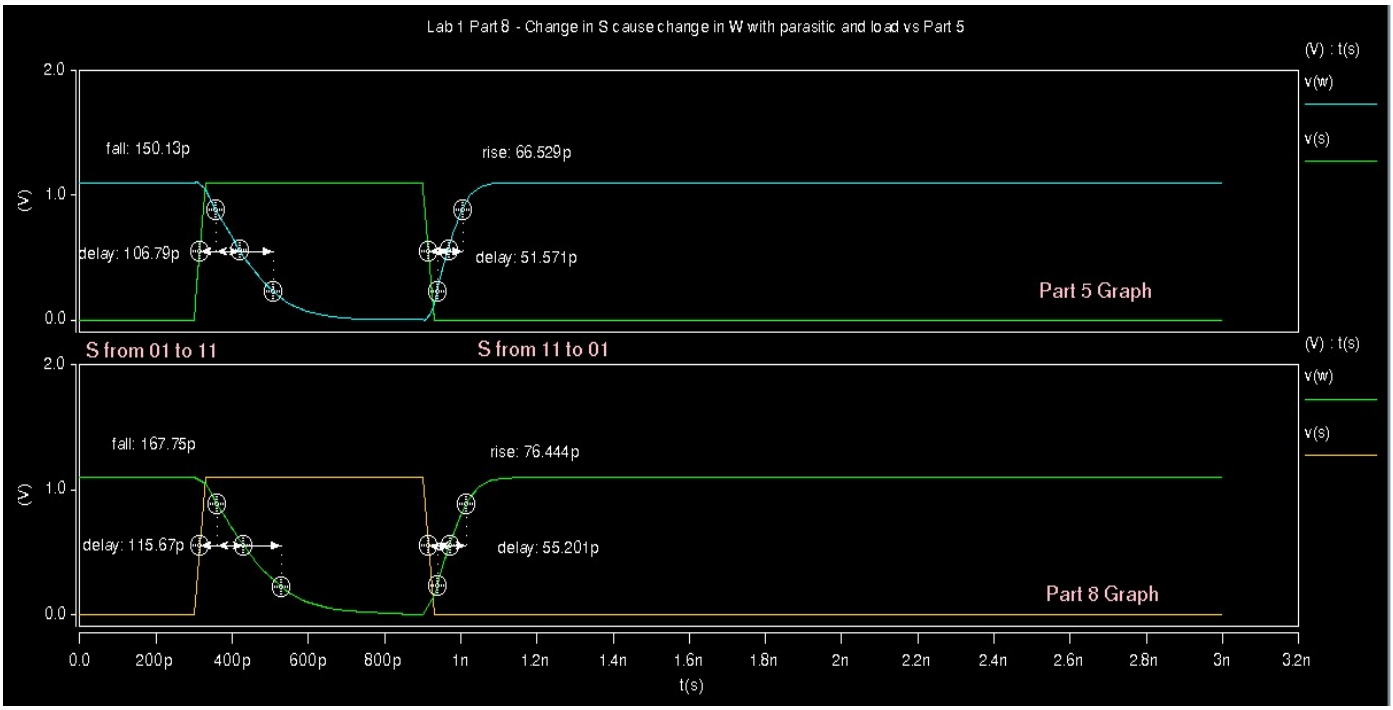
Input pattern: for B – 10,11,11,10 | for S – 01,11,11,01

