Standard Cell Template

AND-OR-INVERTER

Haranadh Chintapalli, Satya Ravi Teja Erpina, Soma Sai Charitha
Group #5
8th Febraury 2017

Introduction and Physical Properties

Cell Description

An AOI circuit implements two or more AND operations and a OR operation. Here a 2-1AOI is used whose logic is (A+B).C

It is implemented with CMOS technology. This AOI is preferred because it reduces the number of gates used when compared to separately implementing the AND, OR and NOT operations. This reduces the fabrication cost and area occupied. It also increases the speed of the circuit.

In this project we are trying to drive different loads (AOI) with another AOI Inverter whose dimensions are to be determined first.
The Design Under Test(DUT) which is the AOI is tested to meet the specifications which are: rise time and Fall time <0.06ns and the propagation delay of high to low and low to high <0.15ns

Cell Symbol

Where applicable use standard logic symbols (i.e. INVERTER, NAND, NOR, AOI has a standard symbol) for the assigned standard cells. This will require editing the symbol after Cadence NSCU CDK creates the boring square or rectangular default symbol. [Note: There are two "extra" pins inside the inverter symbol. These pins may be needed for a workaround when simulating the extracted view of the physical design later in the term and next quarter. The top symbolic pin is for the VDD and the lower for GND. If they are not needed they can be dropped in the schematic as well as in this picture.]

Standard Cell Template

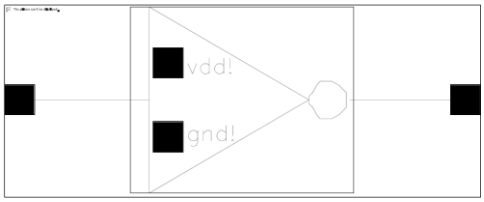

Figure 1: Example Logic Symbol from NCSU Digital Parts. All symbols are 1in tall.

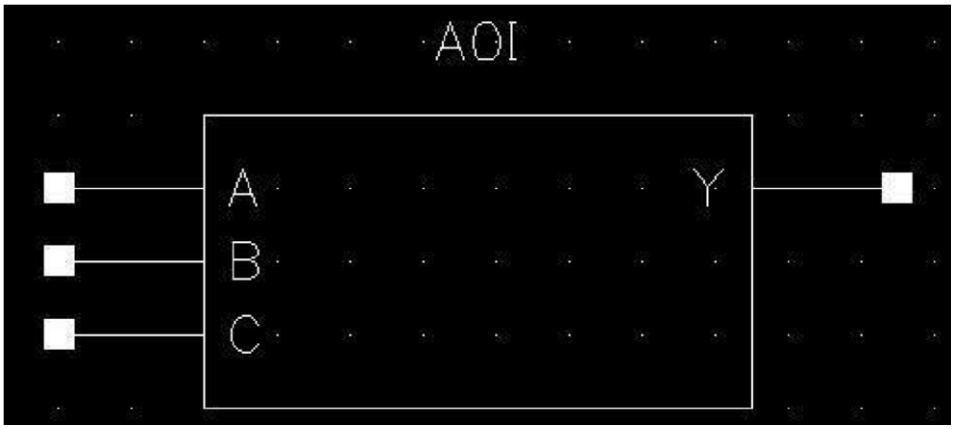

Figure 2: Symbol of AOI

Cell Truth Table

Complete the truth table for all cell outputs using {0, 1} for the input low and high, respectively and {L, H} for the output low and high, respectively. Repeat rows and columns as needed.

A	B	C	Y
0	0	0	H
0	0	1	L
0	1	0	H
0	1	1	L

Standard Cell Template

1	0	0	H
1	0	1	L
1	1	0	L
1	1	1	L

Table 1: Truth Table

Cell Logic Equation

$Y = \overline{(AB)+C}$

Cell Schematic Diagram

Prepare Encapsulated Postscript of the schematic for publication (Cadence has this option in the Virtuoso schematic design). Do not use a screen shot or create Encapsulated Postscript of the raw schematic. For each "publication schematic" in NSCU CDK remove the transistor width and length, model name etc. but leave the instance names of the pins and transistor. This makes the schematic easier to read.

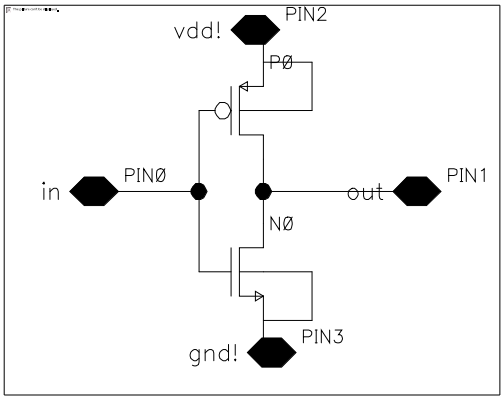

Figure 3: Example Schematic from NCSU CDK Publication Schematic. All schematic figure are 2in tall.

