Standard Cell Template

CMOS INVERTER

Haranadh Chintapalli, Satya Ravi Teja Erpina, Soma Sai Charitha

Group #5

6th Feb 2017

Introduction and Physical Properties

Cell Description

The cell consists of CMOS inverter as the DUT(Device Under Test). And it drives loads of its own circuit with Fanout values of - FO0, FO1, FO2, FO4, FO8.
The CMOS inverter is tested by varying the widths of PMOS and NMOS transistors to meet the specifications given:

Rise time(Tr) and Fall time(Tf) < 0.06ns and
Propagation Delays-High to Low(Tphl) & Low to High(Tplh) <0.07ns.

And the Input to the DUT is fed through a Non Inverting Buffer.

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

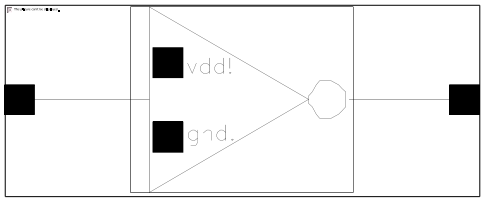

Figure 1: Example Logic Symbol from NCSU Digital Parts. All symbols are 1in tall.

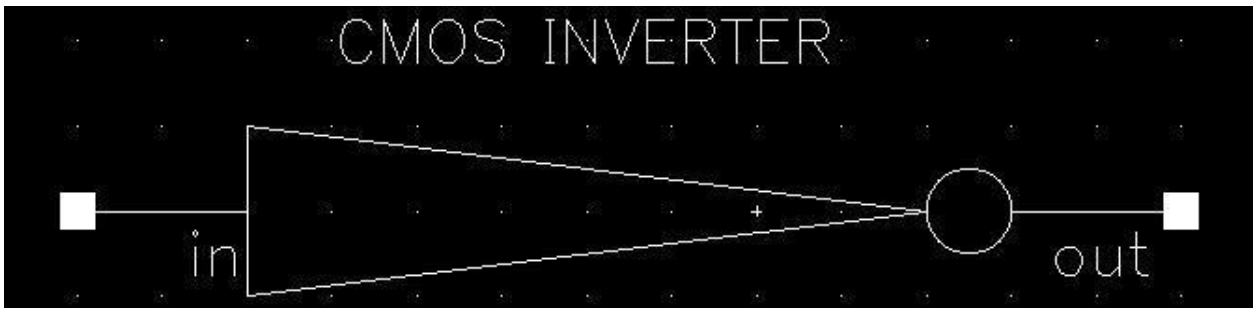

Figure 2: Symbol for DUT of Inverter

Standard Cell Template

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.

Input	Output
0	H
1	L

Table 1: Truth Table

Cell Logic Equation

$Y = \overline{A}$

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.

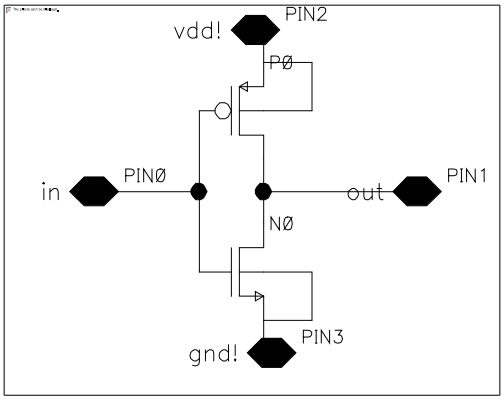

Figure 3: Example Schematic from NCSU CDK Publication Schematic. All schematic figure are 2in tall.