This pattern is similar to that of part 4, 5 and 6

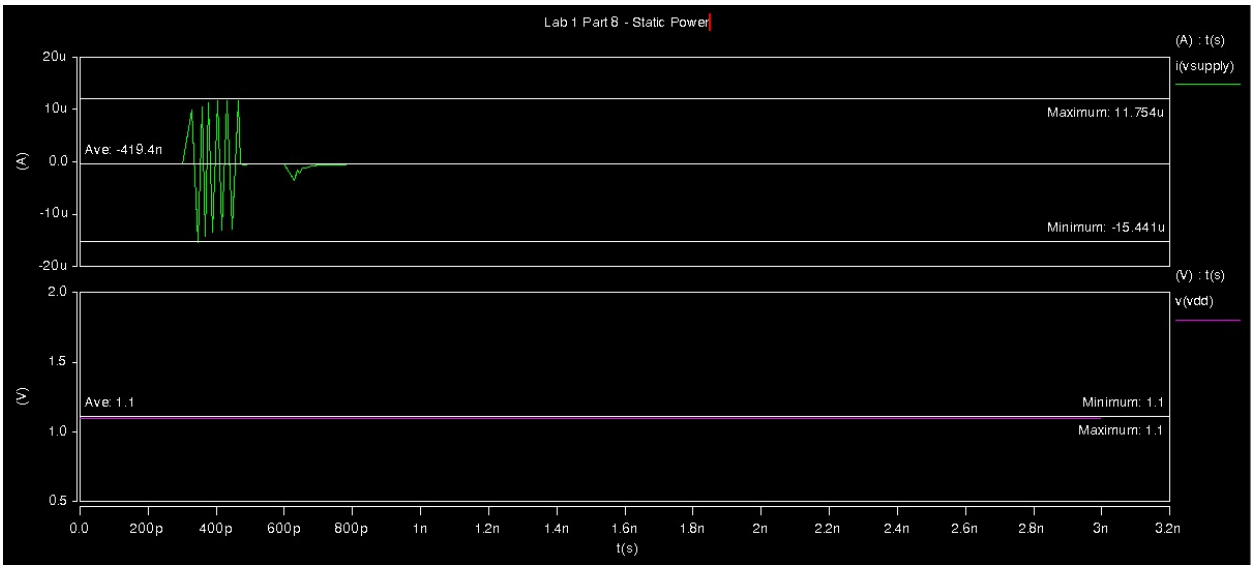
Graph of change in B causes W to change with load and parasitic (10 to 11 and 11 to 10):



Graph of change in S causes W to change with load and parasitic (01 to 11 and 11 to 01):



Power of circuit after load and parasitic where inputs states are 00, 01 and 10 (static values):

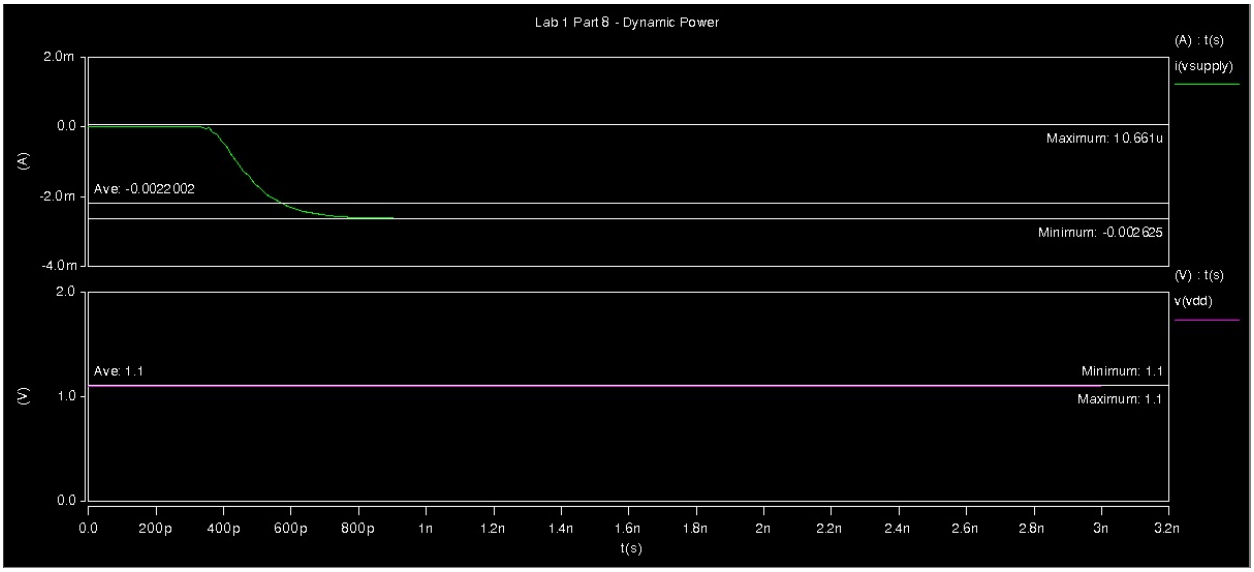



```
$DATA1 SOURCE='HSPICE' VERSION='E-2010.12-SP2 32-BIT'
.TITLE '* file: nand.pex.netlist'
 wn                avgpower          maxpower          temper
                   alter#
    0.              5.418e-07         8.735e-06         25.0000
                   1
```

