In the name of God

Digital Electronics

CA2

Mohammad Yahyapour

Part 1: Design

We define the **Inverter**, **AND**, and **OR** circuits as subcircuits, and then, using these gates, we define the **XOR** circuit.

```
*********Inverter************
.subckt     INV     Vin     Vout
M1      Vout        Vin     0       0       nch_lvt     l=Lmin      w=Wn
M2      Vout        Vin     VD      VD      pch_lvt     l=Lmin      w='2*Wn'
Vsupply     VD      0       VDD
.ends     INV
```

Figure 1: inverter code

```
*********2-Input-Nand********
.subckt     NAND2       A       B       Vout
M1      x       B       0       0       nch_lvt     l=Lmin      w='2*Wn'
M2      Vout    A       x       0       nch_lvt     l=Lmin      w='2*Wn'
M3      Vout    A       VD      VD      pch_lvt     l=Lmin      w='2*Wn'
M4      Vout    B       VD      VD      pch_lvt     l=Lmin      w='2*Wn'
Vsupply     VD      0       VDD
.ends     NAND2

*********2-Input-And*********
.subckt     AND2        A       B       Vout
X1      A       B       out0        NAND2
X2      out0    Vout    INV
.ends     AND2
```

Figure 2 And code

```
*********2-Input-NOR*********
.subckt     NOR2     A       B       Vout
M1      Vout     A       0       0       nch_lvt     l=Lmin      w=Wn
M2      Vout     B       0       0       nch_lvt     l=Lmin      w=Wn
M3      x        A       VD      VD      pch_lvt     l=Lmin      w='4*Wn'
M4      Vout     B       x       VD      pch_lvt     l=Lmin      w='4*Wn'
Vsupply     VD      0       VDD
.ends     NOR2
```

Figure 3 And code

```
**** 2-input xor gate ****
.subckt     XOR2        A       B       Vout
X1      A       Abar    INV
X2      B       Bbar    INV

X3      Abar    B       out0    AND2
X4      A       Bbar    out1    AND2
X5      out0    out1    Vout    OR2
.ends     XOR2
```

Figure 4 XOR code

```
*********GP-Logic***********
.subckt     GP      A       B       G       P
X1      A       B       G       AND2
X2      A       B       P       XOR2
.ends       GP
```

Figure 5 GP code

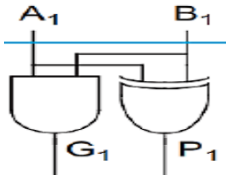

Figure 6 GP logic

```
*********AO21-Logic**********
.subckt     AO21    A       B       C       Vout
X1      B       C       out0    AND2
X2      A       out0    Vout    OR2
.ends       AO21
```

Figure 7 AI21 code

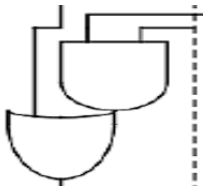

Figure8 Logic AI21

A	B	C	C +1	Condition
0	0	0	0	No Carry Generate
0	0	1	0	
0	1	0	0	
0	1	1	1	No Carry Propogate
1	0	0	0	
1	0	1	1	
1	1	0	1	Carry Generate
1	1	1	1	

Figure 9 CLA truth table

The **carry lookahead adder** is designed to speed up the transfer of carry to the next blocks. For this reason, it examines different cases for each index, such as the possible values of A_i, B_i, and C_{in}, and whether a carry bit will be generated or not. For example, when both bits $A_i$and $B_i$are equal to 1, a generate bit is produced without considering any previous carry bits. In another example, if one of these bits is 1 and the other is 0, then if $C_{in}$is 1, a carry bit is produced through **propagation**, meaning it is passed from the previous blocks. Now, the given expression will be analyzed.

$$G4{:}1 = G4 + P4.G3 + P4.G3 + P4.P3.G2 + P4P3P2G1 + P4.P3.P2.P1.G1$$

Now, based on the explanations above, all the cases in which the carry bit $G_{4:1}$is generated are listed:

1. $G4$ is generated on its own.
2. $G3$ is generated, and P_4propagates it.
3. $G2$ is generated, P_3propagates it, and finally P_4also propagates it.
4. $G1$ is generated, and P_2, P_3, and P_4propagate it.
5. $G1$ is generated, and P_2, P_1, P_3, and P_4propagate it all the way to the output.