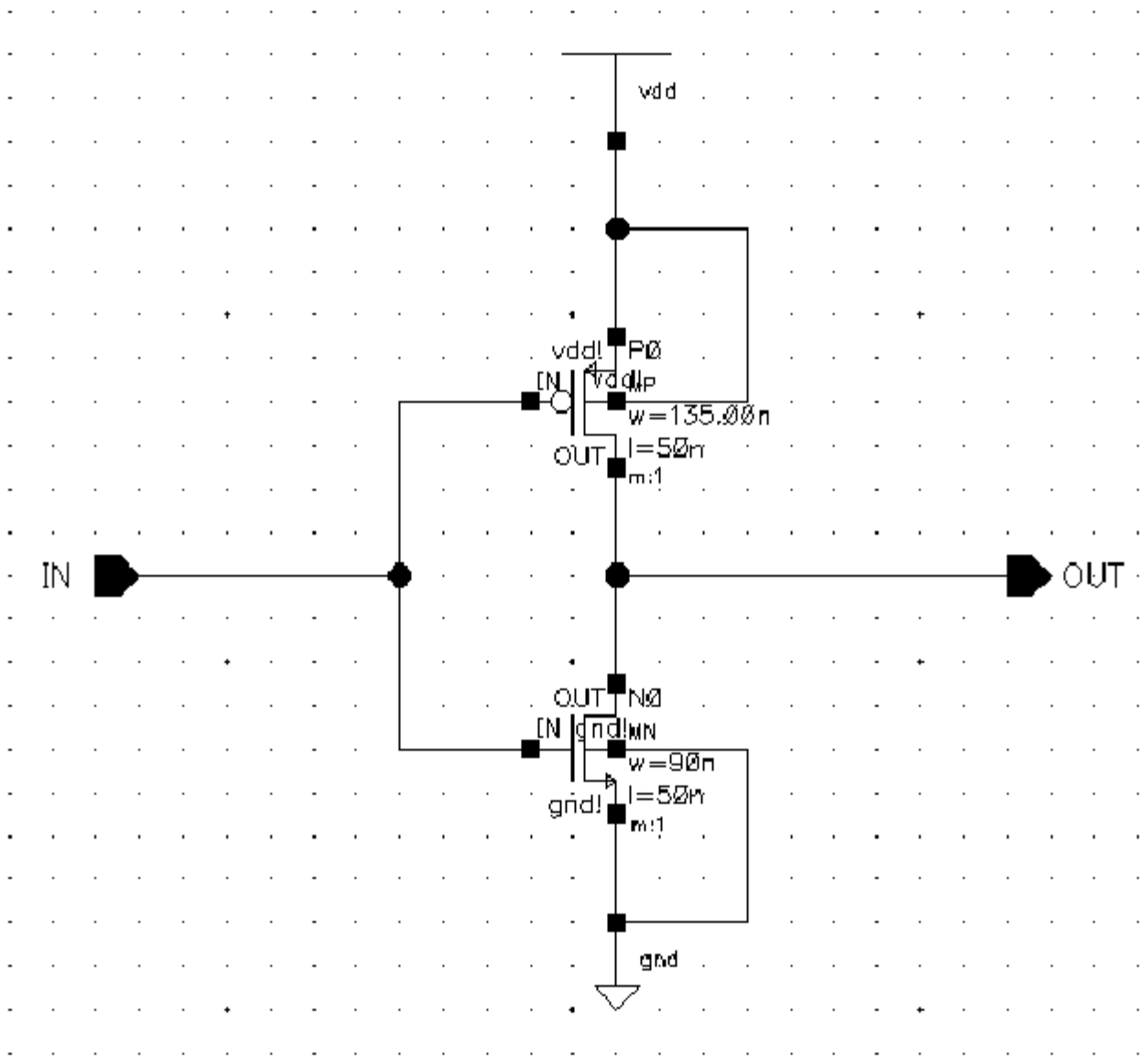

Figure 4: Transistor level of DUT

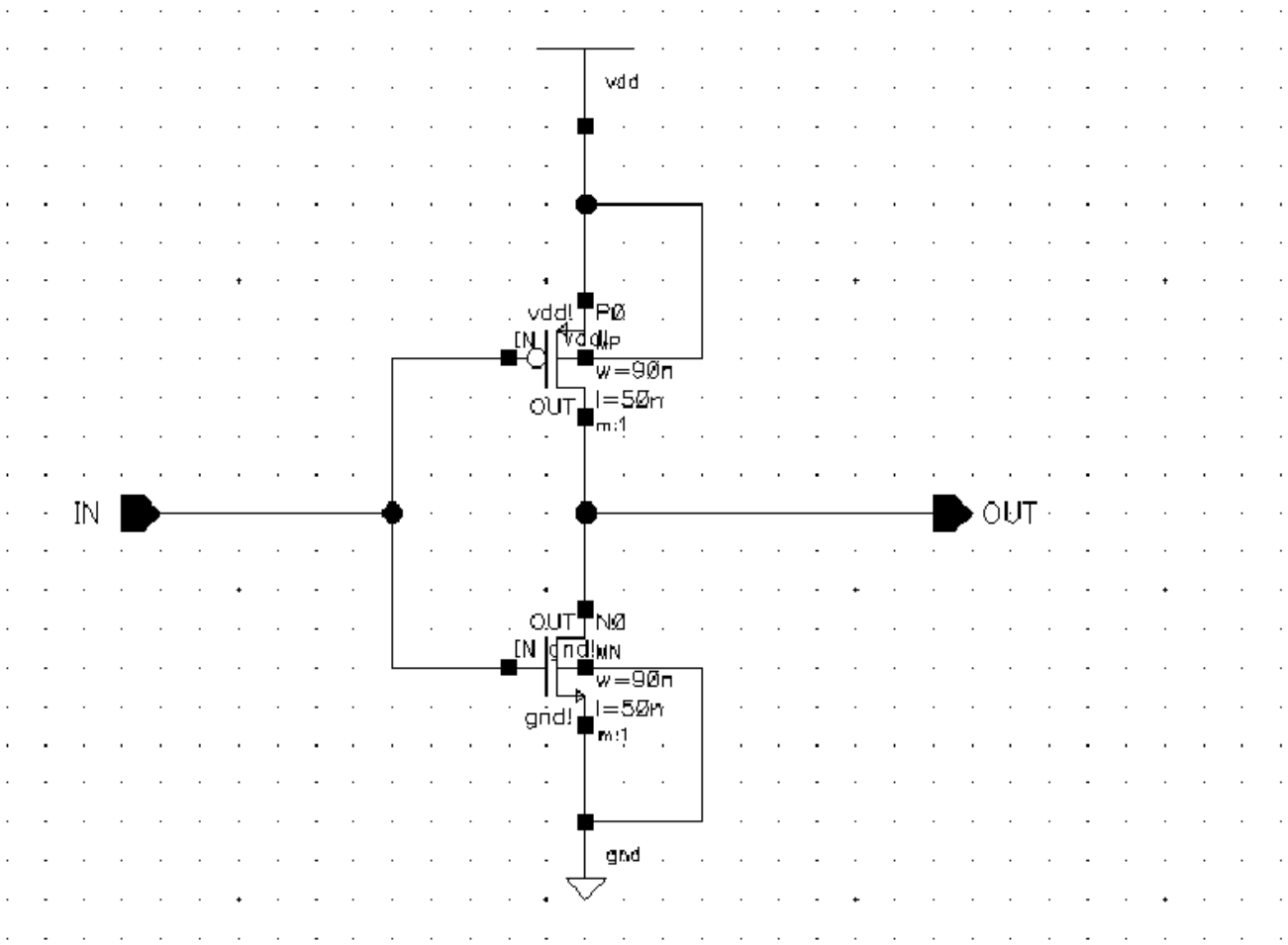

Figure 5: Transistor level of Load

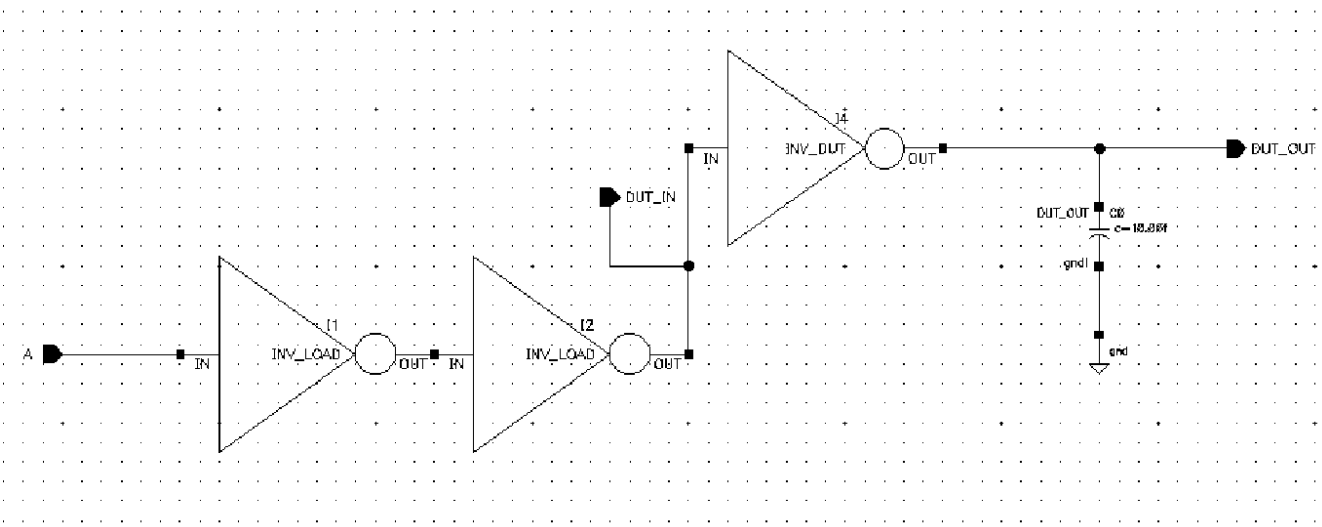

Figure 6: Schematic for FO0

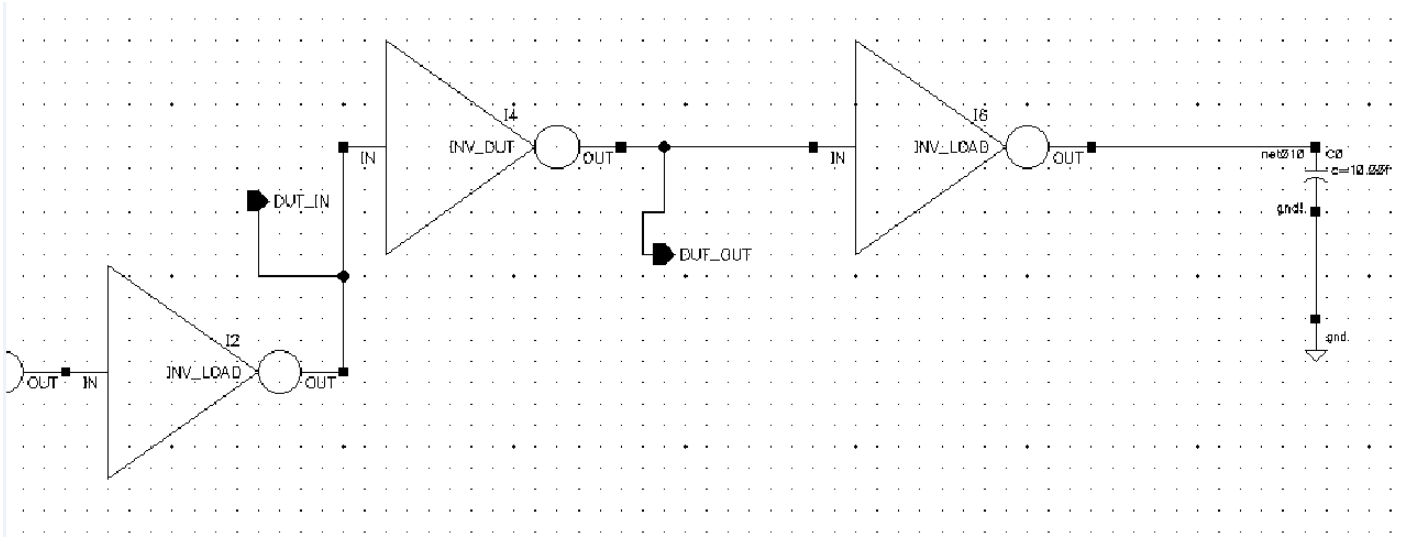

Figure 7: Schematic for FO1

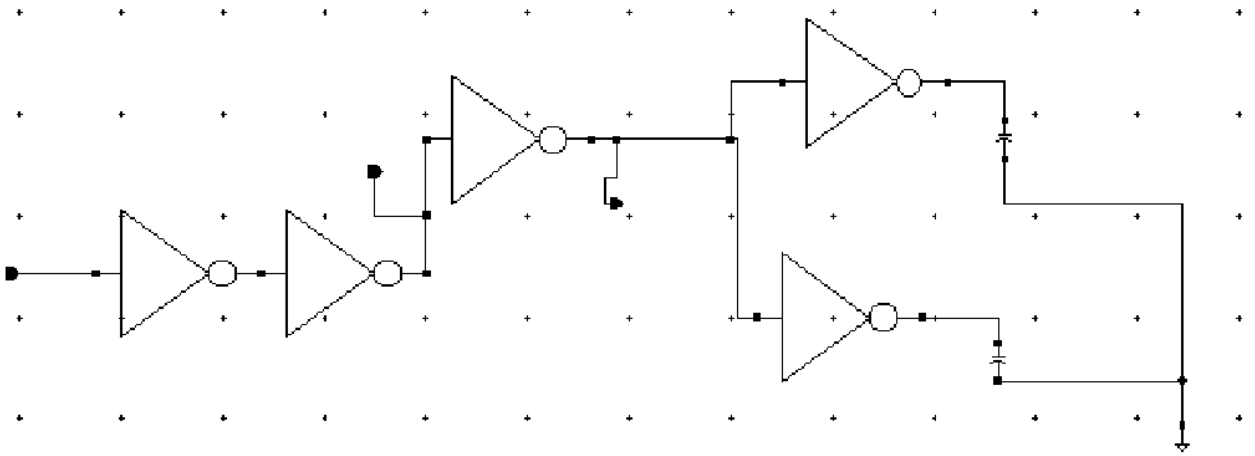

Figure 8: Schematic for FO2