Standard Cell Template

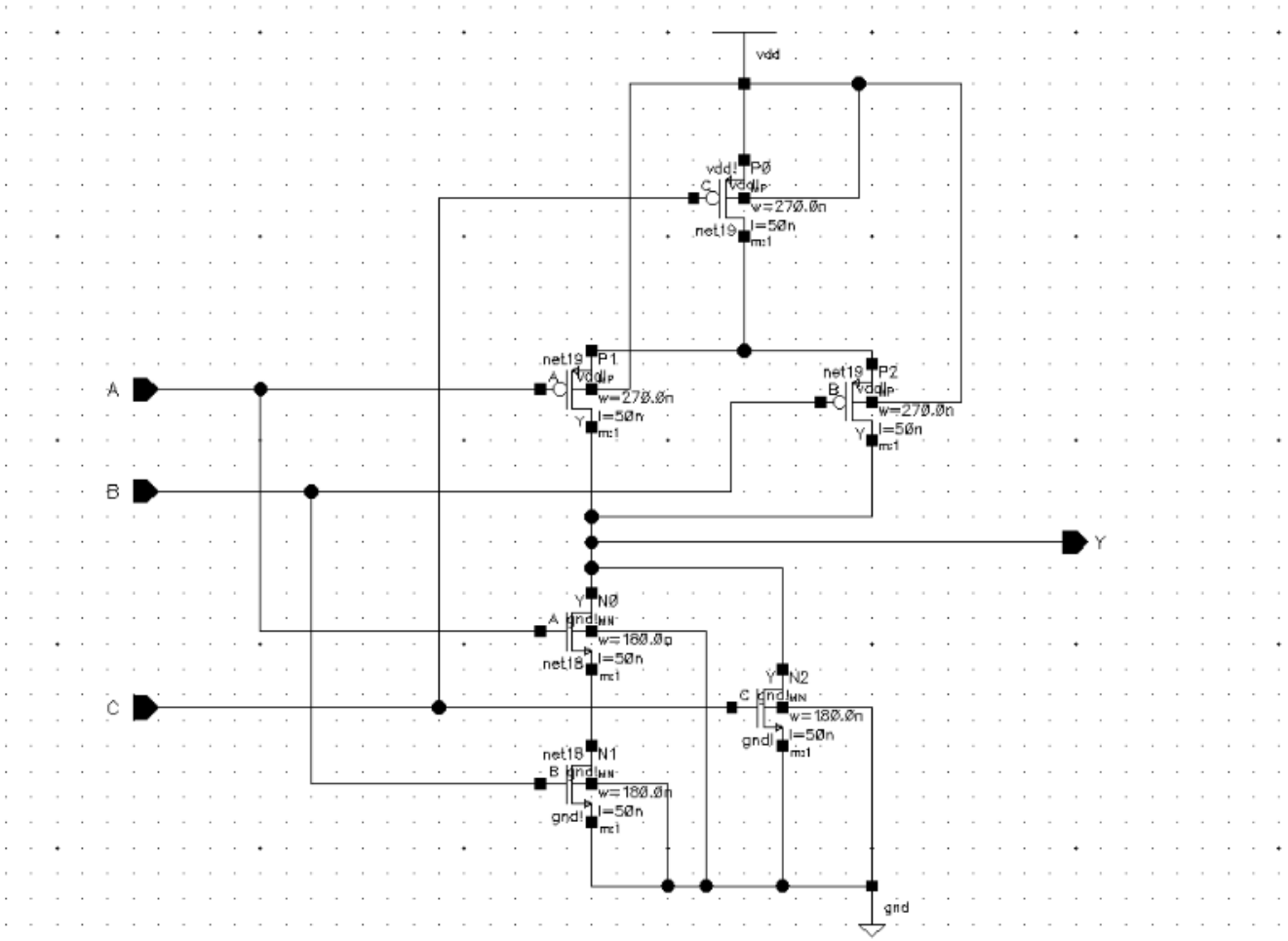

Figure 4: Transistor level of AOI

Standard Cell Template

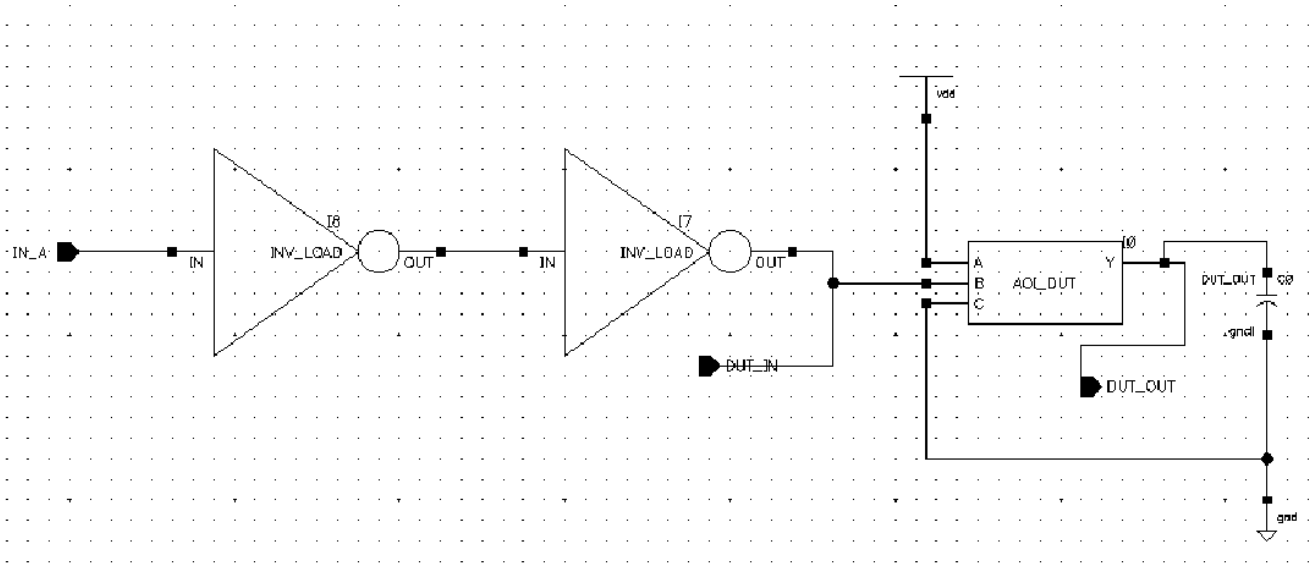

Figure 5: Schematic of FO0

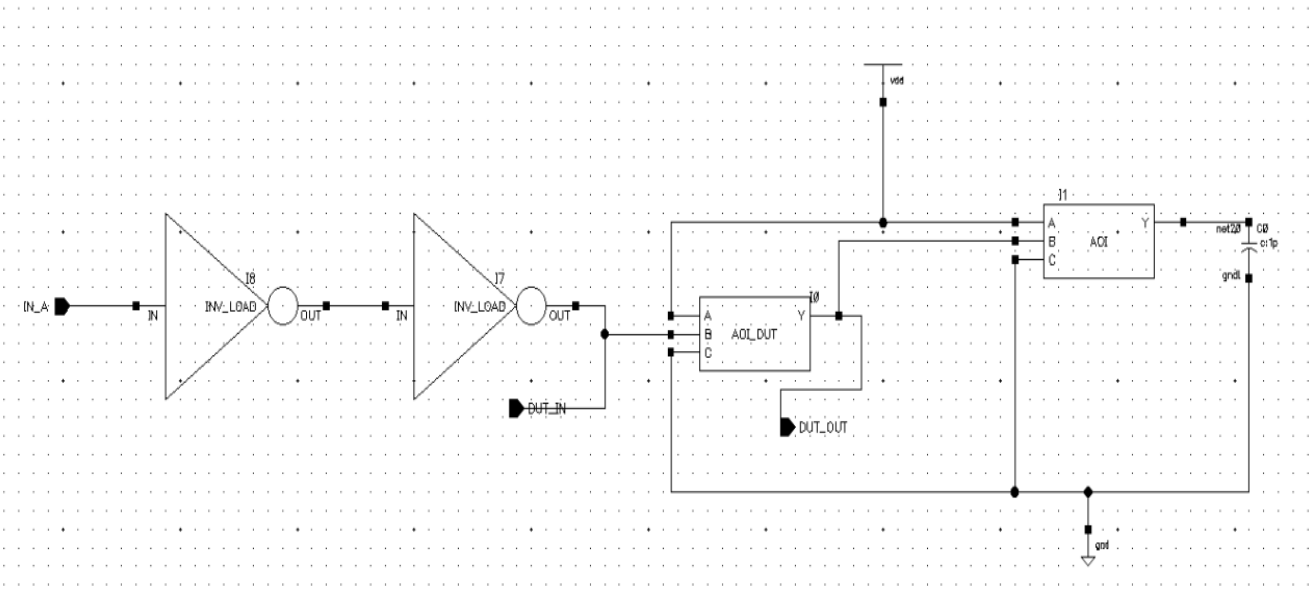

Figure 6: Schematic of FO1

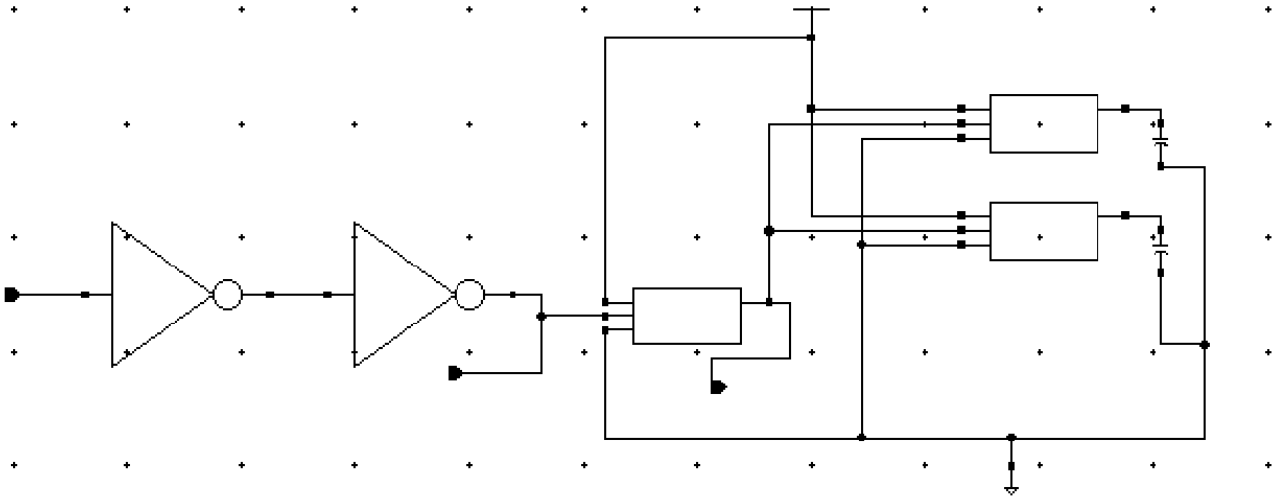

Figure 7: Schematic of FO2

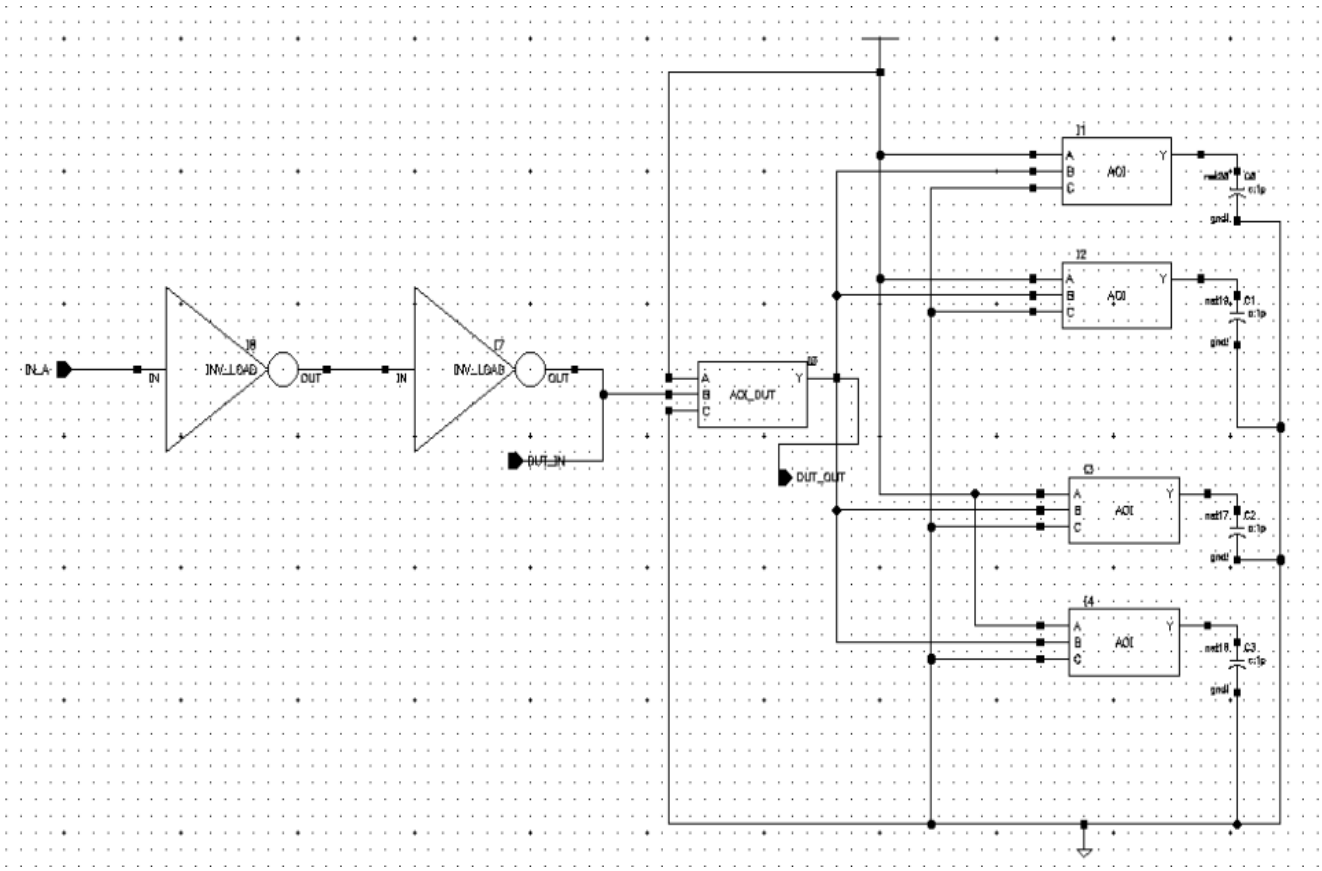

Figure 8: Schematic of FO4

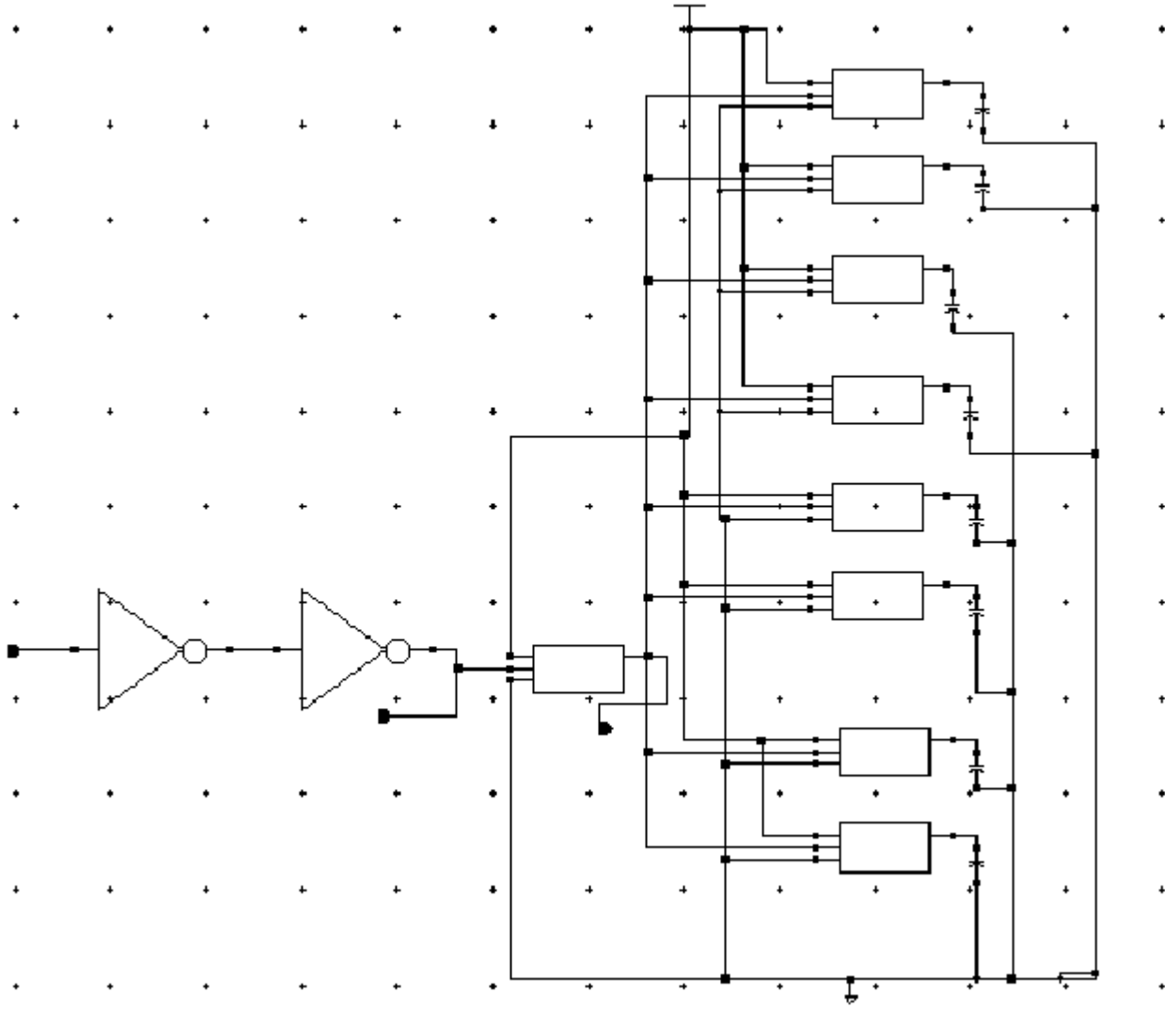

Figure 9: Schematic of FO8

Cell Layout Diagram and Dimensions

Save a colour or black and white layout of the cell in EPS (i.e. Encapsulated Postscript) format. The cell dimensions are saved in both lambda ( ) and microns (m). Record the transistor length and width dimensions (nm).
[Repeat the transistor row as needed.]

Standard Cell Template

Cell Physical Dimensions		
	X	Y
Cell Dimension in □□		
Cell Dimension in □m		
Transisto r Dimensions		
Transistor Instance Number	Length (nm)	Width (nm)