P static graphical = I_static-avg * VDD = -419.4n * 1.1 = -4.613e-07 W

P static simulated = I-avg * VDD = avgpower = 5.418e-07 W

Power of circuit behavior after load and parasitic where inputs are 00 to 11 (switching):



```
$DATA1 SOURCE='HSPICE' VERSION='E-2010.12-SP2 32-BIT'
.TITLE '* file: nand.pex.netlist'
 wn                avgpower          maxpower          temper
                   alter#
    0.              2.423e-03         2.887e-03          25.0000
                   1
```

P dynamic = I_switching-peak * VDD = - 0.00265 * 1.1= -2.893e-03 W = -2.893 mW

P dynamic = I-peak * VDD = maxpower = 5.906e-03 W = 2.887 mW

Compare to part 5:

To start off, parasitic capacitance is formed when two electrical conductors at different voltage are close to each other and their electric field overlaps. This is also the reason why we have rules for spacing between components and we avoid shrinking devices further as parasitic capacitance will become a larger threat as these devices shrink in proximity.

Parasitic capacitance most notable effect on a circuit is noise or and change in oscillation at high frequency. At low frequency (f = 0hz) this effect is negligible because it is simply the DC solution, thus, unaffected by frequency. However, since impedance (and admittance) is heavily influenced by frequency domain, such is the reason why we see large delay or oscillations when taking account of parasitic capacitance.

The data shows that all 4 time data (Tr, Tf, Tdr, and Tdf) of part 8 are slightly longer than that of part 5, which means that by taking parasitic capacitance into consideration the waveform is distorted thus, making the rise and fall time much slower in part 8. In addition, it can be evidently pointed out that, the higher the frequency (as time/frequency goes on), we observe much more delay. This agrees with our statement above, as the higher the frequency, the more toll parasitic capacitance impose on the system.

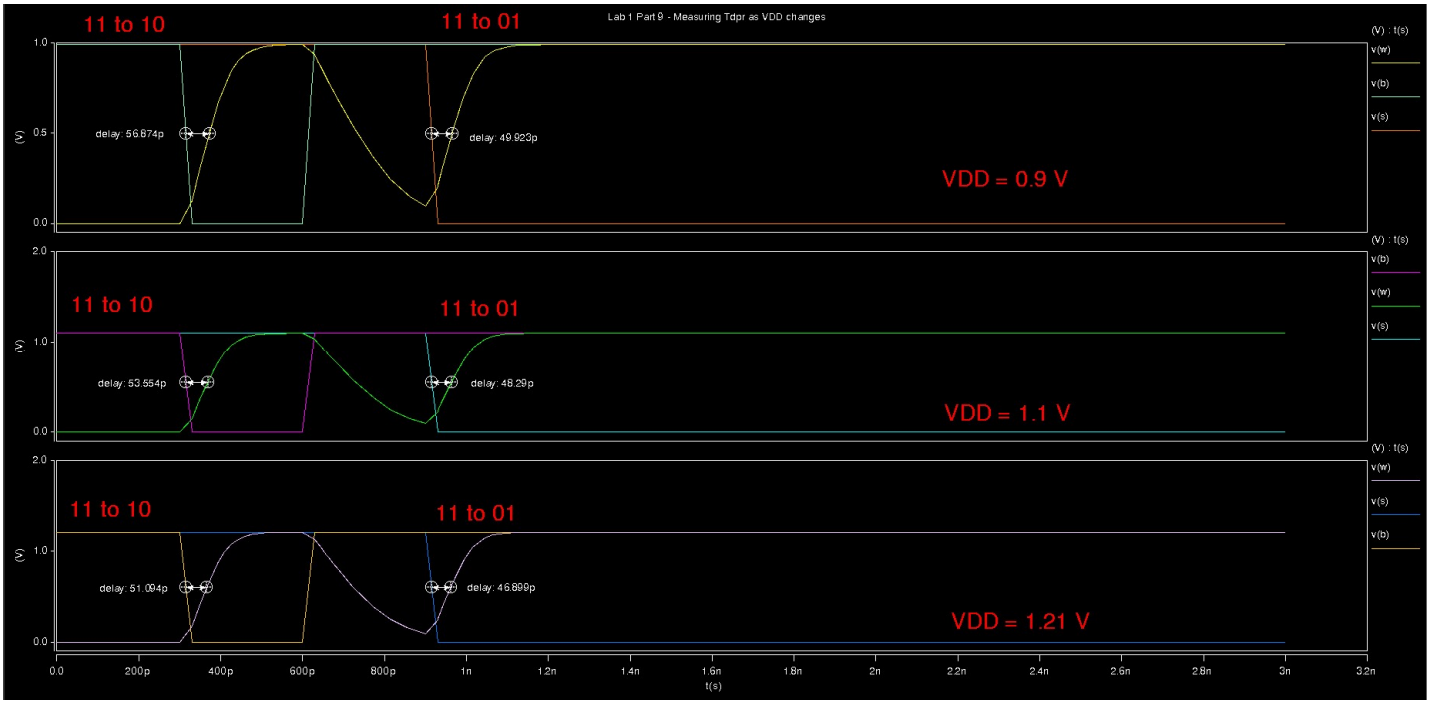
As for power consumption, hspice simulation between part 5 and 8 are roughly equal. While the hand calculation of dynamic power is similar, however, the static power of hand calculation is unusually off between the two. This could be because the parasitic capacitance took away some drive current going the FETS and use that to charge up these parasitic capacitances instead. In a sense, that power is never lost, however it was dissipated at a different place, causing the drive current to be different; ultimately causing the difference between hand calculation between static power of part 5 and 8. Static power is the same as it was only during the brief moment so these capacitances haven't much chance to influence the drive current.