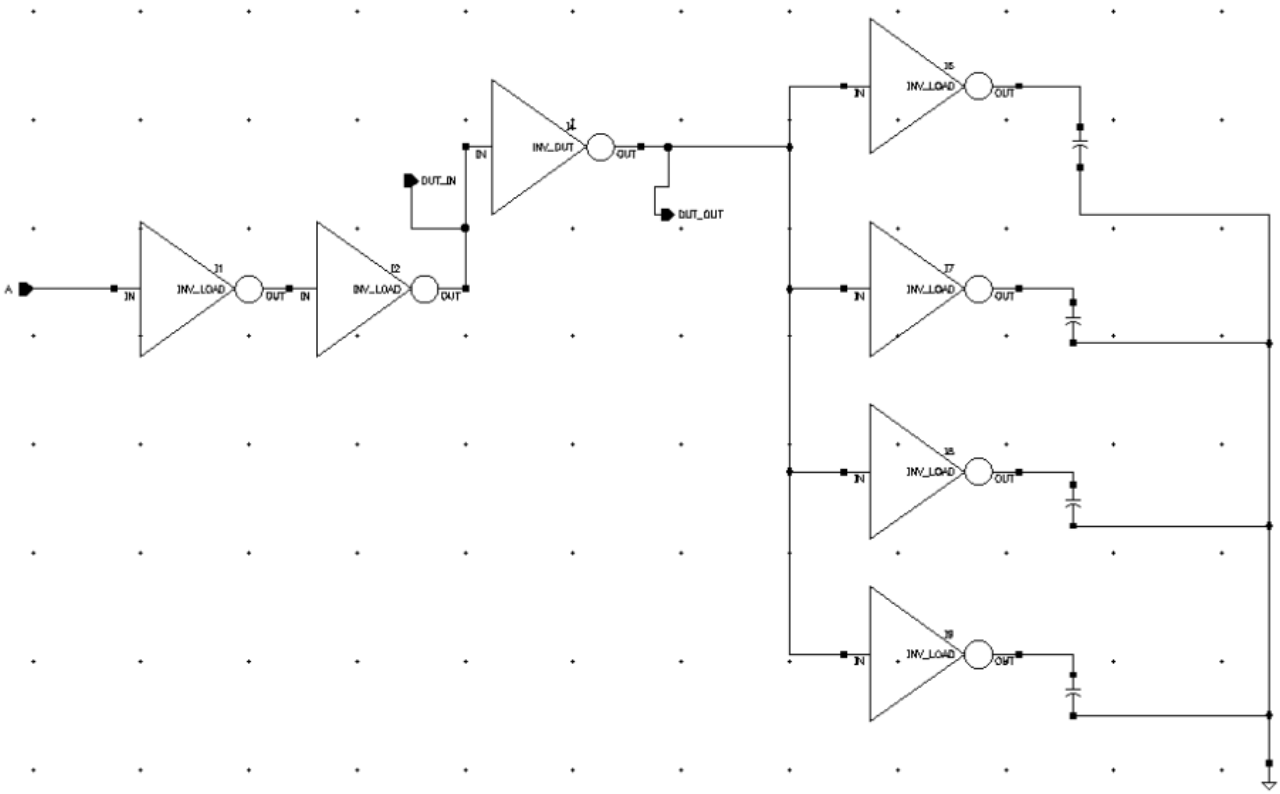

Figure 9: Schematic for FO4

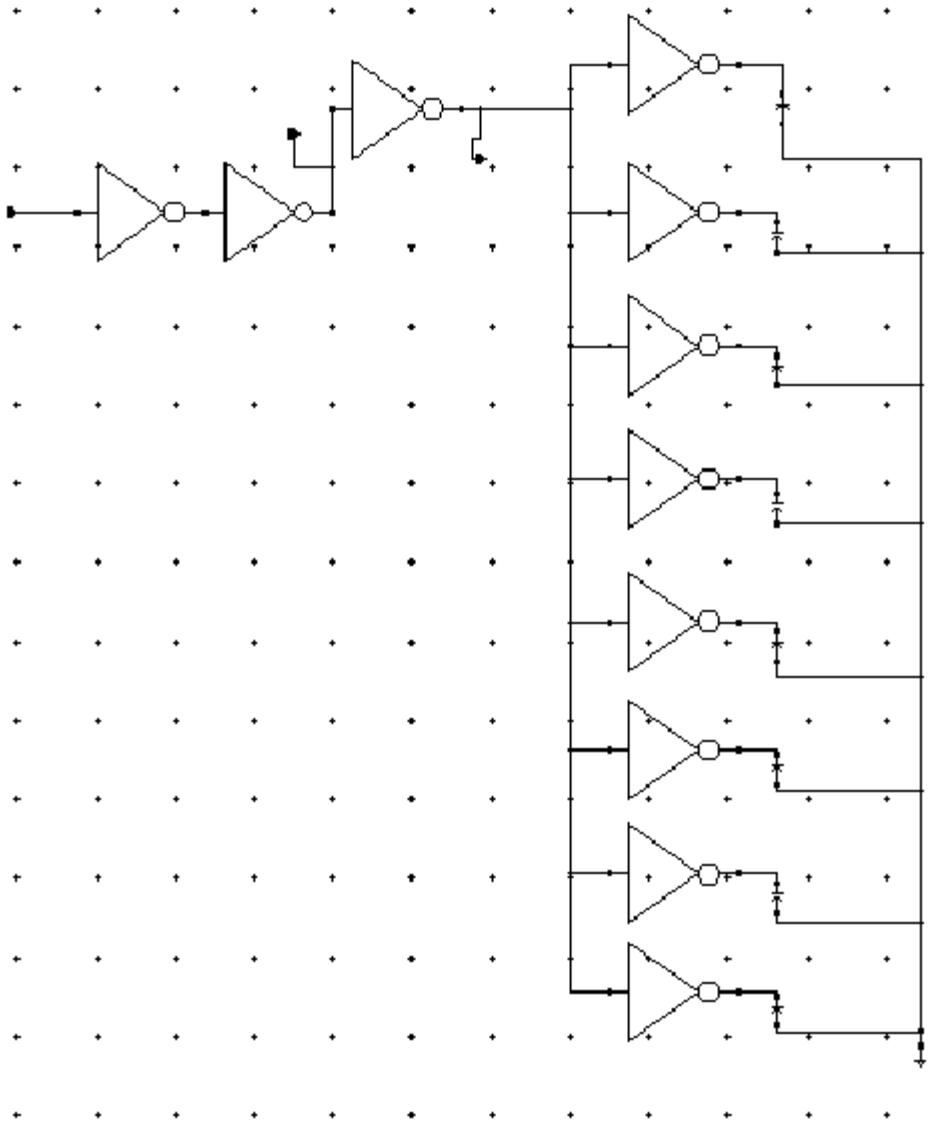

Figure 10: Schematic for FO8

Standard Cell Template

Cell Layout Diagram and Dimensions

Save a color 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.]

Cell Physical Dimensions		
	X	Y
Cell Dimension in □□		
Cell Dimension in □m		
Transistor Dimensions		
Transistor Instance Number	Length (nm)	Width (nm)
PMOS	50	135
NMOS	50	90

Table 2: Dimensions for DUT (CMOS Inverter)

Cell Physical Dimensions		
	X	Y
Cell Dimension in □□		
Cell Dimension in □m		
Transistor Dimensions		
Transistor Instance Number	Length (nm)	Width (nm)
PMOS	50	90
NMOS	50	90

Table 3: Dimensions for Unit Inverter (Loads)

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.

