

VLSI Design Final Project: 2-Bit Unsigned Magnitude Comparator

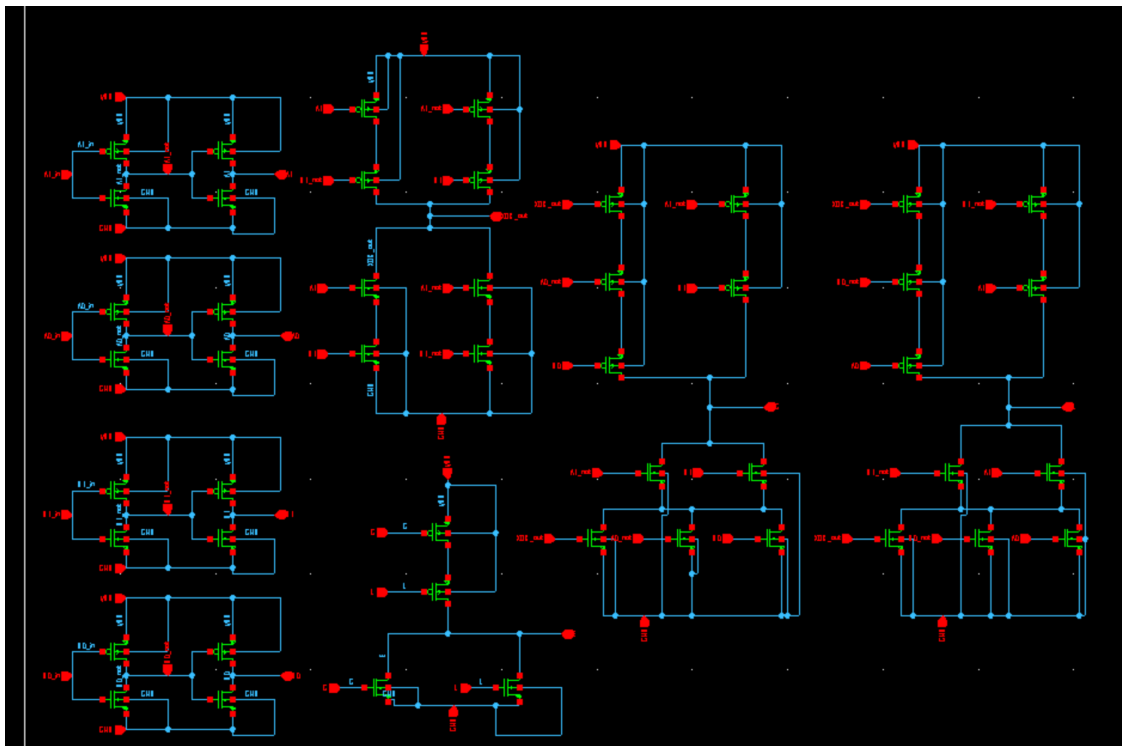
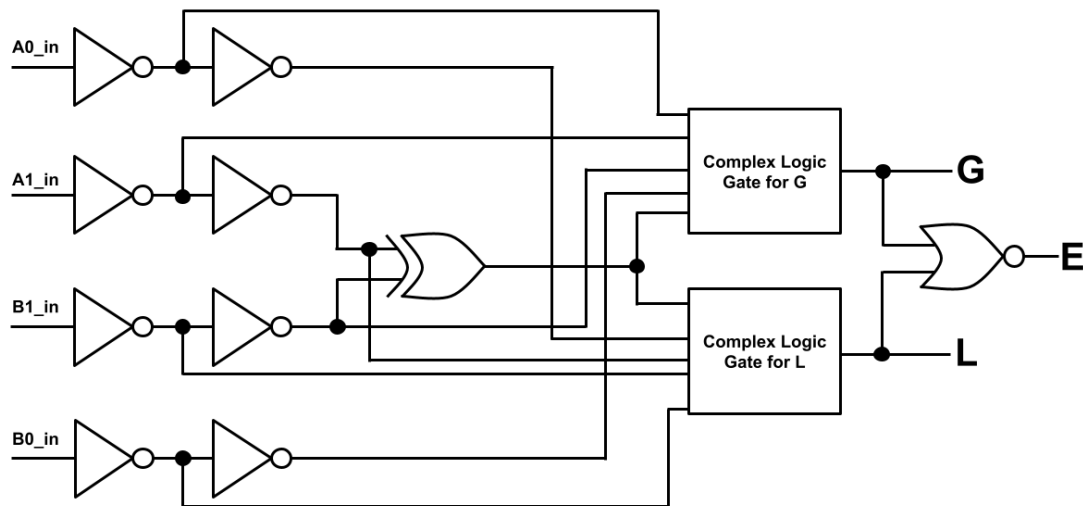
Isaac Medina