Part 2: Vector Test

```
; specifies # of bits associated with each vector
radix 1 4 4 1 4
;*****************************************************
; defines name for each vector. For multi-bit vectors,
; innermost [] provide the bit index range, MSB:LSB
vname Cin A[4:1] B[4:1] Cout S[4:1]
;*****************************************************
; defines vector as input, output, or bi-directional
io i i i o o
; defines time unit
tunit ns
slope 0.001
PERIOD 0.8
ODELAY 0.75
;*****************************************************
vih 1 1 F F 0 0
vil 0 1 F F 0 0
voh 0.85 0 0 0 1 F
vol 0.15 0 0 0 1 F
; tabular data section
0 0 0 0 0
0 2 1 0 3
0 4 2 0 6
0 6 3 0 9
0 9 5 0 E
0 B 5 1 0
0 A 6 1 0
0 A A 1 4
0 F F 1 E
1 5 6 0 C
```

Figure 10 VEC_CLA4bit.txt

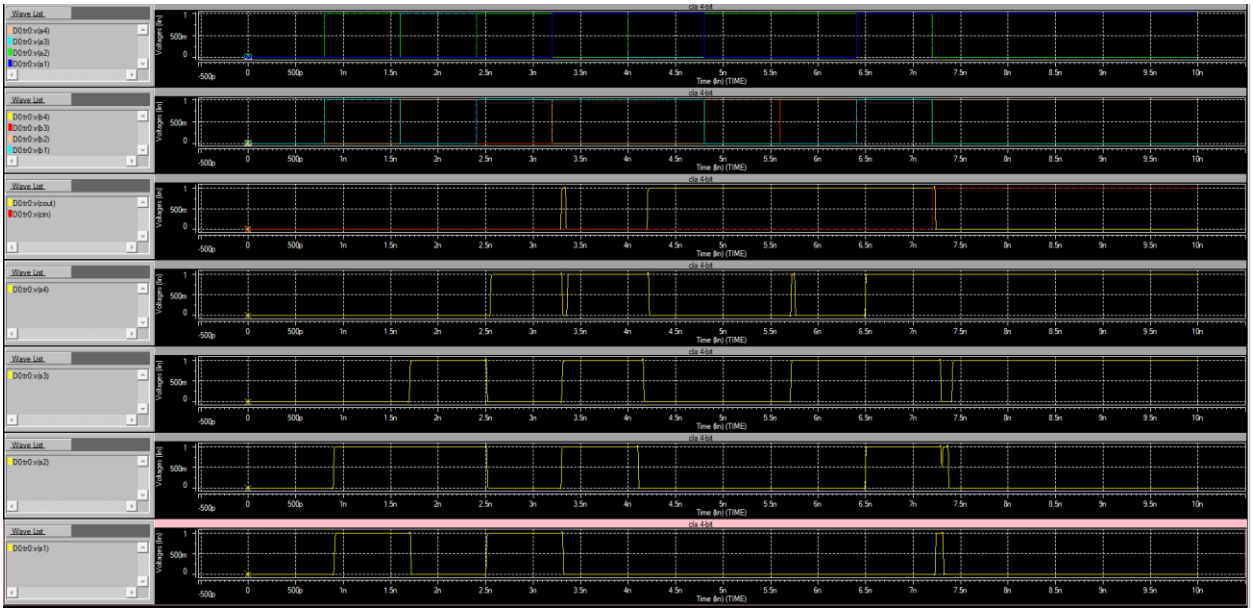

Figure 11 VEC_CLA4bit.txt input and output

The file **VEC_CLA4bit.txt** was used to provide the inputs.