----------Part 9 – Characterizing your design----------

Section a: measuring Tpdr with part 5 load

Since we are measuring the rise only, we only need to concern about when V(w) goes from low to high. Which correspond single bit input change of 11 to 10 and 11 to 01.

Graph of Tpdr vs change in VDD with same load from part 5:



From this graph, we have the table:

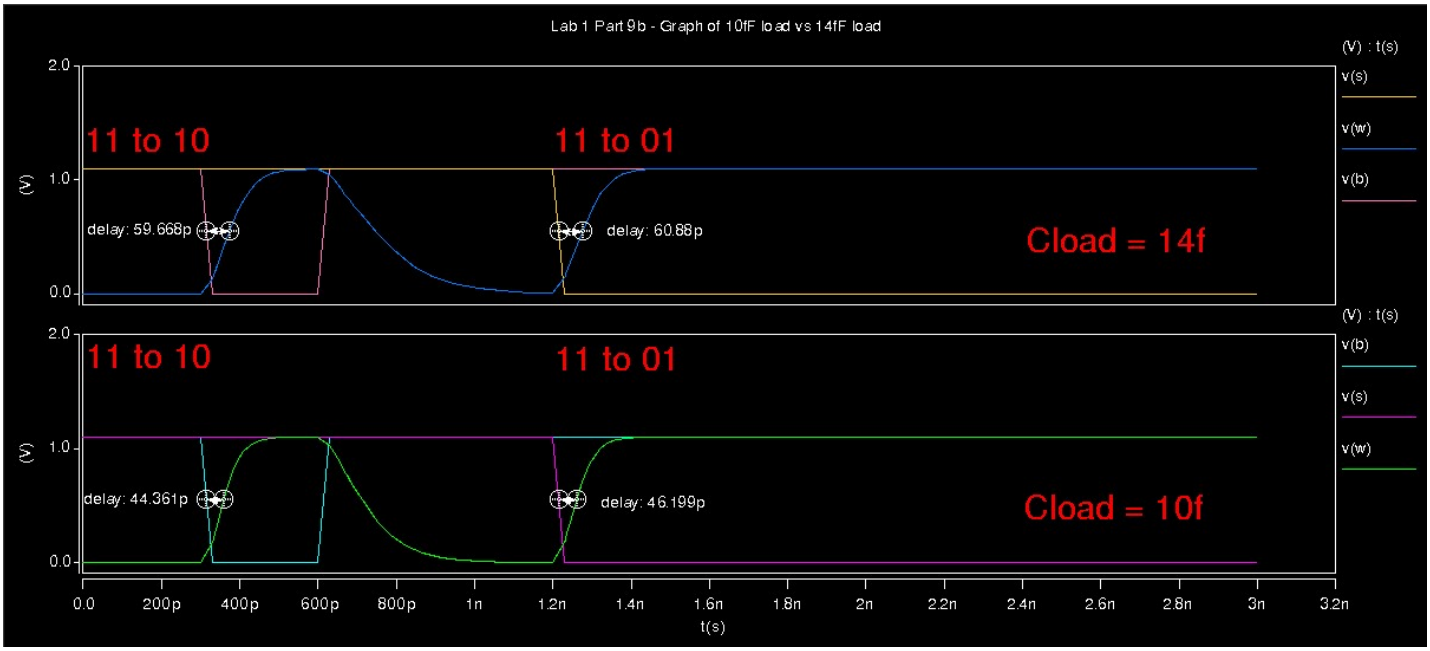
Part 9a – with Load	Input (SB)	VDD (V)	Output delay vs inputs	Time (ps)
B edge	11 to 10	0.99	Tpdr	56.874
B edge	11 to 10	1.1	Tpdr	53.554
B edge	11 to 10	1.21	Tpdr	51.094
S edge	11 to 01	0.99	Tpdr	49.932
S edge	11 to 01	1.1	Tpdr	48.29
S edge	11 to 01	1.21	Tpdr	46.899

