ECE520 Introduction to VLSI, Project 2

Name: Arijit Sengupta, ID: 001441748

PART A

Truth Table of Full Adder:

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Therefore, the **Boolean expressions** for S and Cout are given by –

S = A'B'Cin+A'BCin'+AB'Cin'+ABCin = Cin(AB+A'B') + Cin'(AB'+A'B) = Cin(A'B+A'B)' + Cin'(AB'+A'B)

S = XOR (Cin, XOR (A, B))

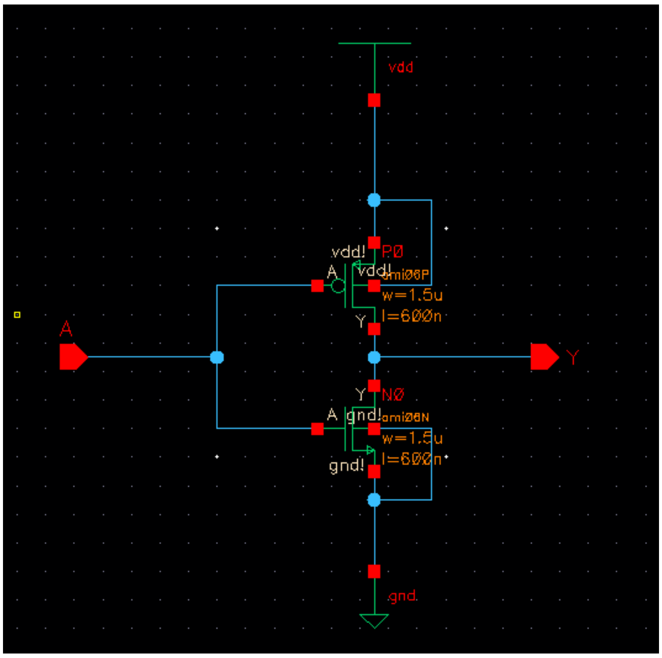
Cout = A'BCin+AB'Cin+ABCin'+ABCin = Cin(A'B+AB')+AB(Cin+Cin') = Cin(A'B+AB') + AB

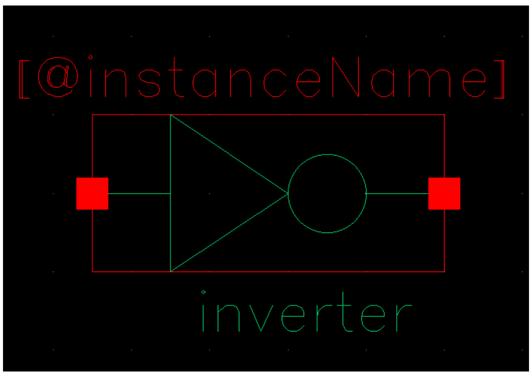
Cout = NOT (NOR (NOT (NAND (A, B)), NOT (NAND (Cin, XOR (A, B))))

The above expressions represent the functions using the NOR, NAND, Inverter (NOT), XOR and XNOR gates only, as per the requirement.

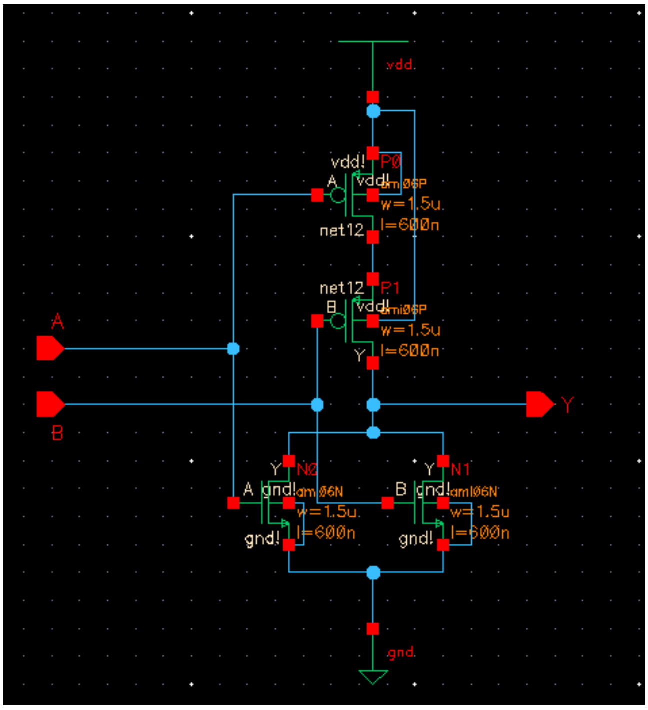
Schematic and Symbols:

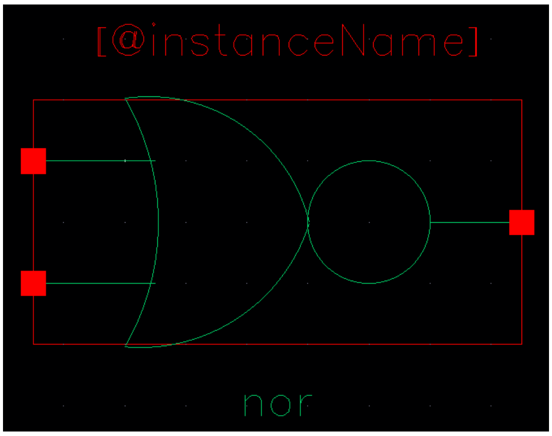
1. **Inverter (NOT Gate)** –



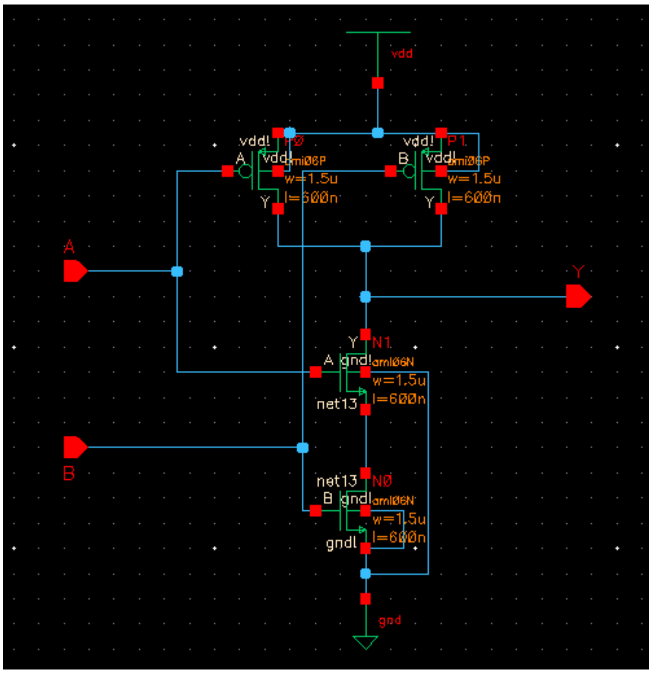


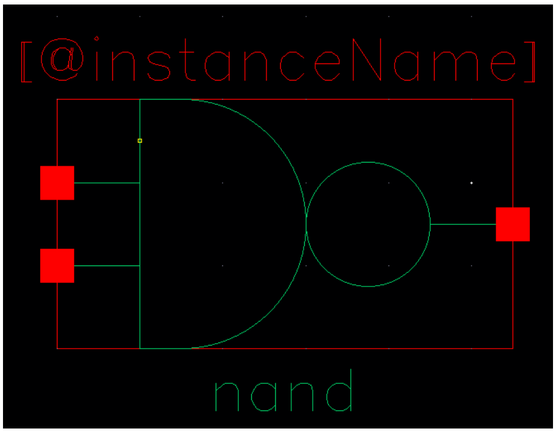
2. **NOR Gate** –



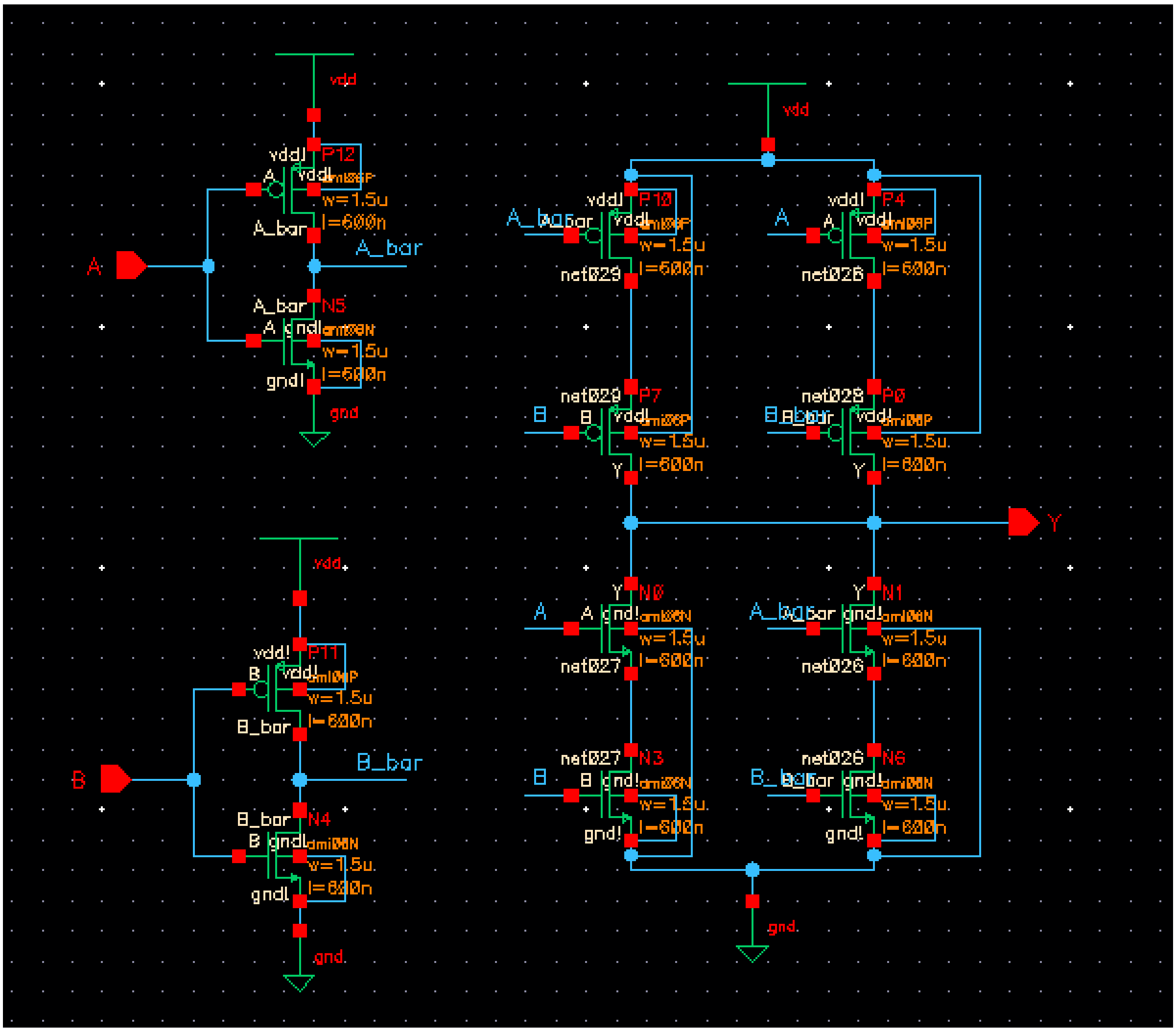


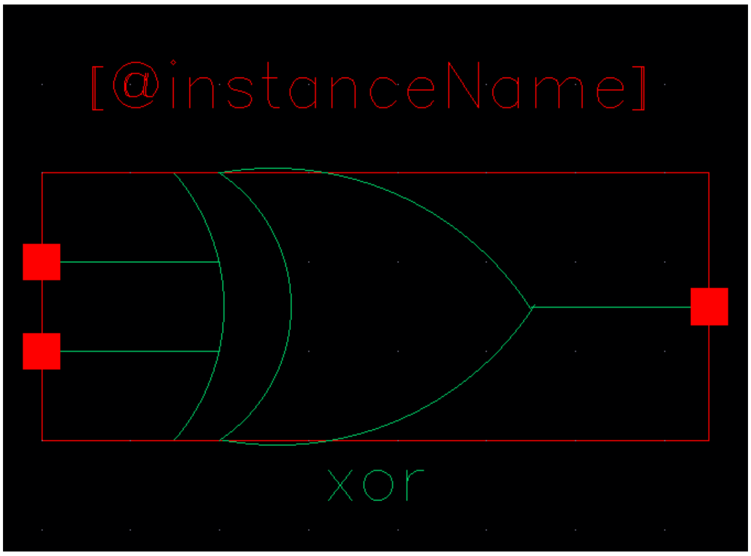