Part 3: Delay and Power

For the delay section, a separate file named **part3.txt** was used to provide the inputs.

```
****** transient analysis tnom=  25.000 temp=  25.000 ******
td_tb1_cout= 205.0131p  targ=   1.0055n   trig= 800.5000p
td_tb1_s4= 220.3126p  targ=   1.0208n   trig= 800.5000p
td_tb1_s3= 166.6639p  targ= 967.1639p   trig= 800.5000p
td_tb1_s2= 112.9615p  targ= 913.4615p   trig= 800.5000p
td_tb1_s1= 115.4398p  targ= 915.9398p   trig= 800.5000p
td_tb2_s4= 117.8598p  targ=   1.7184n   trig=   1.6005n
td_tb3_s4= 207.1672p  targ=   3.4077n   trig=   3.2005n
td_tb4_cout= 152.5850p  targ=   4.9531n   trig=   4.8005n
td_tb4_s4= 167.5396p  targ=   4.9680n   trig=   4.8005n
td_tb4_s3= 113.9279p  targ=   4.9144n   trig=   4.8005n
td_tb4_s2= 113.9024p  targ=   4.9144n   trig=   4.8005n
td_tb4_s1= 110.6091p  targ=   4.9111n   trig=   4.8005n

         ***** job concluded
```

Figure 12 Delays at a temperature of 25 ℃.

```
****** transient analysis tnom=  25.000 temp=   0.000 ******
td_tb1_cout= 199.2170p  targ= 999.7170p   trig= 800.5000p
td_tb1_s4= 214.3158p  targ=   1.0148n   trig= 800.5000p
td_tb1_s3= 162.1069p  targ= 962.6069p   trig= 800.5000p
td_tb1_s2= 109.8725p  targ= 910.3725p   trig= 800.5000p
td_tb1_s1= 112.2505p  targ= 912.7505p   trig= 800.5000p
td_tb2_s4= 114.5208p  targ=   1.7150n   trig=   1.6005n
td_tb3_s4= 200.3497p  targ=   3.4008n   trig=   3.2005n
td_tb4_cout= 148.1115p  targ=   4.9486n   trig=   4.8005n
td_tb4_s4= 163.0096p  targ=   4.9635n   trig=   4.8005n
td_tb4_s3= 110.6850p  targ=   4.9112n   trig=   4.8005n
td_tb4_s2= 110.8125p  targ=   4.9113n   trig=   4.8005n
td_tb4_s1= 107.6391p  targ=   4.9081n   trig=   4.8005n

         ***** job concluded
```

Figure 13 Delays at a temperature of 0 ℃.

```
****** transient analysis tnom=  25.000 temp= 100.000 ******
td_tb1_cout= 223.2959p  targ=   1.0238n   trig= 800.5000p
td_tb1_s4= 238.9275p  targ=   1.0394n   trig= 800.5000p
td_tb1_s3= 180.8903p  targ= 981.3903p   trig= 800.5000p
td_tb1_s2= 122.6669p  targ= 923.1669p   trig= 800.5000p
td_tb1_s1= 124.5833p  targ= 925.0833p   trig= 800.5000p
td_tb2_s4= 128.3040p  targ=   1.7288n   trig=   1.6005n
td_tb3_s4= 225.6382p  targ=   3.4261n   trig=   3.2005n
td_tb4_cout= 166.4517p  targ=   4.9670n   trig=   4.8005n
td_tb4_s4= 182.1402p  targ=   4.9826n   trig=   4.8005n
td_tb4_s3= 124.1409p  targ=   4.9246n   trig=   4.8005n
td_tb4_s2= 123.4960p  targ=   4.9240n   trig=   4.8005n
td_tb4_s1= 119.5401p  targ=   4.9200n   trig=   4.8005n

         ***** job concluded
```

Figure 14 Delays at a temperature of 100 ℃.

As observed in Figures 11 to 13, the delays increase with rising temperature. This is because the circuit resistance increases, and the **Elmore delay**, which is based on RC, reflects this effect.