We see that, as VDD increases, we have lower delay rise time. It is a linear relationship between change in voltage and change in Tpdr. The higher the voltage, the sharper the slope, thus resulting in slower delay rise time between input and output.

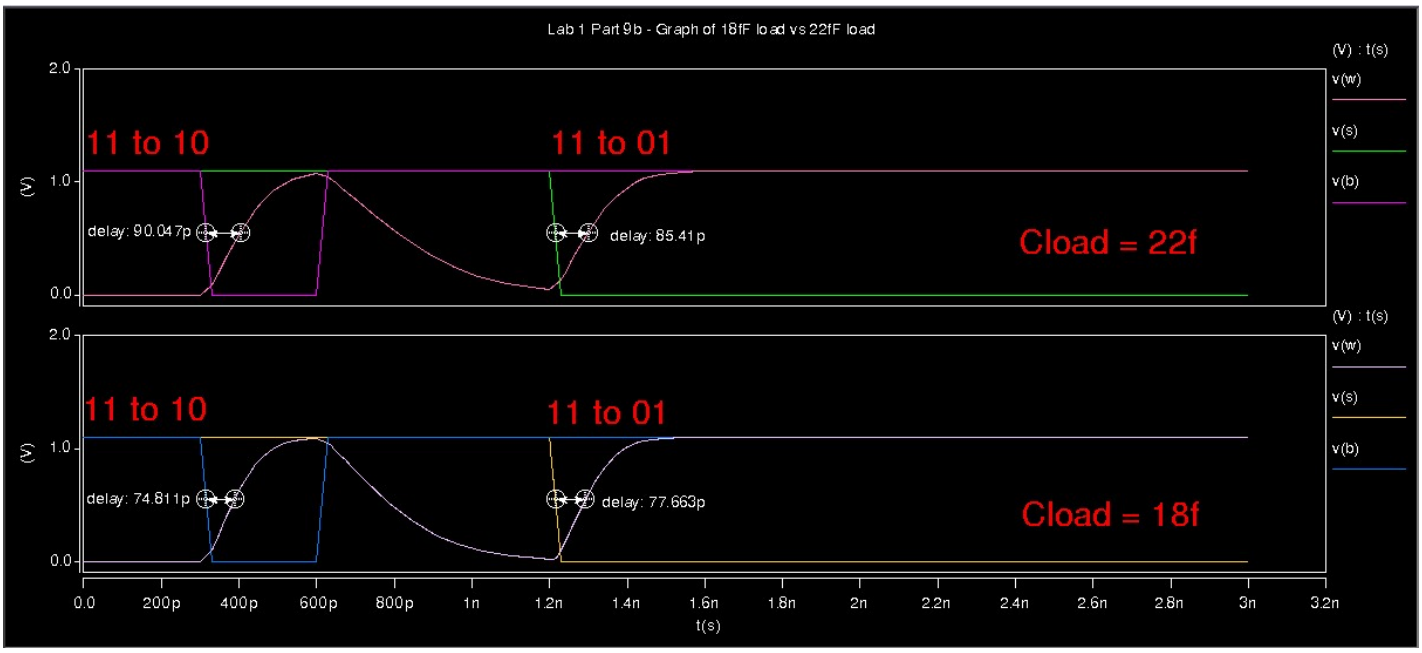
This is because logic gates switch/operate faster as VDD increase => reduce delay time.