PMOS	50	945
NMOS	50	425

Table 2: Dimensions of AOI

Cell Physical Dimensions		
	X	Y
Cell Dimension in □□		
Cell Dimension in □m		
Transisto r Dimensions		
Transistor Instance Number	Length (nm)	Width (nm)
PMOS	50	270
NMOS	50	180

Table 3: Dimensions of AOI (Fanouts)

Input and Output Parasitic Capacitance Table

From the schematic calculate each input's capacitance normalized to the nominal inverter (your inverter standard cell) by the width of the transistor or drain area as needed. This entry should be an integer fraction similar to Weste and Harris. [Note the normalization is to a standard inverter (the standard cell inverter INV1X). Repeat the rows as needed.]

Computed Cell Input Capacitance	
Input Name	**Capacitance (/Cinv)**

Standard Cell Template

Output Name	**Capacitance (/Cinv)**

Performance Analysis

Rise and Fall Times

[Note: It is highly desirable to split the simulation work load among the team members so that each team member learns how to use the tools.]

FOx denotes output loads. The loads are defined by the number of identical logic gates. Use 20%-80% swings for the output rise and fall entries. Use a 1.2V power supply.

For each output load in the table complete transient simulations. Remember to include a CMOS non-inverting buffer between the ideal voltage source and the logic gate driving the FOx load. Note rise t_r / fall t_f times are at the input to the logic gate driving the load, **not** the rise/fall times for the input ideal voltage source.

Complete the number needed copies (copies = No. input stacks x No. outputs) of the table below.

For multi-input gates, complete tables for each transistor stack (i.e. each branch connected to the output) using the stack's worst case single controlling input transition in the stack. Label the tables with worst case input in each stack and the output. Replace **X** below with the signal name.