3. **NAND Gate** –



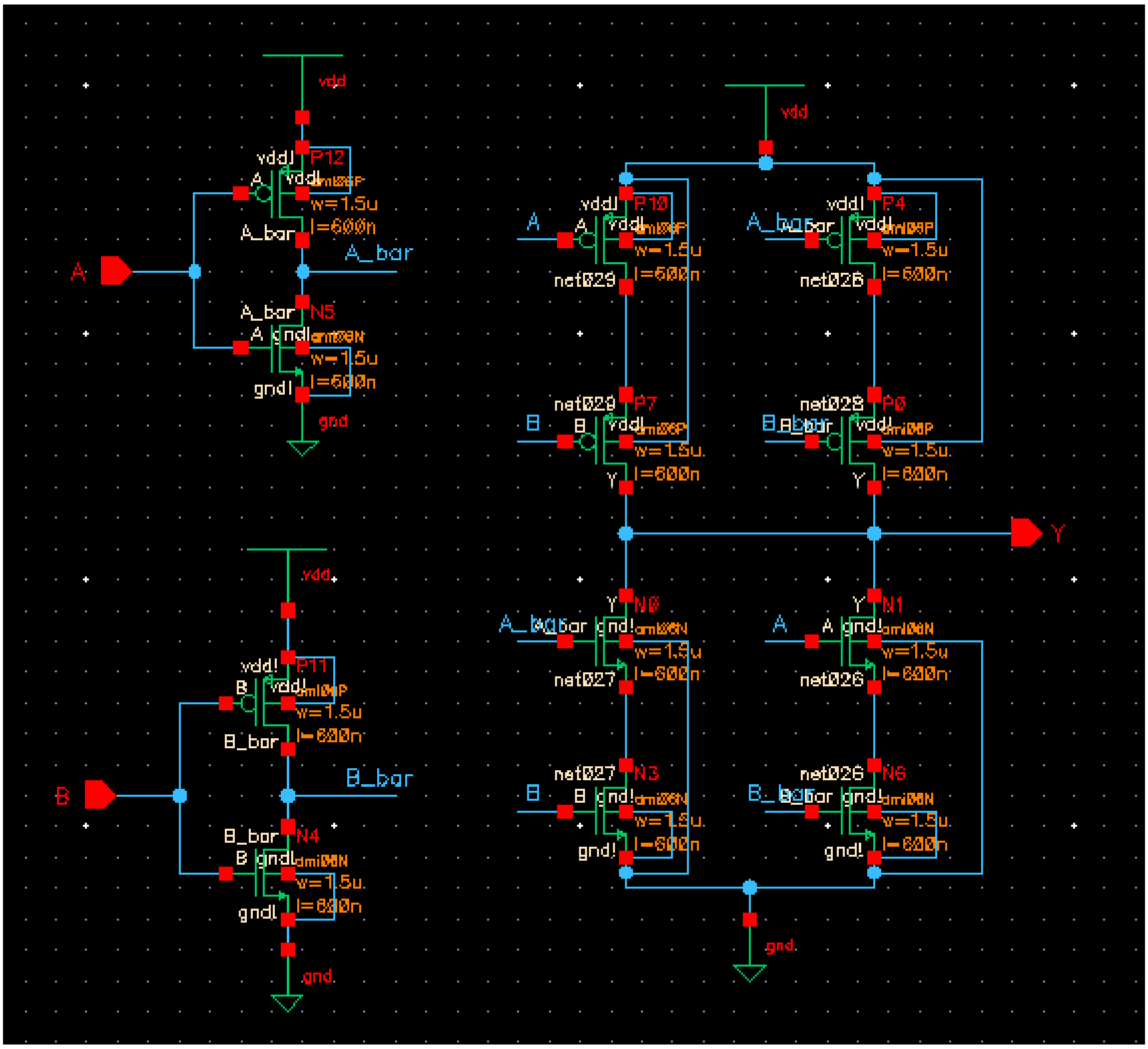


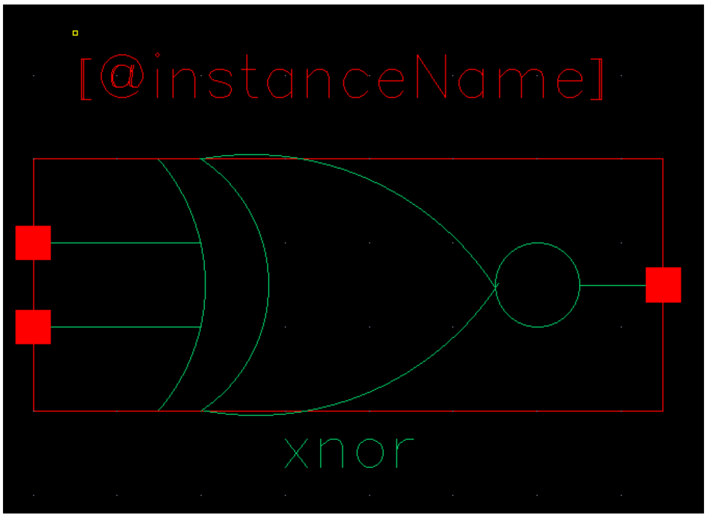
4. **XOR Gate** –


vdd
P12
vdd!
A
A_bar
w=1.5u
l=600n
N5
gnd!
gnd
P11
B
B_bar
N4
P10
P4
net029
net028
P7
P8
Y
N0
N1
net027
net026
N3
N6