Prof. Yingjie Lao

Began: November 14, 2025

Due: December 5, 2025

[GitHub](#)



Introduction

This project uses CMOS transistor topology to implement the digital logic function of a 2-bit unsigned magnitude comparator.

$$A_1A_0 \geq B_1B_0$$

Equation 1: 2-bit Unsigned Magnitude Comparator Logic Function

Given 2-bit unsigned numbers A and B (Decimal range: $[0, 1, 2, 3]$), the CMOS transistor network computes whether A is greater than, equal to, or less than B . There are three output signals G , E , and L .

- If $A > B$, then $G = 1$; $E = 0$; $L = 0$
- If $A = B$, then $G = 0$; $E = 1$; $L = 0$
- If $A < B$, then $G = 0$; $E = 0$; $L = 1$

The goals of this project were to (1) successfully implement the 2-bit unsigned magnitude comparator using industry-standard EDA tools (***Synopsys Custom Compiler*** and ***Hspice***), (2) minimize the total number of transistors through logic restructuring and topology optimization, and (3) reduce the worst-case propagation delay by appropriately sizing each transistor and, when necessary, inserting inverter buffers.

Design Requirements:

1. CMOS only (no pass transistors)
2. Width for all transistors in the range $[300 \text{ nm}, 30,000 \text{ nm}]$
3. Input buffers consisting of two inverters in series, where the first inverter is non-skewed: i.e., $W_p = 480 \text{ nm}$, $W_n = 300 \text{ nm}$
4. A load on each primary output $C_L = 128C_{in}$, where $C_{in} \approx 2.7 \text{ fF} \rightarrow C_L \approx 350 \text{ fF}$ ($H = 128$ for each primary path)
5. Each gate needs to be non-skewed, i.e., $t_{phl} \approx t_{plh}$ for all gates.

Design Summary:

Parameter	Value
Technology	SAED 32/100 nm CMOS
Logic Function	2-bit unsigned magnitude comparator
Transistor Count	48
Critical Path	A1/B1 \rightarrow XOR \rightarrow Complex Gate \rightarrow NOR \rightarrow E
Worst-Case Delay (initial)	$\sim 20 \text{ ns}$
Worst-Case Delays (Non-buffered)	$\sim 2.1487\text{--}2.4935 \text{ ns}$
Worst-Case Delays (Buffered)	$\sim 1.7733\text{--}1.947 \text{ ns}$
Max Operating Frequency	$\sim 500 \text{ MHz}$
Output Load	$C_L \approx 350 \text{ fF}$ ($128C_{in}$)

Table 1: Design Spec Summary

Table of Contents

Click on any section to jump to it.