Input X: Output Rise Time Data t_r (ns)					
Input rise/fall time (ns)	Output Load (FOx) 0	1	2	4	8
0.04	0.0864	0.055	0.0604	0.0715	0. 0934
0.06					

Standard Cell Template

Stack Input Combination: ***Replace with Boolean Product***

Stack S, Input X: Output Fall Time Data t_f (ns)					
Input rise/fall time (ns)	Output Load (FOx) 0	1	2	4	8
0.04	0.0792	0.0594	0.0628	0.0698	0.0847
0.06					

Stack Input Combination: ***Replace with Boolean Product***

Propagation Delays

For the range of output loads shown in the table simulate propagation delays (low to high t_{plh} and high to low t_{phl}) for the stack's worst case single controlling input transition. The input controlling the output is the same input reported in the rise and fall time section. Use a 1.2V power supply and timing measurements start when input to the logic gate driving the FOx load crosses the 50% of the rail and stop when the logic gate driving output crosses 50% of the rail. Negative values are entered as 0.

Label the tables with the Boolean product (e.g. AB) of the transistor stack and the output. Complete copies of the table below for each branch connected to the output.

Data Worst Case Low to High Propagation Delay Data t_{plh} (ns)					
Input rise/fall time (ns)	Output Load (FOx) 0	1	2	4	8
0.04	0.0847	0.0492	0.0533	0.0611	0.0764
0.06					

Worse Case Input Combination: ***Replace with Boolean Product***

Data Worst Case High to Low Propagation Delay Data t_{phl} (ns)

Standard Cell Template

Input rise/fall time (ns)	Output Load (FOx) 0	1	2	4	8
0.04	0.0441	0.0454	0.0823	0.0589	0.0748
0.06					

Worse Case Input Combination: ***Replace with Boolean Product*** From each row of the slew rate data compute the best fit linear propagation delay equation for low-to-high T_{plh} (h) and high-to-low T_{phl} (h). The model predicts a delay, in nanoseconds, as a function of the output load, *h, Cout/Cin =FOx*. The model line is parameterized by a slope, *m*, and an intercept, *b*. The units of *m* are (*ns/FOx*) and the units of *b* are *ns*.

Complete the table below by increasing the number of rows for multiple input gates. The row labelled **All data** is the computed slope and intercept after combining data from all slew rates.

Complete the **Model** row for the gate using the assumptions and methods of the linear delay model from Weste and Harris. Only skewed standard cells will have different values propagation models for rising and falling inputs.

All data means combine the results for both slew rates into a single model.

Discuss in your own words the differences in the calibration and the Weste Harris linear delay model. Discuss the differences in high-to-low versus low-to-high models.

Data Model Propagation Delay Equation $T_{pd}(h) = b + m \cdot h$				
Input Slew Rate (ns)	Parasitic Rising Delay (b_r)	Rising Logical Effort (m_r)	Parasitic Falling Delay (b_r)	Falling Logical Effort (m_f)
0.04				
0.06				
All data				

Standard Cell Template

In the table below normalize the model for the T_{pd} (h) results of the table above to give the logical effort model *D(h)* described in Weste and Harris. *D(h)* is a unitless value and predicts the delay as multiples of the standard inverter delay.
Normalization is based on the observed CMOS inverter parasitic delay, b_{inv}. *Recall* ***all data*** p_{inv} *1.* □

Inverter Normalized Data Model Propagation Delay Equation *D(h)* $= p + g \cdot h$

Inverter Normalized Data Model Propagation Delay Equation				
Input Slew Rate (ns)	Parasitic Rising Delay (p_r)	Rising Logical Effort (g_r)	Parasitic Falling Delay (p_f)	Falling Logical Effort (g_f)
0.04				
0.06				
All data				
W&H Model				

Power-Delay

Simulate the cell for a sequence of input combinations based on the Gray code and compute the time averaged power (mW), average delay (ns), and average powerdelay product (mW ns = pJ). The Gray code restricts the simulations to single input transitions and ignores the large number of multiple input change combinations. Use the same slew rate for all input transitions. Use equal output loads for multiple output gates. Use a period of 2X maximum output delay with FO=8.

Average Power Data (mW)

Standard Cell Template

Input Slew (ns)	Output Load (FOx) 0	1	2	4	8
0.04					
0.06					

Average Delay Data (ns)					
Input Slew (ns)	Output Load (FOx) 0	1	2	4	8
0.04					
0.06					

Average Power-Delay Data (pJ)					
Input Slew (ns)	Output Load (FOx) 0	1	2	4	8
0.04					
Average Power-Delay Data (pJ)					
0.06					

Standard Cell Template

LAB 2					
AOI -FO4 (for Input tr=0.05ns, tf=0.05ns)					
DUT		Rise/Fall and delay time			
Wp(nm)	Wn(nm)	Tr(ns)	Tf(ns)	Tplh(ns)	Tphl(ns)
270	180	0.1382	0.0759	0.1012	0.0691
315	225	0.1153	0.0781	0.0909	0.0595
360	270	0.150	0.0704	0.0850	0.0535
405	315	0.111	0.068	0.08	0.051
450	360	0.0908	0.0625	0.077	0.045
495	405	0.086	0.061	0.0765	0.0448
540	450	0.0833	0.0602	0.0748	0.0425
585	495	0.079	0.0603	0.735	0.0410
630	540	0.078	0.0595	0.0725	0.04
675	585	0.0735	0.059	0.0718	0.038
720	630	0.074	0.061	0.071	0.0367
765	675	0.0727	0.0639	0.0709	0.0358
810	720	0.0723	0.067	0.0688	0.0365
855	765	0.0695	0.072	0.0671	0.037
900	810	0.0694	0.075	0.0578	0.049
1035	425	0.069	0.0733	0.0591	0.058
945	425	0.0713	0.0693	0.0612	0.059
1035	360	0.0663	0.077	0.0563	0.0697
900	425	0.0727	0.0678	0.0623	0.0575
1125	450	0.068	0.0757	0.0585	0.0626
1035	900	0.0866	0.0689	0.0807	0.0253
1035	945	0.0883	0.0691	0.0828	0.0224

Justification for the Widths chosen:

As the width of the PMOS transistor (W_p) increases, rise time (T_r) decreases and fall time (T_f) increases.

As the width of the NMOS transistor (W_n) increases, rise time (T_r) increases and fall time (T_f) decreases.

It is a matter of size vs speed considerations. It is a good way to increase/decrease the width of the transistor(pmos/nmos) to increase/decrease the speed, but this should not be done beyond certain limitation because it makes it difficult (not commercially

feasible) to fabricate such large dimension MOSFETS. Several iterations are made, so that rise time, fall time, propagation time (lowhigh), propagation time (high-low) meet the specifications.

The widths W_p=945nm and W_n=425nm, gives the best possible values which nearly meet the specifications.
In almost all trials, either rise time or fall time meet the specifications but both the values are not going less than 0.06ns. They might go less than 0.06ns, but we have already tried the P-mos widths that go beyond 1000 .Fabricating such large values of P-mos is not practically feasible. Increasing the width N-mos is also not leading us to the to the desired specifications, it decreases the fall time and delay of high to low. So, this seems like an infinite loop of increasing P-mos and N-mos widths.