Standard Cell Template

[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)**
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.

Standard Cell Template

Input X: Output Rise Time Data t_r (ns)					
Input rise/fall time (ns)	Output Load (FOx) 0	1	2	4	8
0.04	0.2161	0.0234	0.0313	0.0484	0.0851
0.06					

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	0.1489	0.022	0.0273	0.0384	0.0613
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.

Standard Cell Template

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	0.1515	0.0229	0.0285	0.0393	0.0609
0.06					

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

Data Worst Case High to Low Propagation Delay Data t_{phl} (ns)					
Input rise/fall time (ns)	Output Load (FOx) 0	1	2	4	8
0.04	0.1179	0.02124	0.0261	0.0356	0.053
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 labeled **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.

Standard Cell Template

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

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*** $p_{inv} \approx 1$.

Inverter Normalized Data Model Propagation Delay Equation $D(h) = p + g \cdot h$				
Input Slew Rate (ns)	Rising Logical Effort (g_r)	Falling Logical Effort (g_f)	Parasitic Rising Delay (p_r)	Parasitic Falling Delay (p_r)
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)					
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					

LAB 2					
CMOS INVERTER-FO4, input tr=0.05ns,tf=0.05ns					
DUT		Rise/Fall and delay time			
Wp	Wn	Tr(ns)	Tf(ns)	Tplh(ns)	Tphl(ns)
90nm	90nm	0.0717	0.0366	0.0543	0.0331
135nm	135nm	0.0503	0.0302	0.0416	0.027
180nm	180nm	0.0408	0.028	0.0363	0.0237
135nm	90nm	0.0478	0.0385	0.0394	0.0356
90nm	135nm	0.0748	0.0281	0.0575	0.025
135nm	180nm	0.052	0.0268	0.0439	0.0223
180nm	135nm	0.0388	0.0313	0.0344	0.0288

Justification for the Widths chosen:

In the simulations implemented for the CMOS INVERTER_DUT, as the gate width of the MOS transistors changes, it has direct influence on the Propagation delays, Rise and Fall times.

Based on the results table, as the gate width of the PMOS transistor (W_p) increases, the Low to High Propagation Delay(Tplh), Rise time (T_r) decreases.
While, as the gate width of the NMOS transistor (Wn) increases, the Tphl , Tf values decreases.

Hence as the size of gates changes, the speed of the gate switching is affected. Hence for having good performance of speed, due to fabrication constraints, it is not feasible to always increase the

widths and make large sized transistors. Instead we should also consider other scaling options like voltage scaling, channel length scaling.

Upon simulating the Inverter_DUT with various values of Widths(Wp, Wn), the PMOS transistor width of Wp=135nm, NMOS transistor width of Wn=90nm meets the specification required precisely. Also, it is noted that larger values of widths also met the specifications(i.e.T_{Plh} and T_{Phl} < 0.07ns and T_r and T_f < 0.06ns) but those are not effective at the time of fabrication. Hence, inorder to meet the specification mentioned, we set the width of the PMOS transistor as Wp=135nm and the width of the NMOS transistor as Wn=90nm.

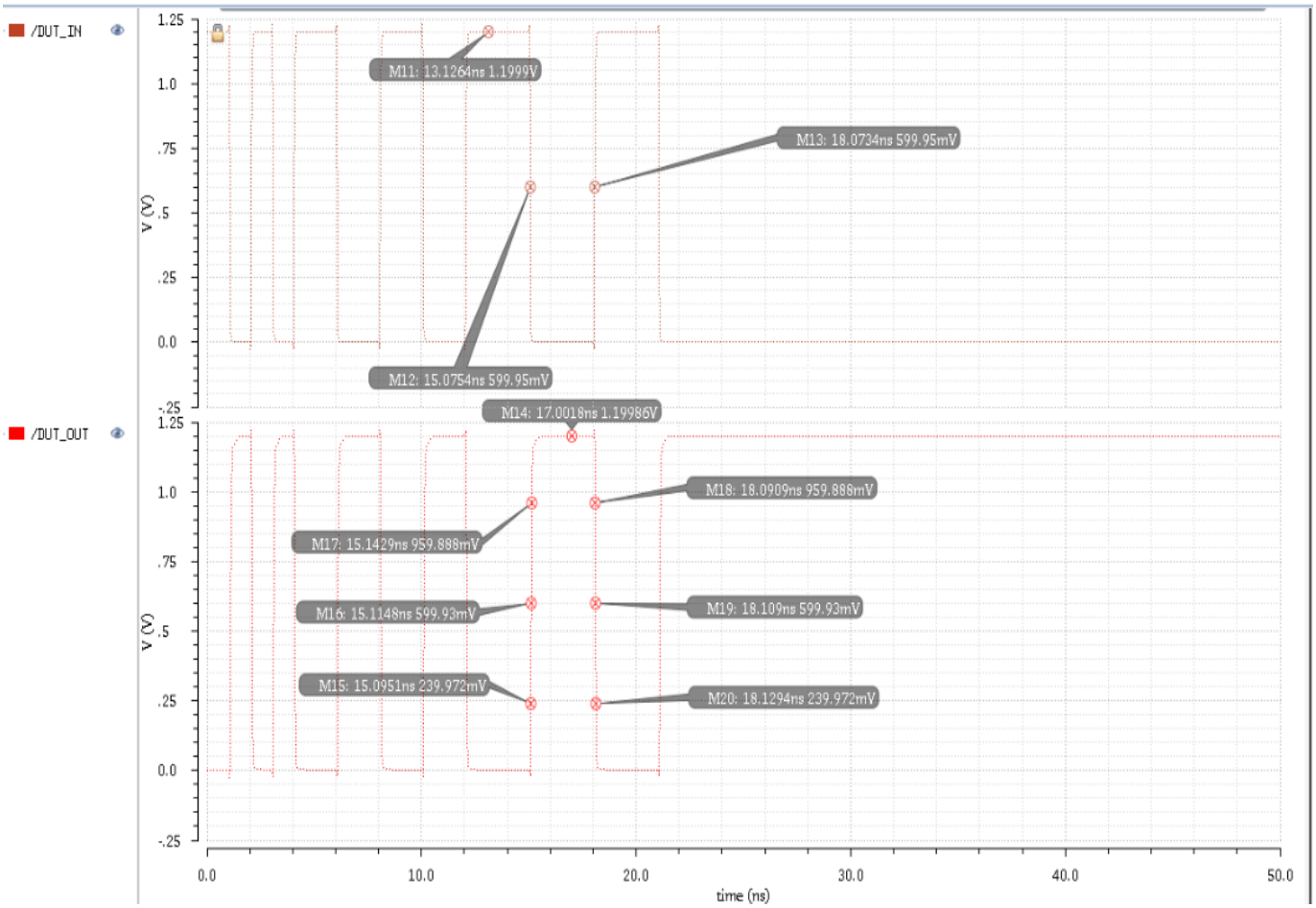

Figure 11: Transient Analysis for Inverter with Pmos-135 and N-mos-90 dimensions (for input tr=0.05ns,tf=0.05ns)