Title Page	1
Introduction	2
Table of Contents	3
Digital Logic Design	4
Truth Table	4
Karnaugh Maps	4
Boolean Expressions	5
Minimizing Transistors from Boolean Expressions	5
Logic Circuit Diagrams and Bubble Pushing	6
VHDL Verification of Design Logic	7
Reducing Transistors with a Custom Complex Logic Gate	7
CMOS Circuit Design	9
Gate Level Schematic	9
Synopsys Custom Compiler Schematic	9
Verification of Functionality	10
Initial Transistor Sizing	10
Input Buffers	10
XOR Gate	11
NOR Gate	11
Custom Complex Gate	12
Critical Path	12
Worst Case Transition Vectors	12
Initial Timing Parameters (no scaling)	13
Minimizing Delay with Path Effort Analysis	15
Logical Effort Values	15
Intrinsic Delay Values	15
Path Effort Analysis Derivation	16
Optimal Transistor Sizing (no buffer insertion)	16
Critical Path Delay with New Sizing (No Buffers)	17
Confirmation of Critical Path	17
Timing Parameters with Scaling	18
Non-Buffered Transistor Netlist Table	19
Delay Optimizaion with Buffer Insertion	20
Buffer Insertion Timing Contradiction	22
Reducing Delay with Manual Simulation and Hspice Optimization	22
Hspice Optimization	22
Manual Simulation and Resizing	23
Buffered Transistor Netlist Table	23
Buffered Netlist Timing Parameters.	24
Conclusions	25
References	26

Digital Logic Design

Truth Table:

A1	A0	B1	B0	G	E	L
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

Table 2: Truth table for the 2-bit unsigned comparator with outputs G ($A > B$), E ($A = B$), and L ($A < B$)

Karnough Maps:

		A_1A_0			
		00	01	11	10
B_1B_0	00	0	1	1	1
	01	0	0	1	1
	11	0	0	0	0
	10	0	0	1	0

Karnaugh Map for G

		A_1A_0			
		00	01	11	10
B_1B_0	00	0	0	0	0
	01	1	0	0	0
	11	1	1	0	1
	10	1	1	0	0

Karnaugh Map for L

Boolean Expressions:

The Karnaugh Maps above give the minterm expressions for G and L as follows:

$$G = A_1 \overline{B_1} + A_0 \overline{B_1} \overline{B_0} + A_1 A_0 \overline{B_0}$$

Boolean Expression #1 for G

$$L = B_1 \overline{A_1} + B_0 \overline{A_1} \overline{A_0} + B_1 B_0 \overline{A_0}$$

Boolean Expression #1 for L

These expressions are based solely on the K-maps, however, the expressions can also be made using an XNOR gate.

$$G = (\overline{A_1 \oplus B_1}) \cdot (A_0 \overline{B_0}) + A_1 \overline{B_1}$$

Boolean Expression #2 for G

$$L = (\overline{A_1 \oplus B_1}) \cdot (B_0 \overline{A_0}) + B_1 \overline{A_1}$$

Boolean Expression #2 for L

This says: If the most significant bits A1 and B1 are equal (XNOR operation), check if the least significant bits are different. OR, if the MSBs are different, the result can be determined that way.

Signal *E* can be expressed as:

$$E = (\overline{A_1 \oplus B_1}) \cdot (\overline{A_0 \oplus B_0})$$

Boolean Expression #1 for E

This Boolean expression checks if the MSBs AND LSBs are equal using XNOR logic.

Minimizing Transistors from Boolean Expressions:

Each expressions gives several possible ways to implement the design. To minimize transistors, consider that only two individual networks need to be made, then a NOR gate can be used to make the final signal.

Gate transistor counts:

1. Inverter $\rightarrow 2$
2. 2 Input NAND/NOR $\rightarrow 4$
3. 3 Input NAND/NOR $\rightarrow 6$
4. 2 XOR/XNOR $\rightarrow 8$

Given the different topologies from the Boolean expressions above, the combinations of possible circuits can be considered to minimize the transistor counts.