Section b: measuring Tpdr varying Load Capacitance, keeping VDD = 1.1V

Graph of Tpdr vs change in load capacitance (no inverters):



Graph of Tpdr vs change in load capacitance (no inverters):

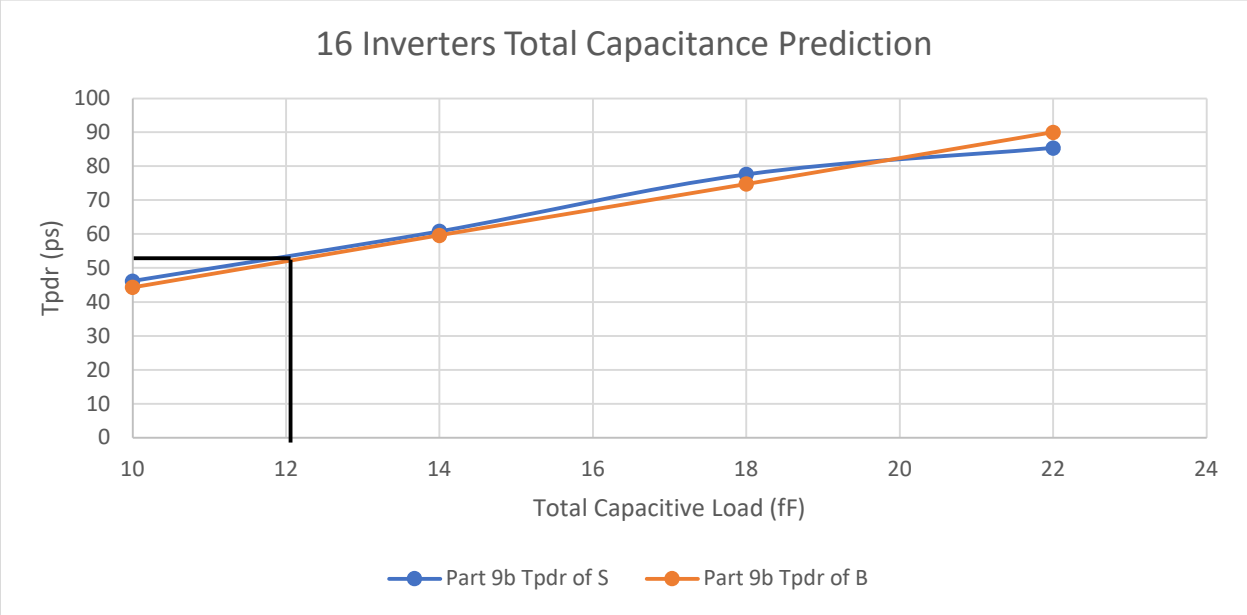


By keeping VDD = 1.1V, and only changing the load capacitance, we could estimate how much is the equivalent capacitance of all 16 inverters without having to perform all the mathematical steps behind it. Thus, by comparing the results obtained from 9b to part 5, we can estimate inverters' capacitive load by:

Ctotal = 7f + C_16inverters (Note that capacitors in parallel is additive)

We have the following table for 9b:

Part 9b – with varying Load	Input (SB)	Cload (fF)	Output delay vs inputs	Time (ps)
B edge	11 to 10	10	Tpdr	44.361
B edge	11 to 10	14	Tpdr	59.668
B edge	11 to 10	18	Tpdr	74.811
B edge	11 to 10	22	Tpdr	90.047
S edge	11 to 01	10	Tpdr	46.199
S edge	11 to 01	14	Tpdr	60.88
S edge	11 to 01	18	Tpdr	77.663
S edge	11 to 01	22	Tpdr	85.41



Recall from part 5 that Tpdr of loaded NAND gate with16 inverters and 7fF are:

Change in S Tpdr = 51.571 ps | Change in B Tpdr = 52.348 ps

Therefore, the black line drawn in the graph shows where the level of 16 inverters and 7fF Tpdr would be according to part 9b guessed loaded capacitance, which it comes out to be around 12fF.

Since capacitors in parallel have additive property:

Ctotal = 7fF + C_16inverters = 12fF => C_16inverters = 12fF – 7fF = 5fF

In conclusion, the capacitive load of 16 inverters is roughly equal to 5fF.