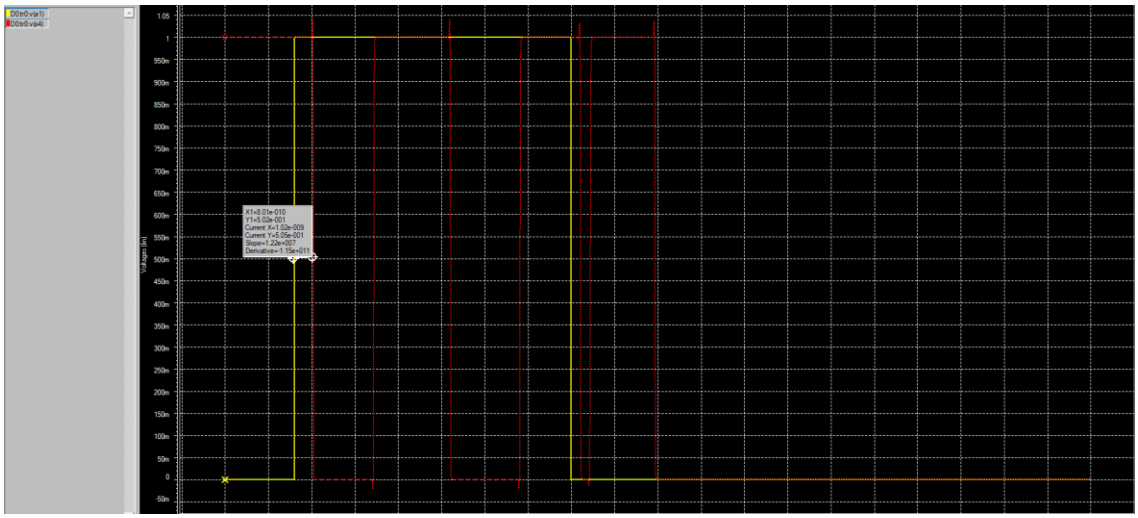

Figure 15 Delay of A1 to S3

At a temperature of 25 °C, it takes approximately **0.21 nanoseconds**, which is close to **0.205 ns**.

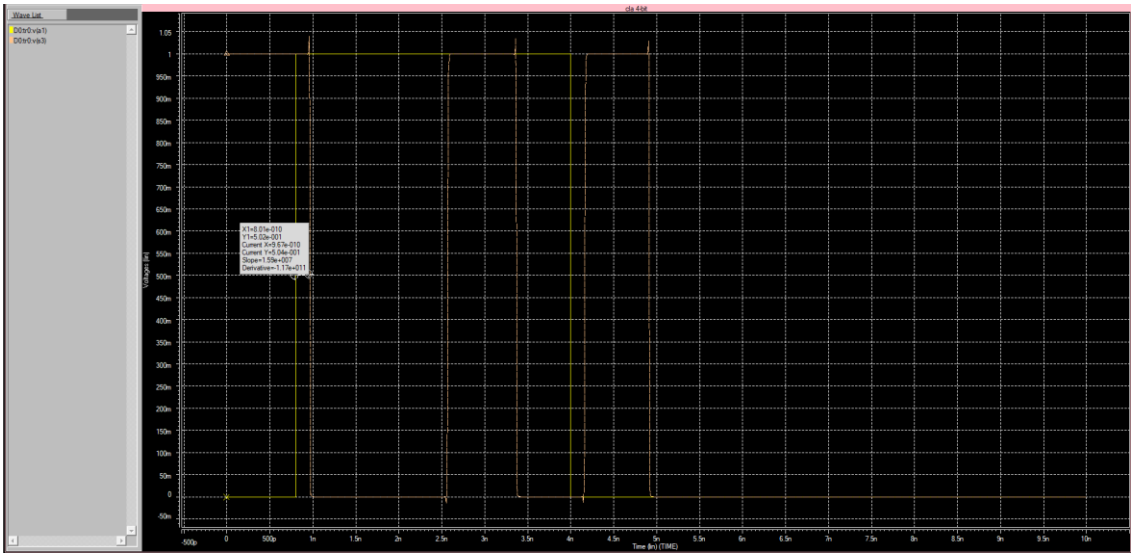

Figure 16 Delay of A1 to S3

At a temperature of 25 °C, it takes approximately **0.167 nanoseconds**, which is close to **0.166 ns**.

In this section, we calculate the **power consumption** as requested. Therefore, the file **part3_power.txt** is used to provide the inputs, and the corresponding file for this section has been placed in the **power version** folder.

Figure 16 Power at a temperature of 0 ℃.

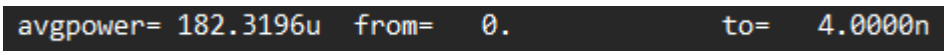

Figure 17 Power at a temperature of 25 ℃.

Figure 18 Power at a temperature of100 ℃.

As can be seen, power consumption also increases with rising temperature. This is because most of the power is consumed during switching, and as the switching time increases, power consumption also increases.