The best case combination that I came up with from the Boolean expressions was making the gate-level logic networks for G and E , then NORing for L .

$$G = (\overline{A_1 \oplus B_1}) \cdot (A_0 \overline{B_0}) + A_1 \overline{B_1}$$

$$E = (\overline{A_1 \oplus B_1}) \cdot (\overline{A_0 \oplus B_0})$$

$$L = \overline{G + E}$$

Minimal Transistor Implementation From Boolean Expressions

This gate-level implementation with no simplification is 66 transistors. 16 for the input buffer network, $8 + 8 = 16$ for the XNOR gates, $6 \times 5 = 30$ for the AND + OR gates, and 4 for the NOR gate.

Logic Circuit Diagrams and Bubble Pushing:

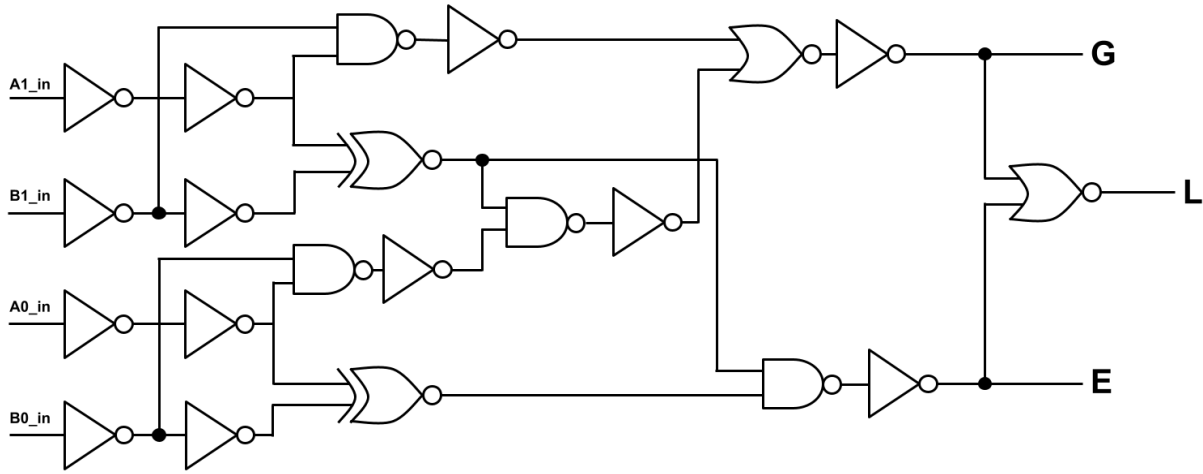


Figure 1: Initial Logic Circuit Based on Expressions above | 66 Transistors

Applying DeMorgan's Law(s) and bubble pushing the circuit can be simplified to

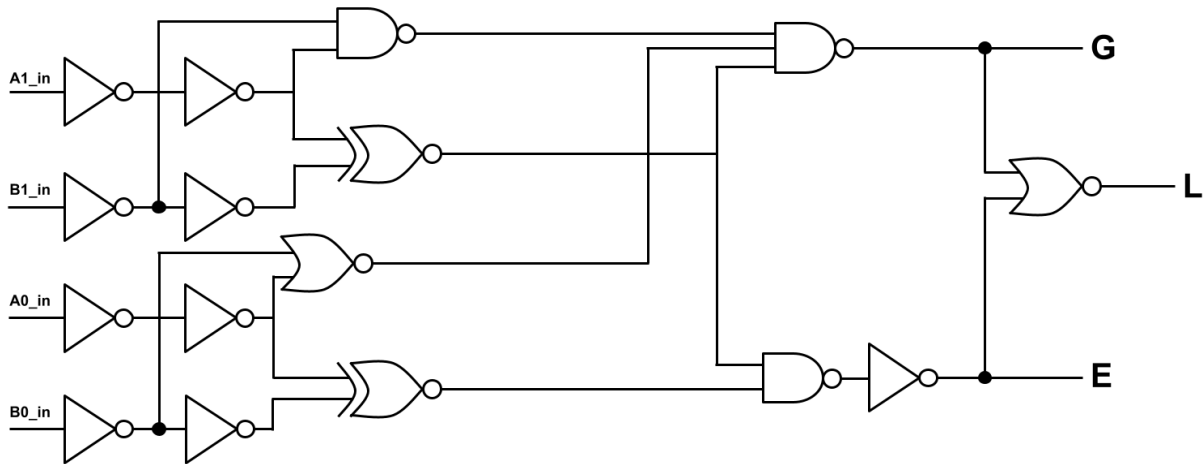


Figure 2: Simplified Logic Circuit | 56 Transistors

The circuit in Figure 2 has 16 transistors from the input buffer, 16 from the two XNOR gates, $8 + 6$ from the NAND gates, 8 from the NOR gates, and 2 from the last inverter.

$$16 + 16 + 8 + 6 + 8 + 2 = 56 \text{ transistors}$$

VHDL Verification of Design Logic:

To verify the functionality of the digital circuit, I made a VHDL program and testbench script to test the input combinations against the truth table in Table 1. The gate-level design was verified using GTKWave.

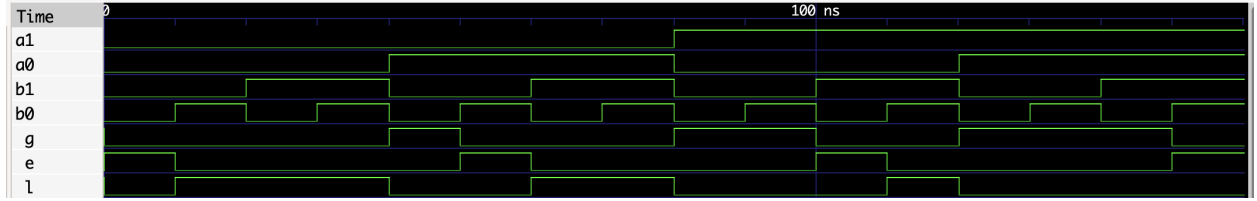