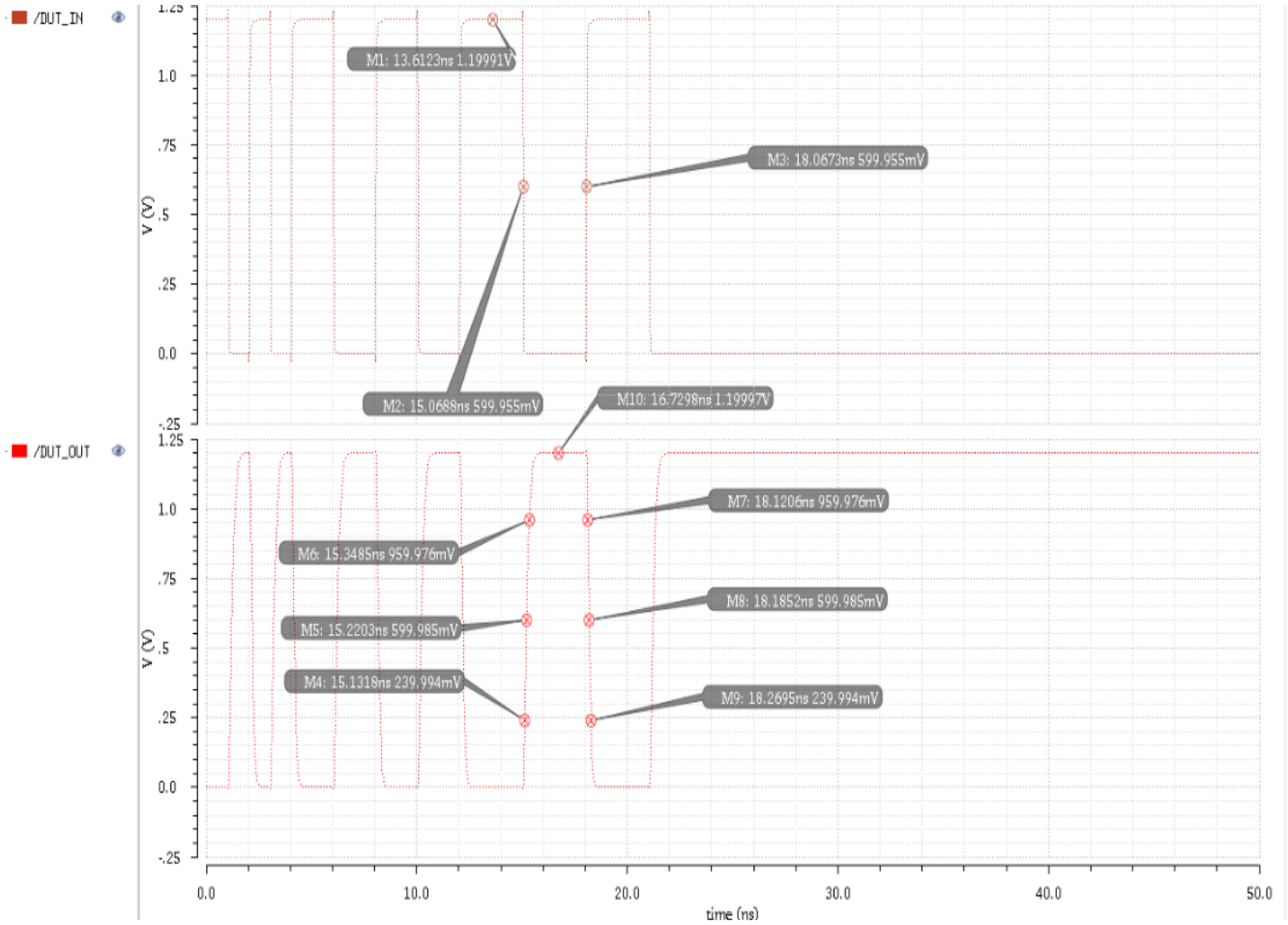

Figure 12: Transient Analysis for FO0 (input tr=0.04ns,tf=0.04ns)