W_p=945nm and W_n=425nm are chosen because both rise time and fall time are almost similar. Such a circuit is practically more useful. Also, for these dimensions the delay values are meeting the specifications.
Some other dimensions give values in which only some of them (either rise or fall time ; either tplh or tphl) meet the specifications. Hence we choose W_p=945nm and W_n=425nm.

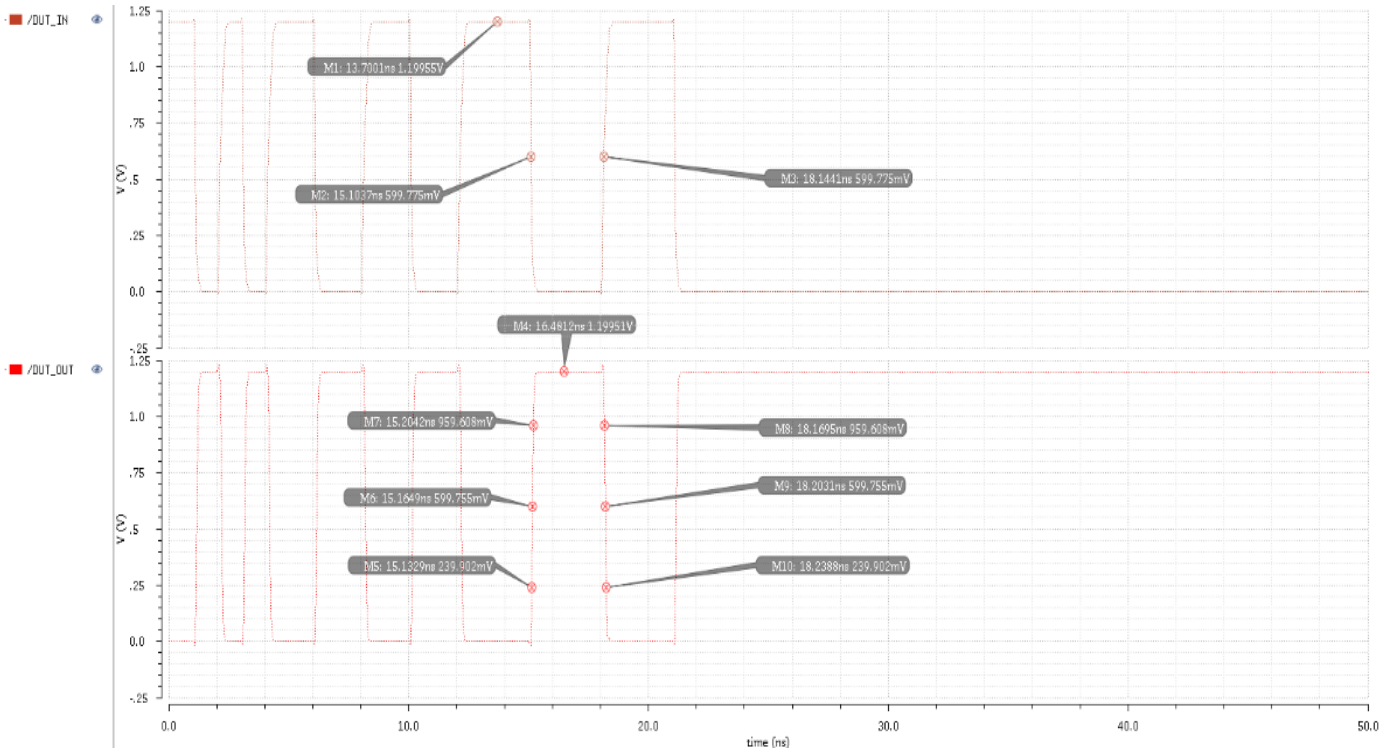

Figure 10: **Transient Analysis for Inverter with Pmos-945and N-mos-425 dimensions** (for Input tr=0.05ns, tf=0.05ns)