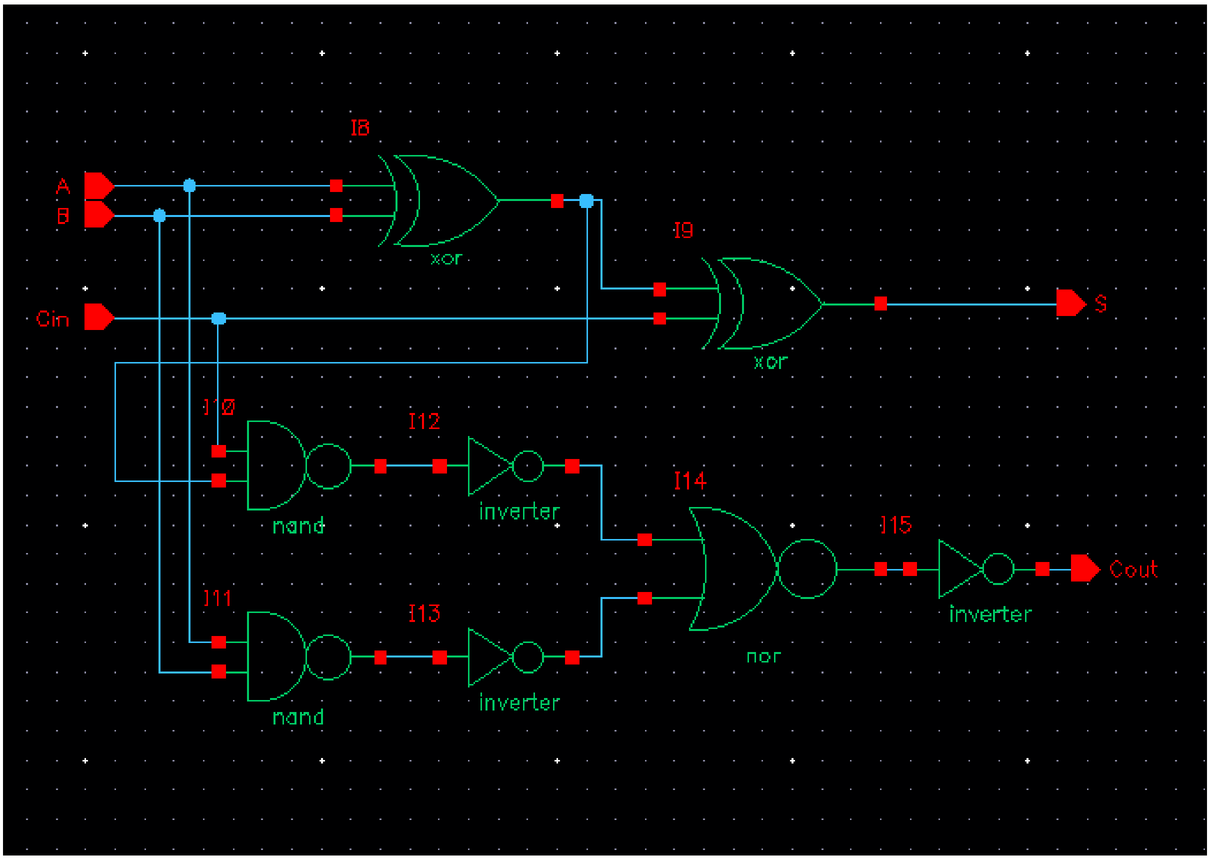

[@instanceName]
xor

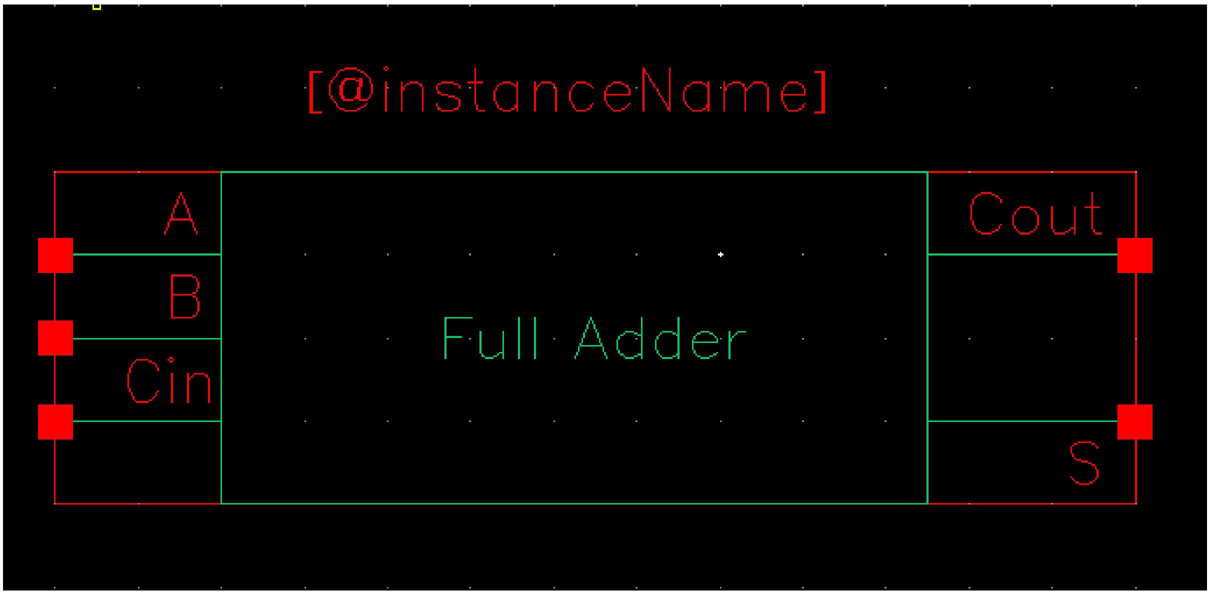
5. **XNOR Gate** –





Full Adder Schematic and Symbol:





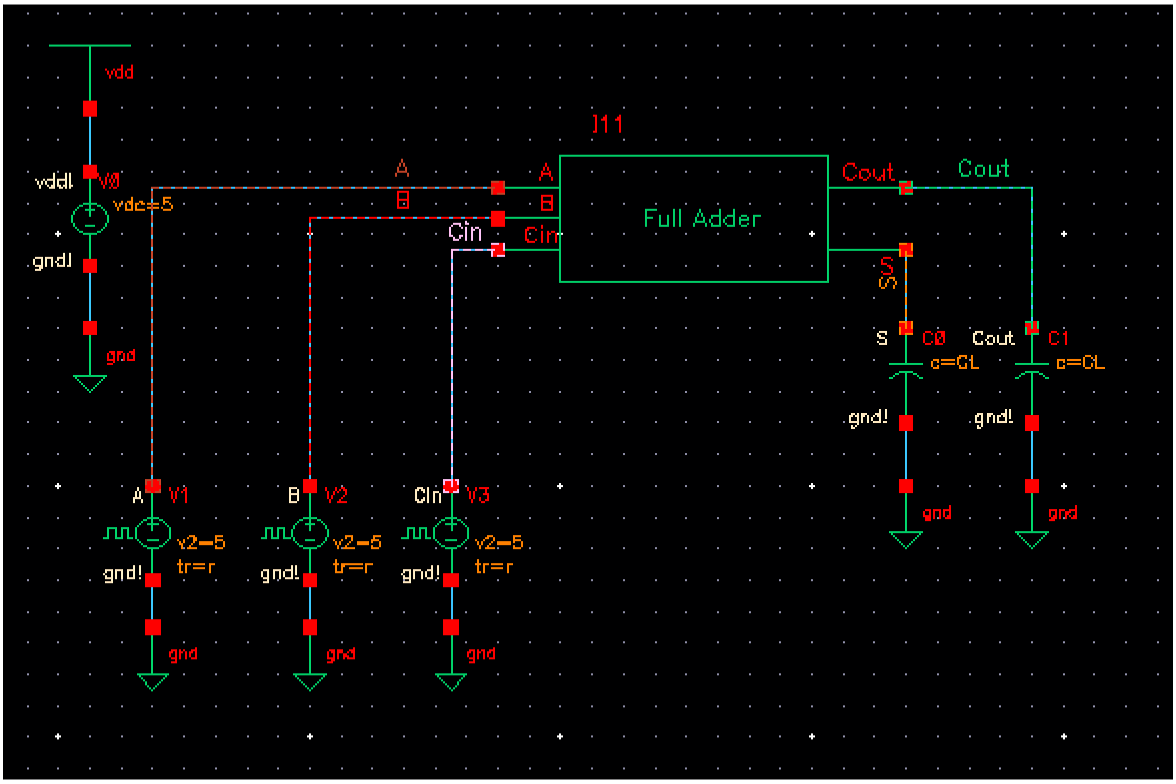
The FullAdder is run in a test bench named **TBFullAdder**.

Parameters for analysis:

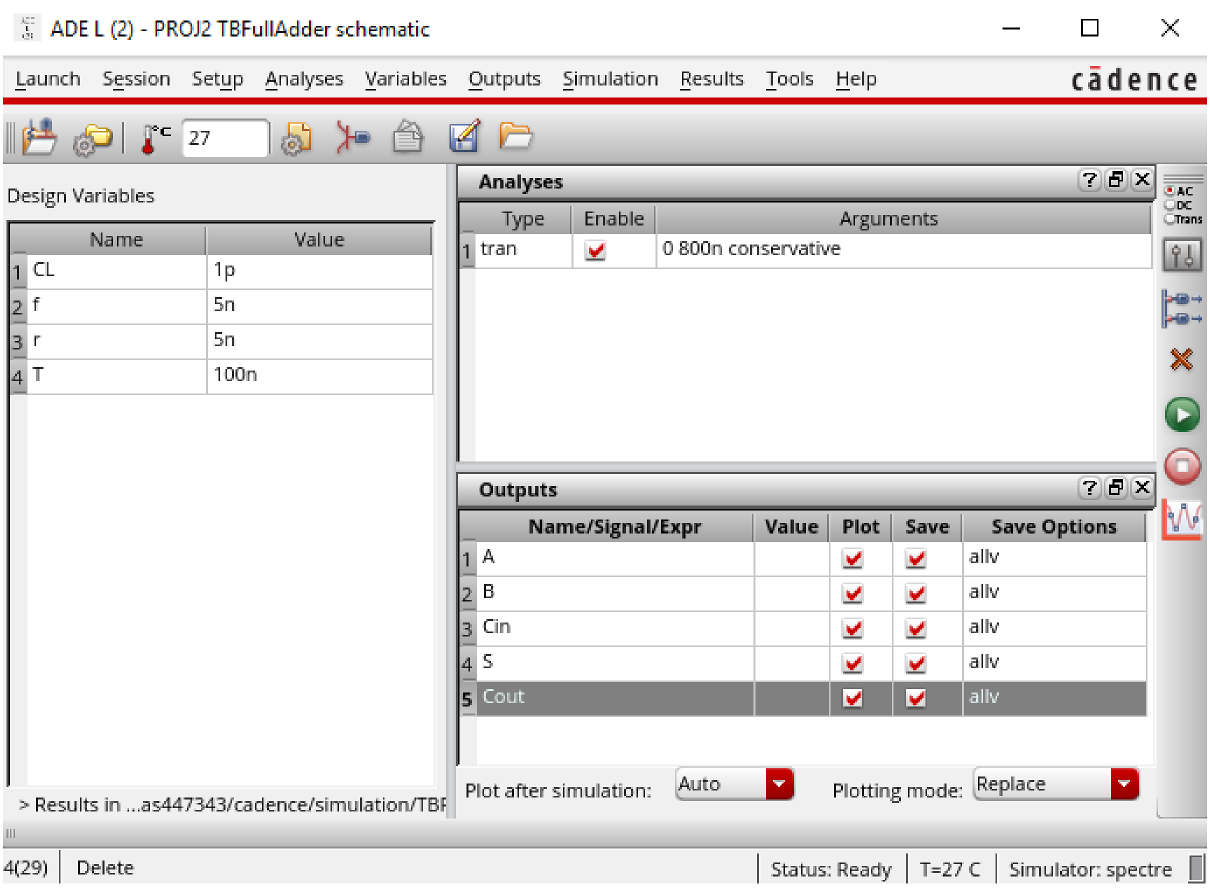
Times for vpulse – **input A**: Delay=1ns, Rise time = r, Fall time = f, Pulse width = (T-r-f)/2, Period = T; **input B**: Delay=1ns, Rise time = r, Fall time = f, Pulse width = (2*T-r-f)/2, Period = 2*T.

CL=1E-12 (1 pF), **T**=100E-9 (100 ns), **r**=5E-9 (5ns), **t**=5E-9 (5ns).