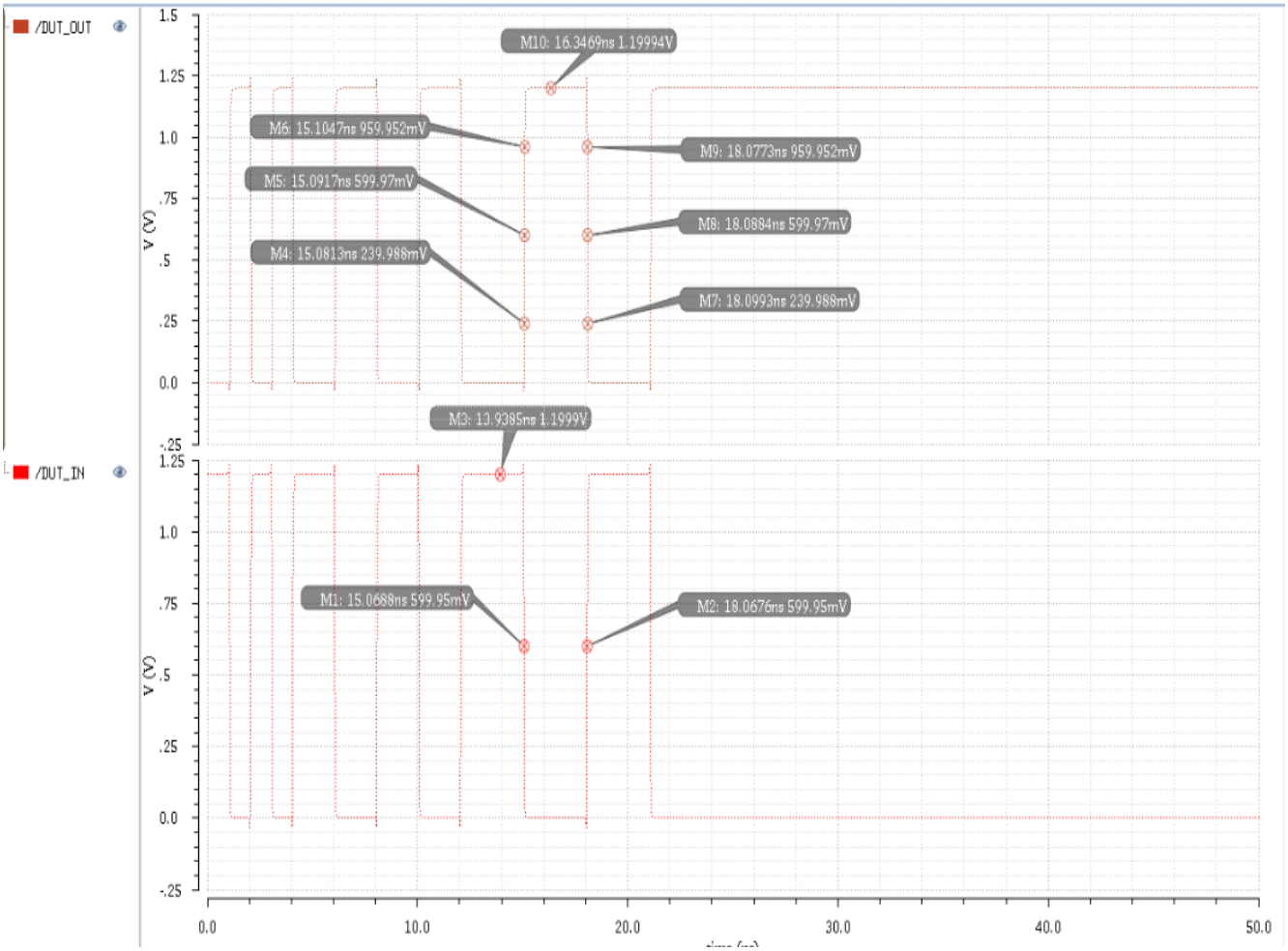

Figure 13: Transient Analysis for FO1 (input tr=0.04ns,tf=0.04ns)

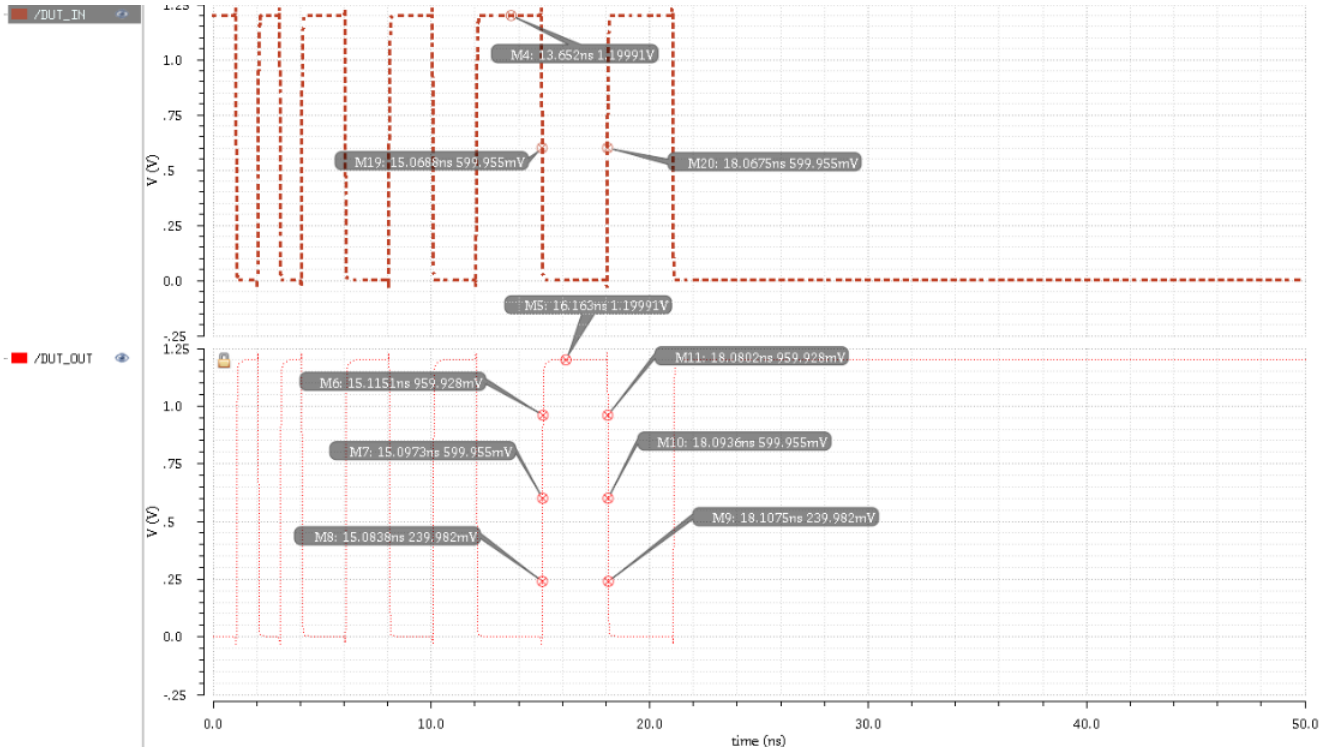

Figure 14: Transient Analysis for FO2 (input tr=0.04ns,tf=0.04ns)