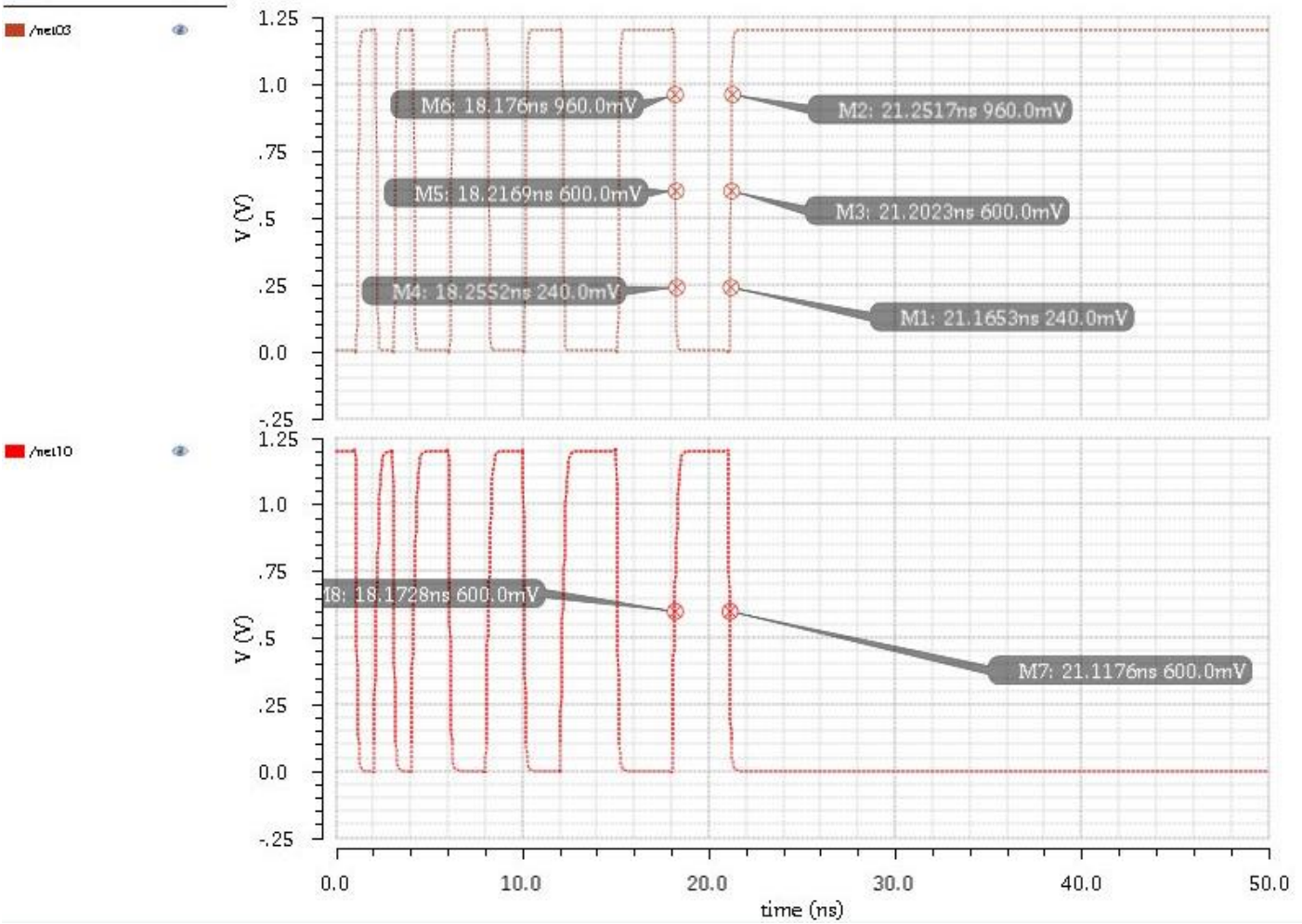

Figure 11: Transient Analysis of FO0 (for Input tr=0.04ns, tf=0.04ns)

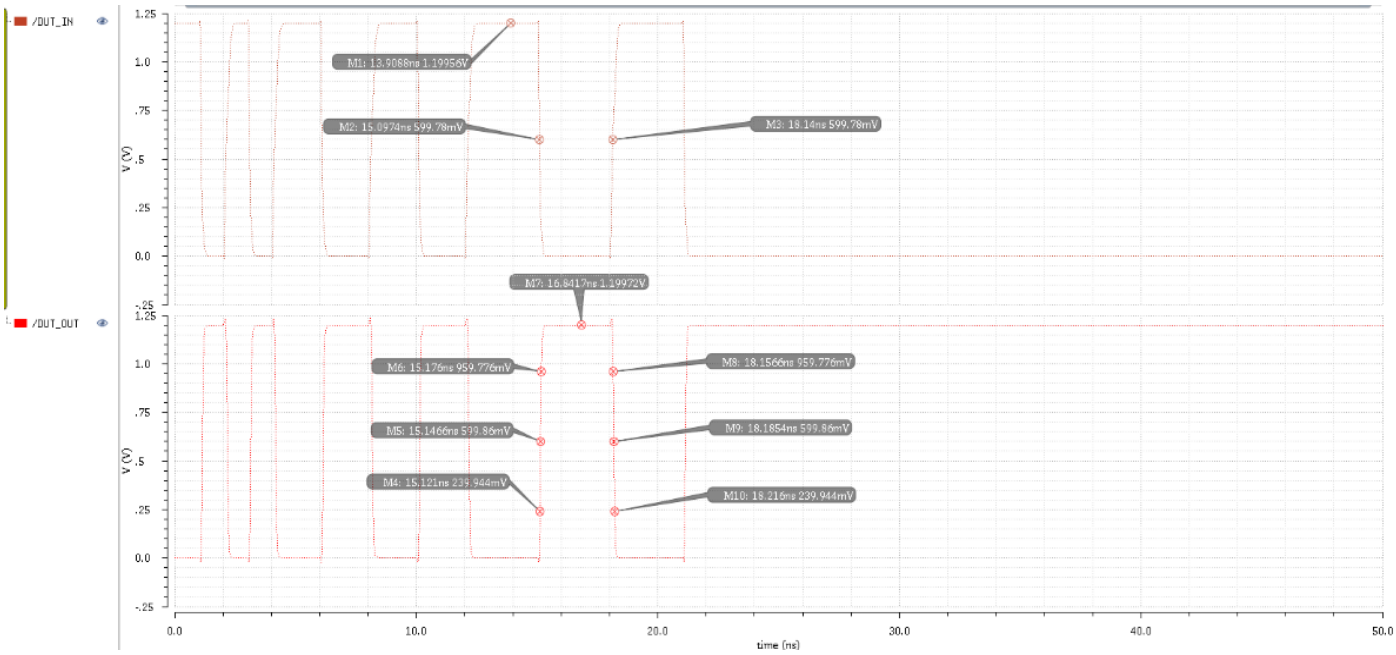

Figure 12: Transient Analysis of FO1 (for Input tr=0.04ns, tf=0.04ns)

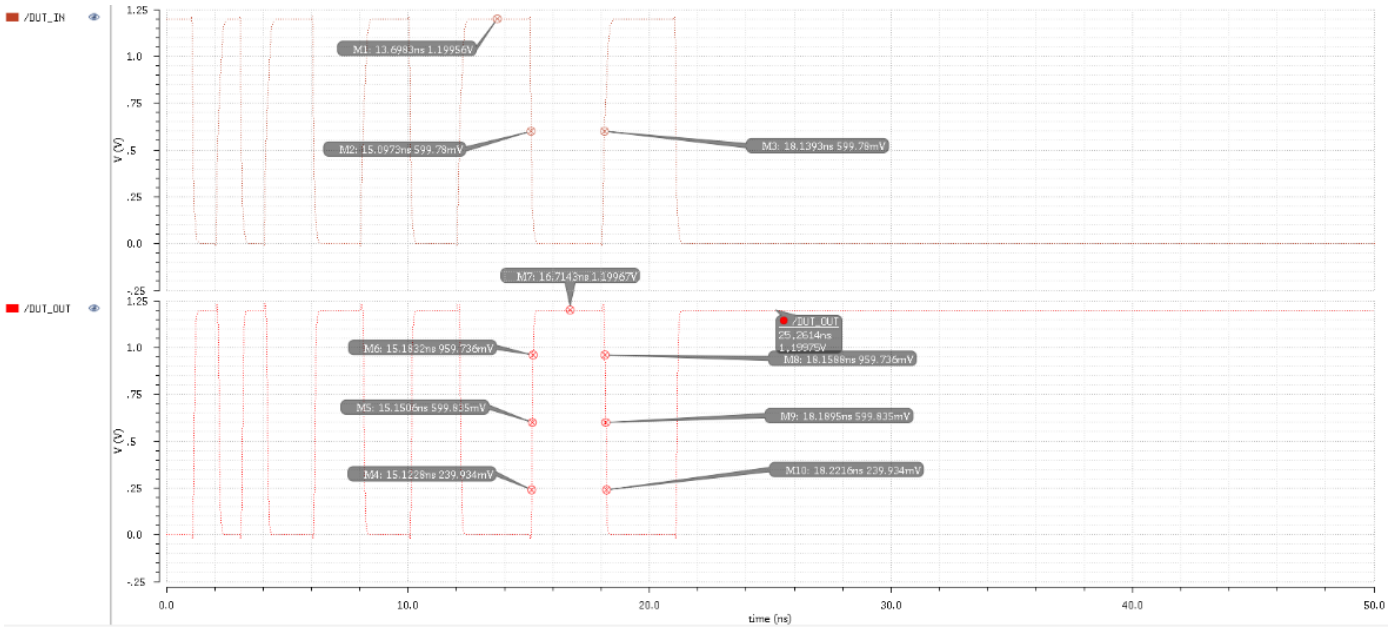

Figure 13: Transient Analysis of FO2 (for Input tr=0.04ns, tf=0.04ns)