Figure 3: Verification of Gate-Level Circuit Design | GTKWave

The waveform in Figure 3 follows the truth table results in Table 1. Additionally, this gives a ground truth waveform that the CMOS circuit made in Synopsys can be tested against.

To see the VHDL program code, visit the GitHub linked in the title page.

Reducing Transistors with a Custom Complex Logic Gate

After reviewing Pages 9 and 10 of the *VLSI Combinational Circuit* slides, I realized I could reduce the transistor count by switching my design logic and making a custom complex logic gate.

Consider that in the second Boolean expressions for G and L , they both share the XNOR output signal $\overline{A_1 \oplus B_1}$. Let $X = \overline{A_1 \oplus B_1}$. Therefore,

$$G = X \cdot (A_0 \overline{B_0}) + A_1 \overline{B_1}$$

and,

$$L = X \cdot (B_0 \overline{A_0}) + B_1 \overline{A_1}$$

Then,

$$E = \overline{G + L}$$

This enables me to make a complex logic gate with the following layout for implementing this logic. Additionally, because X is generated once and shared across both G and L complex gates, the circuit avoids duplicating an 8-transistor XOR/XNOR structure.

Logic sharing across outputs is a common ASIC optimization technique, as it reduces redundant logic and minimizes both area and delay. In this design, sharing the internal XNOR signal between the G and L networks eliminated duplicate gates and helped enable a reduction from 66 to 48 transistors.