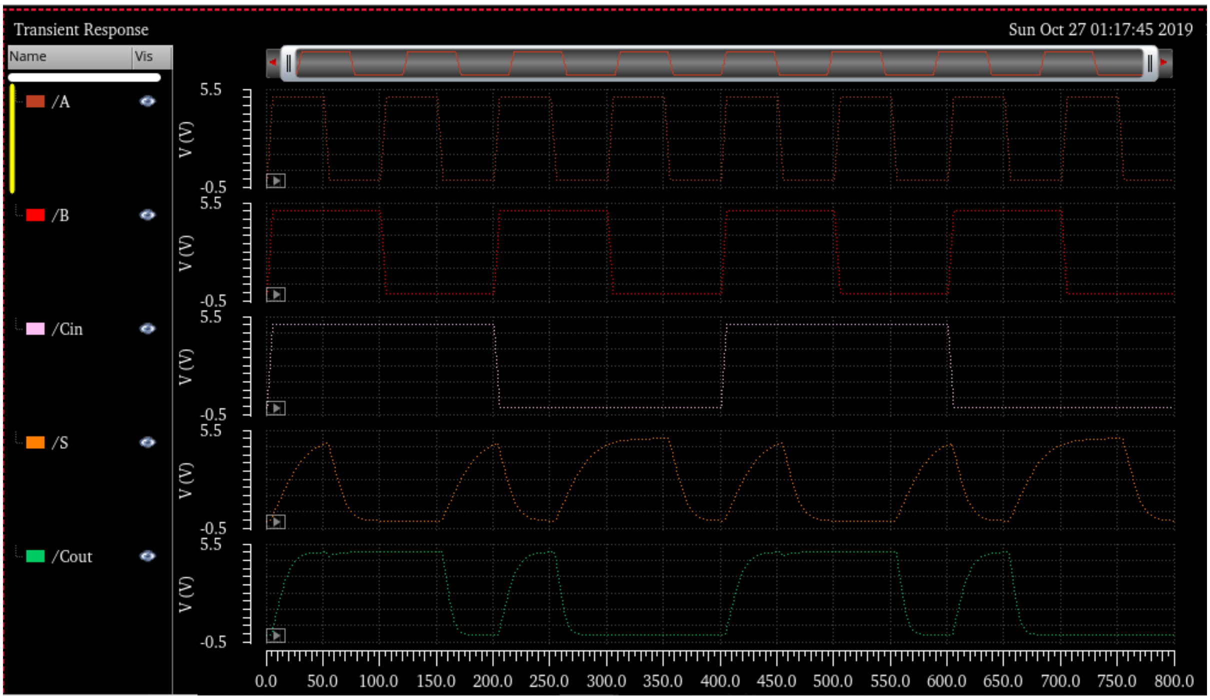
TBFullAdder Schematic:



The circuit is run using the following values:



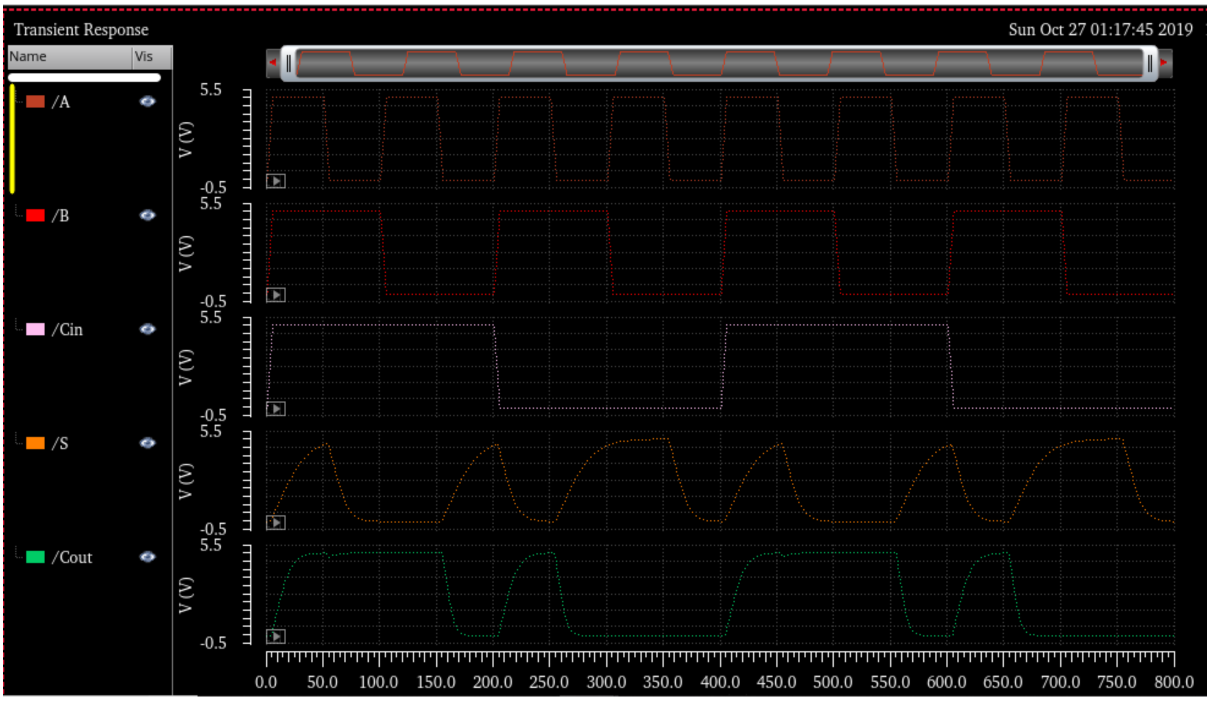
Waveform of A, B, Cin inputs and S, Cout outputs:



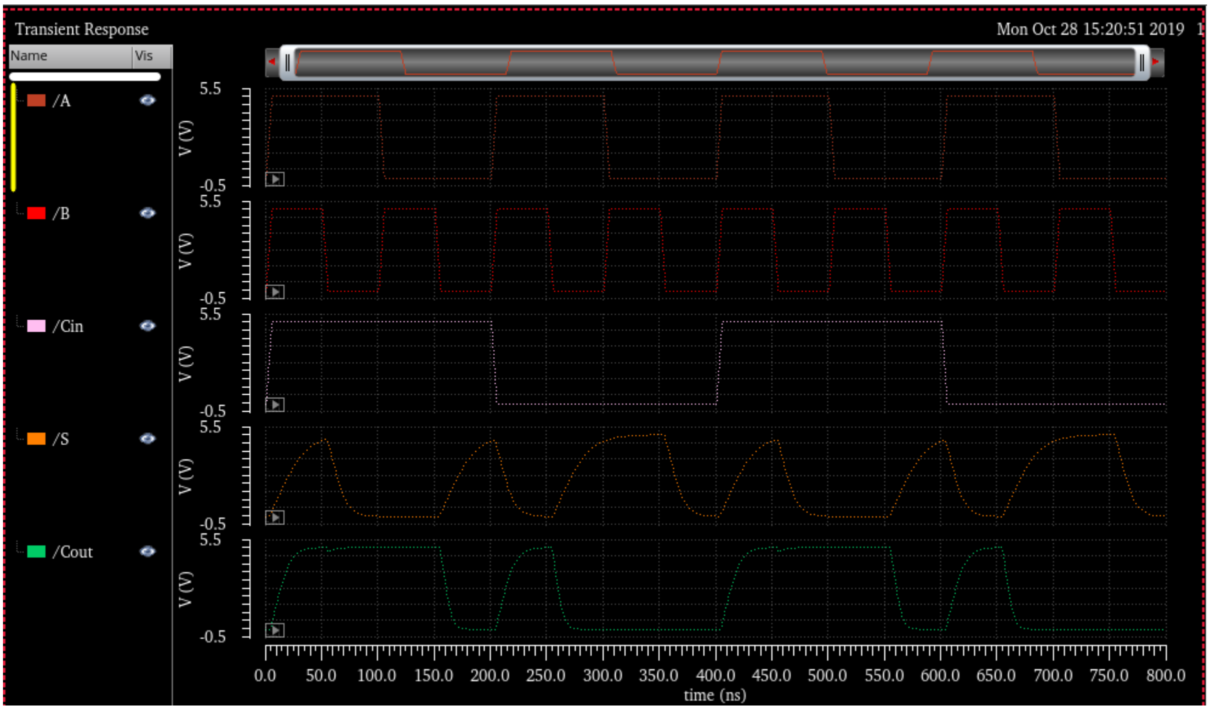
PART B

We modify the inputs A & B in order to get all the required transition delays for this part.

Waveform 1 - Time periods: A= T s, B = 2T s, Cin = 4T s



Waveform 2 - Time periods: A= 2T s, B = T s, Cin = 4T s



These two waveforms are enough to give us the required 12 propagation delay values.

The measured propagation delays using Calculator are:

Delay	Expression	Value (ns)
tpdr (A --> S)	17.17E-9	17.17ns
tpdr (B --> S)	17.17E-9	17.17ns
tpdr (Cin --> S)	17.17E-9	17.17ns
tpdr (A --> Cout)	10.01E-9	10.01ns
tpdr (B --> Cout)	10.01E-9	10.01ns
tpdr (Cin --> Cout)	10.01E-9	10.01ns
tpdf (A --> S)	9.938E-9	9.938ns
tpdf (B --> S)	10.26E-9	10.26ns
tpdf (Cin --> S)	9.221E-9	9.221ns
tpdf (A --> Cout)	7.37E-9	7.37ns
tpdf (B --> Cout)	7.331E-9	7.331ns
tpdf (Cin --> Cout)	10.02E-9	10.02ns

The Calculator stack:

From Waveform 1 -

```
delay(?wf1 v("/Cin" ?result "tran"), ?value1 2.5, ?edge1 "falling", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 v("/Cout" ?result "tran"), ?value2 2.5, ?edge2 "rising", ?nth2 2, ?tol2 nil,  ?td2 nil , ?stop 5, ?multiple nil)
delay(?wf1 v("/A" ?result "tran"), ?value1 2.5, ?edge1 "falling", ?nth1 2, ?td1 0.0, ?tol1 nil, ?wf2 v("/Cout" ?result "tran"), ?value2 2.5, ?edge2 "falling", ?nth2 1, ?tol2 nil,  ?td2 nil , ?stop 5, ?multiple nil)
delay(?wf1 v("/Cin" ?result "tran"), ?value1 2.5, ?edge1 "falling", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 v("/S" ?result "tran"), ?value2 2.5, ?edge2 "falling", ?nth2 2, ?tol2 nil,  ?td2 nil , ?stop 5, ?multiple nil)
delay(?wf1 v("/A" ?result "tran"), ?value1 2.5, ?edge1 "falling", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 v("/S" ?result "tran"), ?value2 2.5, ?edge2 "falling", ?nth2 1, ?tol2 nil,  ?td2 nil , ?stop 5, ?multiple nil)
delay(?wf1 v("/Cin" ?result "tran"), ?value1 2.5, ?edge1 "rising", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 v("/Cout" ?result "tran"), ?value2 2.5, ?edge2 "rising", ?nth2 1, ?tol2 nil,  ?td2 nil , ?stop 5, ?multiple nil)
delay(?wf1 v("/B" ?result "tran"), ?value1 2.5, ?edge1 "rising", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 v("/Cout" ?result "tran"), ?value2 2.5, ?edge2 "rising", ?nth2 1, ?tol2 nil,  ?td2 nil , ?stop 5, ?multiple nil)
delay(?wf1 v("/A" ?result "tran"), ?value1 2.5, ?edge1 "rising", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 v("/Cout" ?result "tran"), ?value2 2.5, ?edge2 "rising", ?nth2 1, ?tol2 nil,  ?td2 nil , ?stop 5, ?multiple nil)
delay(?wf1 v("/Cin" ?result "tran"), ?value1 2.5, ?edge1 "rising", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 v("/S" ?result "tran"), ?value2 2.5, ?edge2 "rising", ?nth2 1, ?tol2 nil,  ?td2 nil , ?stop 5, ?multiple nil)
delay(?wf1 v("/B" ?result "tran"), ?value1 2.5, ?edge1 "rising", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 v("/S" ?result "tran"), ?value2 2.5, ?edge2 "rising", ?nth2 1, ?tol2 nil,  ?td2 nil , ?stop 5, ?multiple nil)
delay(?wf1 v("/A" ?result "tran"), ?value1 2.5, ?edge1 "rising", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 v("/S" ?result "tran"), ?value2 2.5, ?edge2 "rising", ?nth2 1, ?tol2 nil,  ?td2 nil , ?stop 5, ?multiple nil)
```