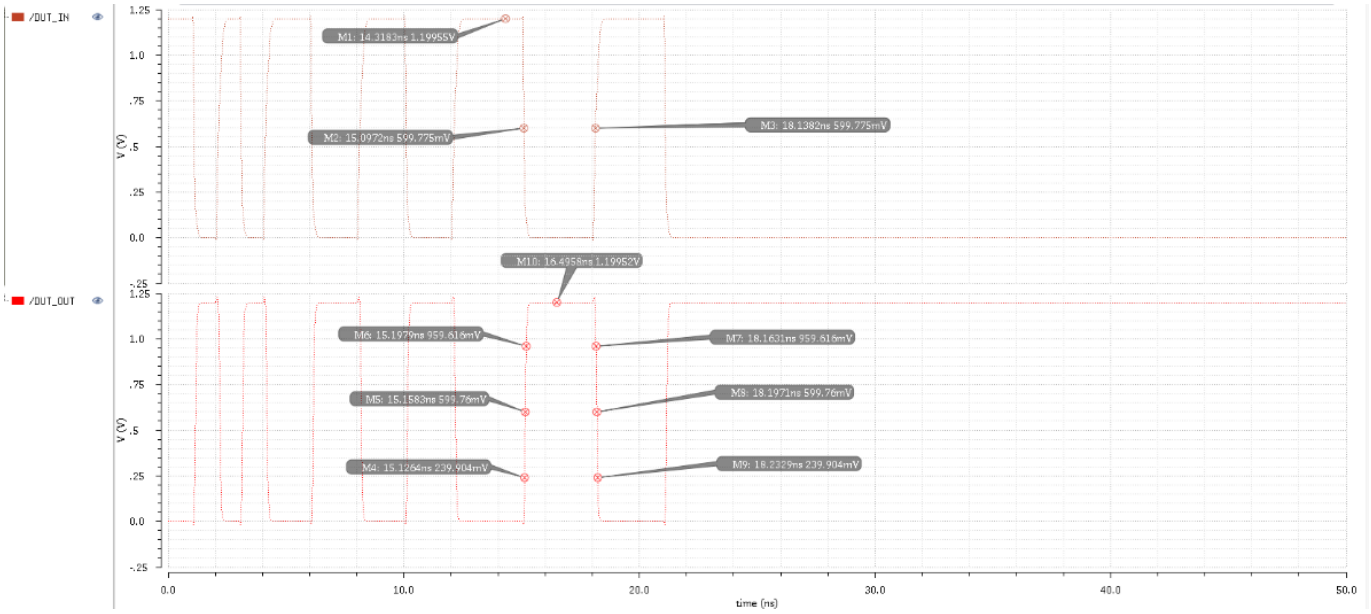

Figure 14: Transient Analysis of FO4 (for Input tr=0.04ns, tf=0.04ns)

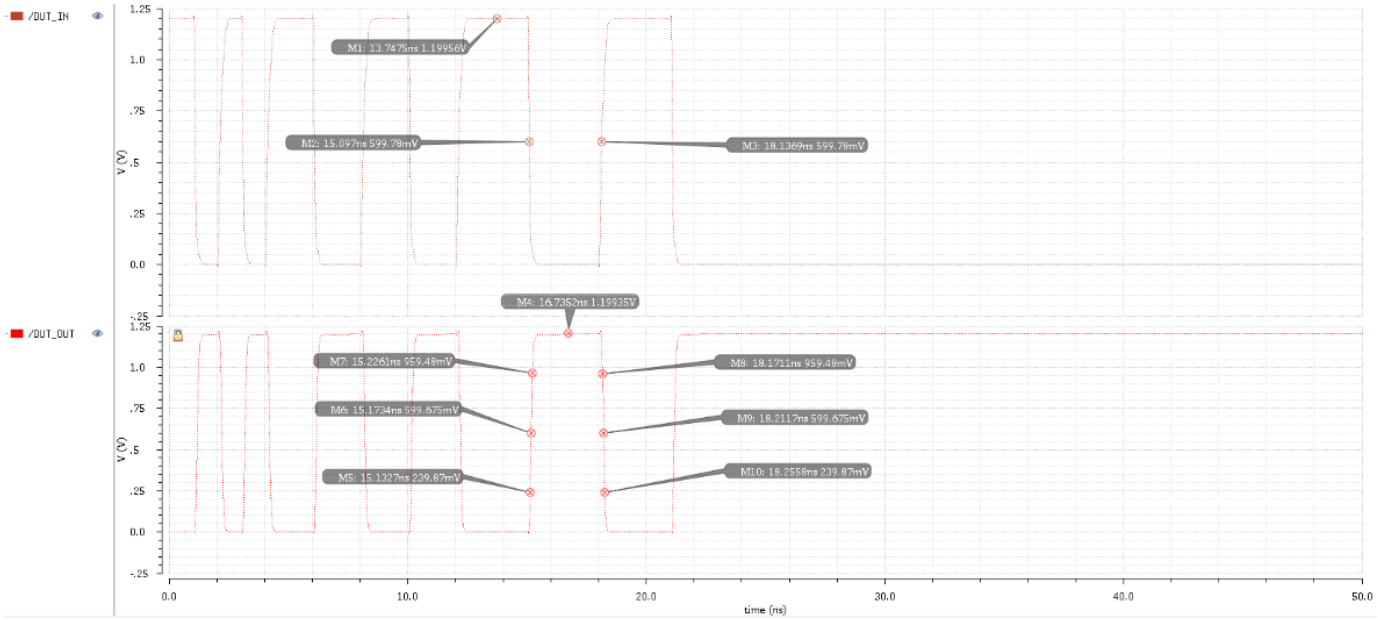

Figure 15: Transient Analysis of FO8 (for Input tr=0.04ns, tf=0.04ns)