See complex logic gate on next page ↓

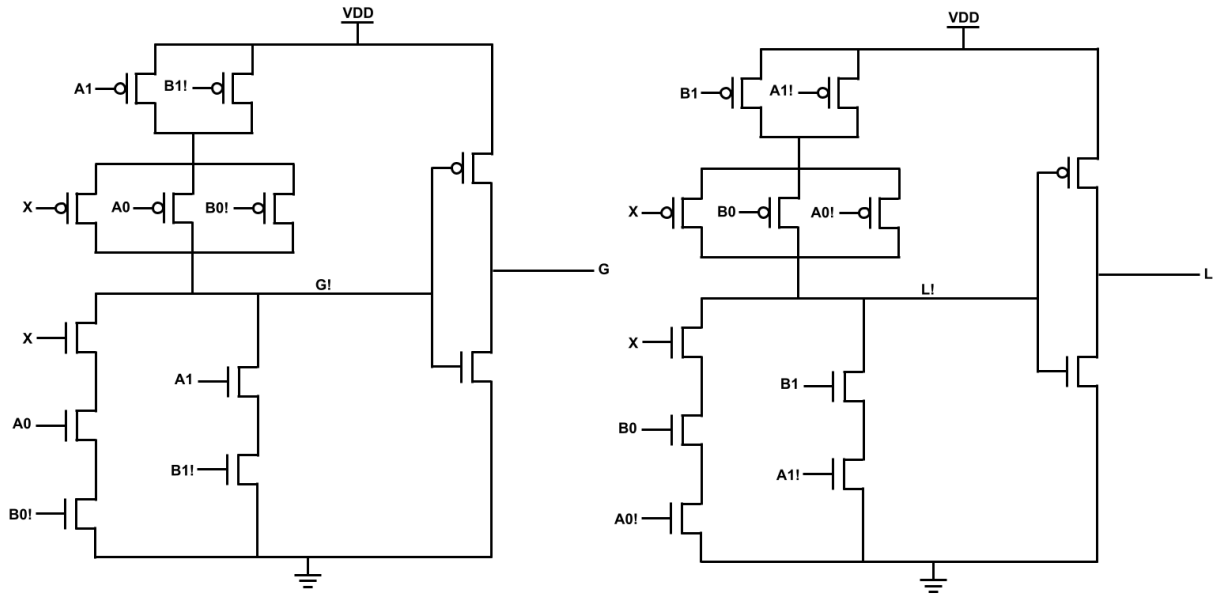


Figure 4: Initial Complex Logic Gate Topology Design for G and L

This topology for G and L reduced the transistor count to 52. However, applying DeMorgan's theorem, the PDN and PUN topologies can be swapped with the inverted signals gating the MOSFETs. This drops another 4 transistors because the inverters are not needed.