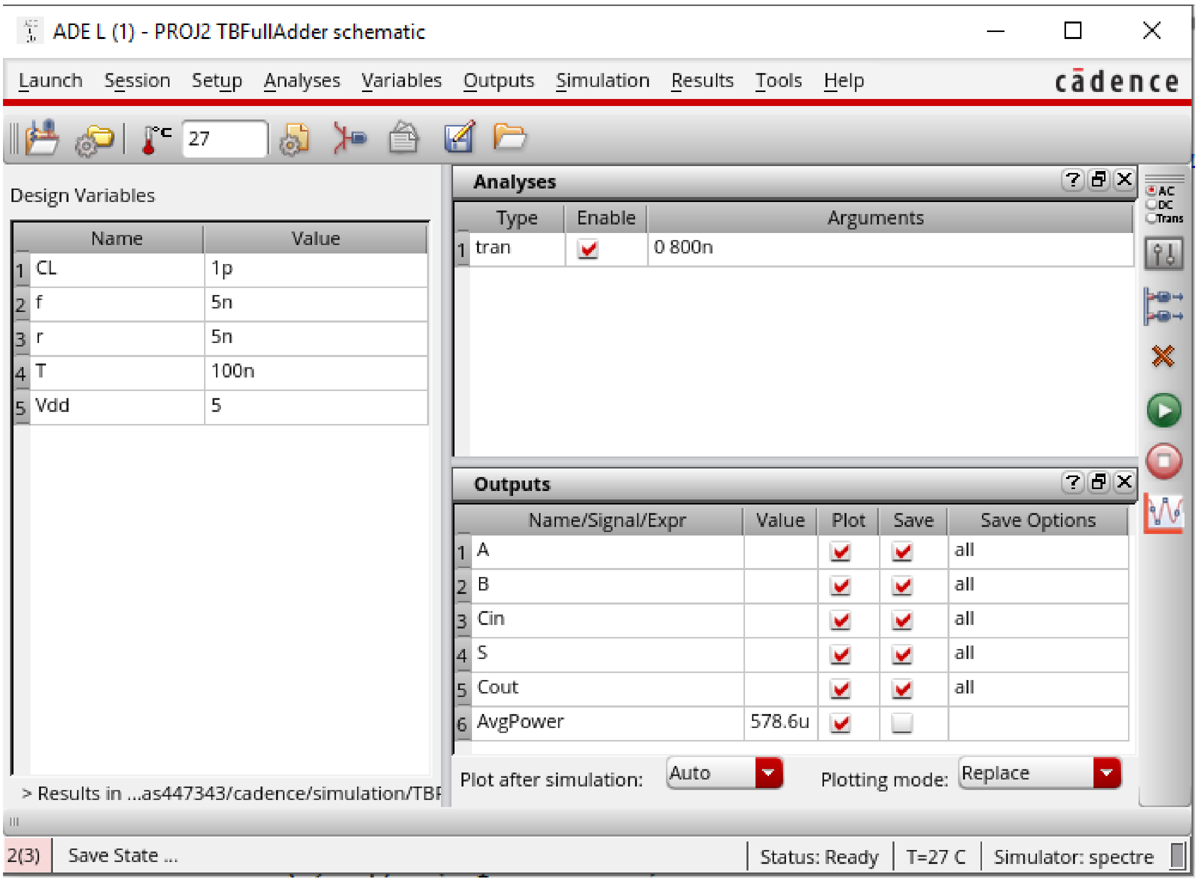
From Waveform 2 –

```
delay(?wf1 v("/B" ?result "tran"), ?value1 2.5, ?edge1 "falling", ?nth1 2, ?td1 0.0, ?tol1 nil, ?wf2 v("/Cout" ?result "tran"), ?value2 2.5, ?edge2 "falling", ?nth2 1, ?tol2 nil,  ?td2 nil , ?stop 5, ?multiple nil)
delay(?wf1 v("/B" ?result "tran"), ?value1 2.5, ?edge1 "falling", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 v("/S" ?result "tran"), ?value2 2.5, ?edge2 "falling", ?nth2 1, ?tol2 nil,  ?td2 nil , ?stop 5, ?multiple nil)
v("/Cout" ?result "tran")
v("/S" ?result "tran")
v("/Cin" ?result "tran")
v("/B" ?result "tran")
v("/A" ?result "tran")
```

<u>PART C</u>

Average Power Consumption:

Expression = **VAR("Vdd") * integ(IT("/V0/MINUS") 0 VAR("T") " ") / VAR("T")**



Analysis 1.1: Increasing T in 2x steps (keeping values of CL = 1pF, r = 5ns, f = 5ns)

T (ns)	AvgPower (µW)
100ns	578.6µW
200ns	295.7µW
400ns	147.9µW
800ns	73.98µW
1600ns	33.6µW

Analysis 1.2: Increasing T in 2x steps (keeping values of CL = 1pF, r = 5% of T, f = 5% of T)

T (ns)	r (ns)	f (ns)	AvgPower (µW)
100ns	5ns	5ns	578.6µW
200ns	10ns	10ns	328.4µW
400ns	20ns	20ns	200.2µW
800ns	40ns	40ns	134.4µW
1600ns	80ns	80ns	74.61µW

Hence it is observed that increasing the time period, i.e. **decreasing the frequency** keeping all the other parameters constant, the **average power consumption decreases**.

Analysis 2: Decreasing CL in 2x steps (keeping T = 100ns, r = 5ns, f = 5ns)

CL (pF)	AvgPower (µW)
2pF	937.3µW
1pF	578.6µW
500fF	341.6µW
250fF	216.8µW
125fF	154.3µW

Hence it is observed that **decreasing the load capacitance** keeping all the other parameters constant, the **average power consumption decreases.**

Analysis 3: Increasing r,f in 5% of T steps (keeping T = 100ns, CL = 1pF)

r (ns)	f (ns)	AvgPower (µW)
5ns	5ns	578.6µW
10ns	10ns	644.1µW
15ns	15ns	716.3µW
20ns	20ns	787.8µW
25ns	25ns	857.6µW
30ns	30ns	933.2µW

Hence it is observed that **increasing the rise/fall times** keeping all the other parameters constant, the **average power consumption increases**.