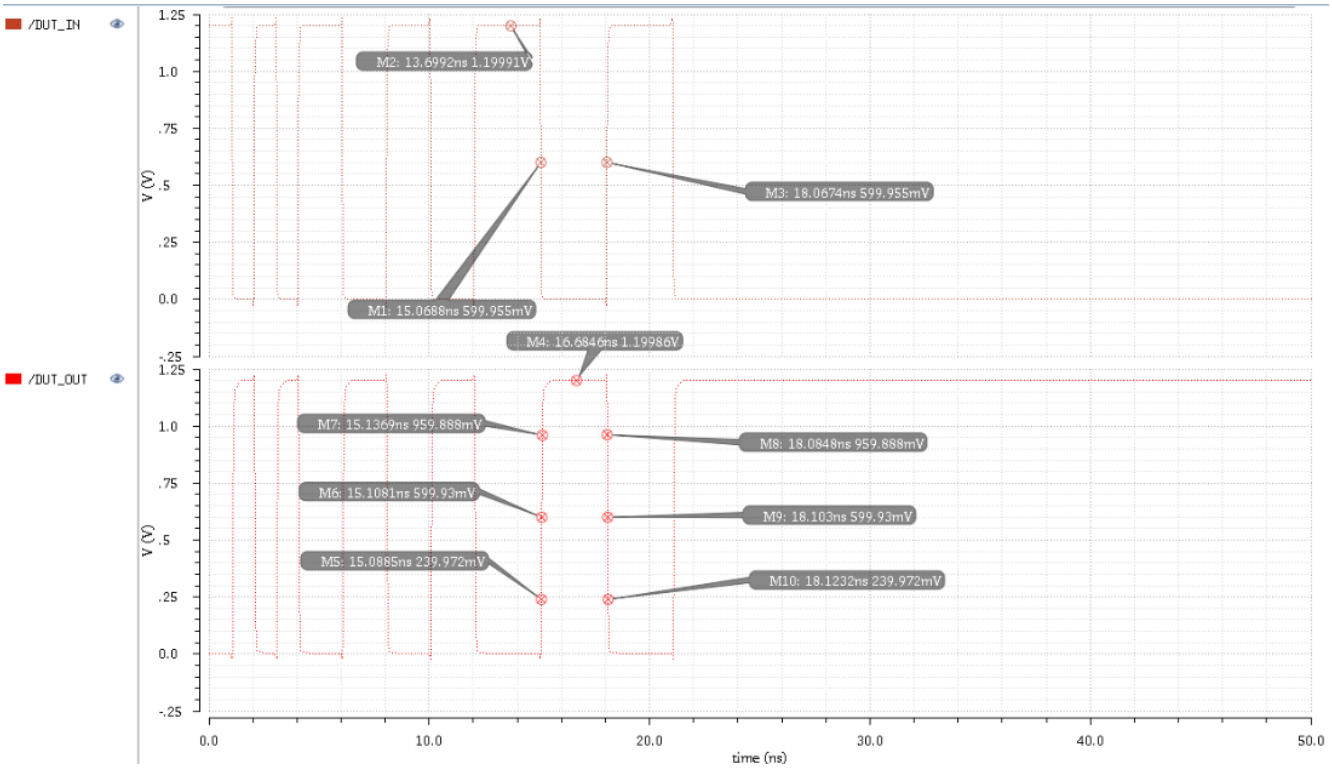

Figure 15: Transient Analysis for FO4 (input tr=0.04ns,tf=0.04ns)

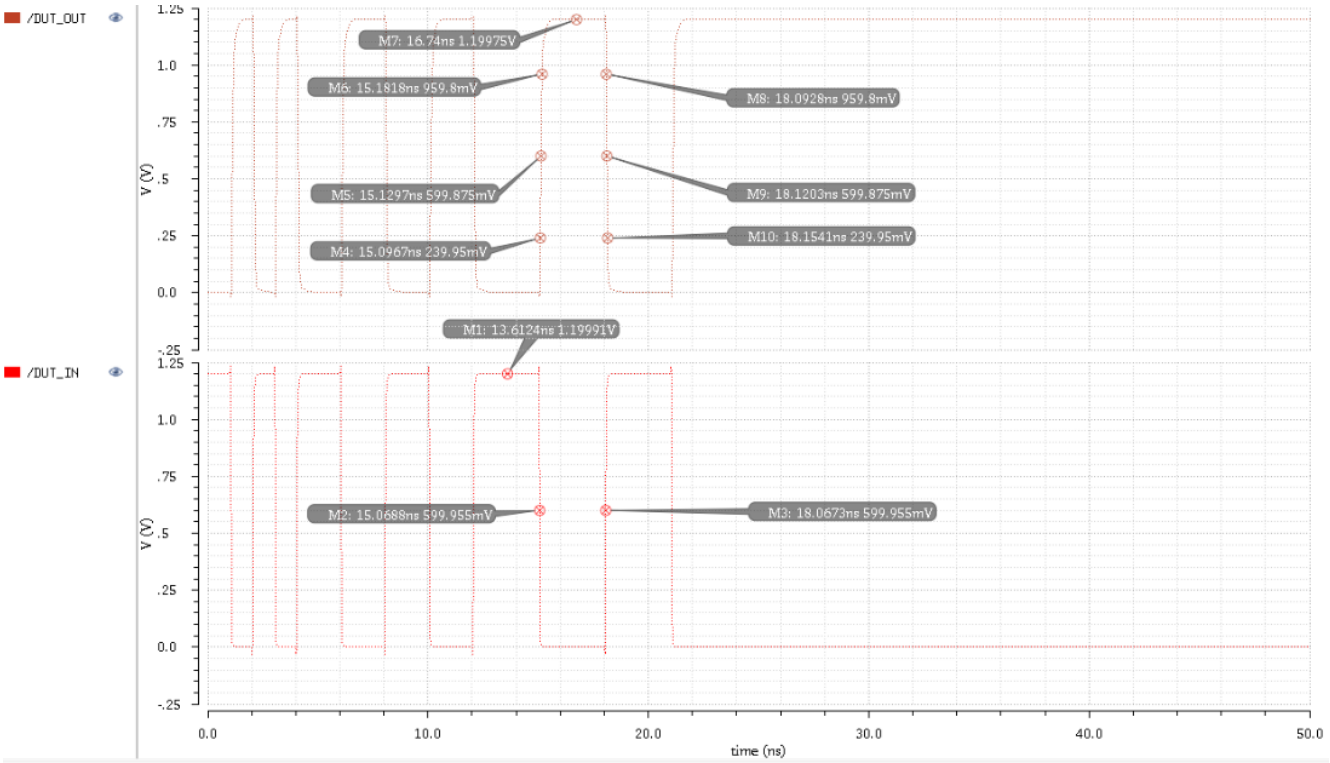

Figure 16: Transient Analysis for FO8 (input tr=0.04ns,tf=0.04ns)