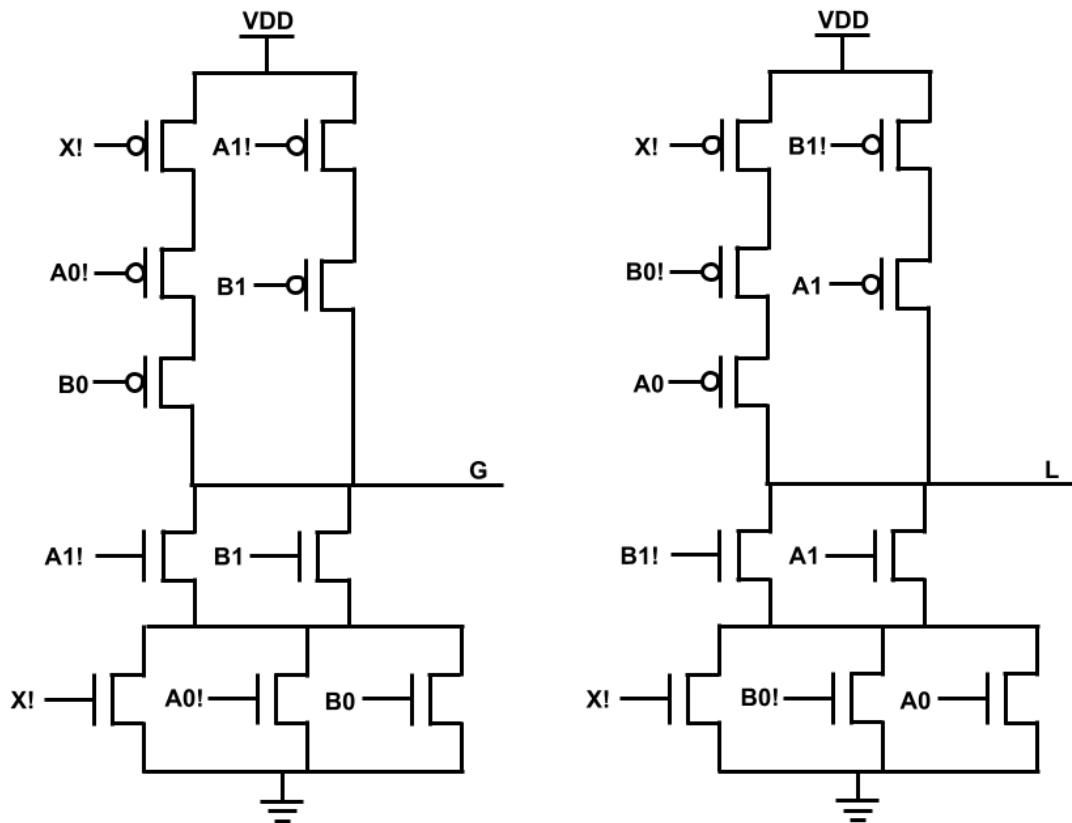


Figure 5: Improved Complex Logic Gate Design for G and L

CMOS Circuit Design

With the new design above, consider that $\overline{X} = A \oplus B$, which has the same transistor count as the XNOR gate.

The custom gate above reduces the transistor count to 48:

$$16 \text{ (inputs buffers)} + 8 \text{ (XOR gate)} + 10 \times 2 \text{ (complex gates for } G/L) + 4 \text{ (NOR for } E) = 48$$

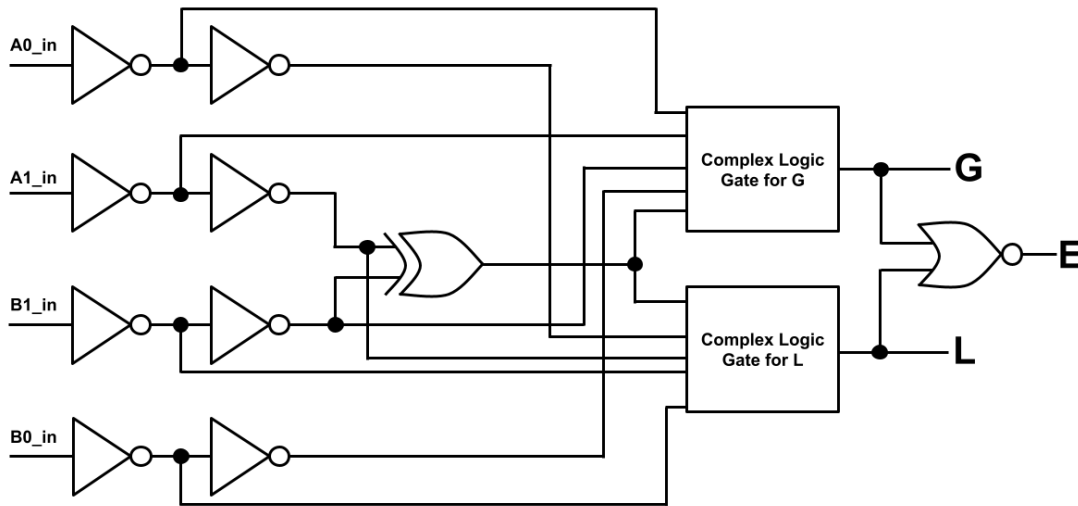


Figure 6: Final Gate-Level Design Schematic | 48 Transistors

Synopsys Custom Compiler Circuit Schematic:

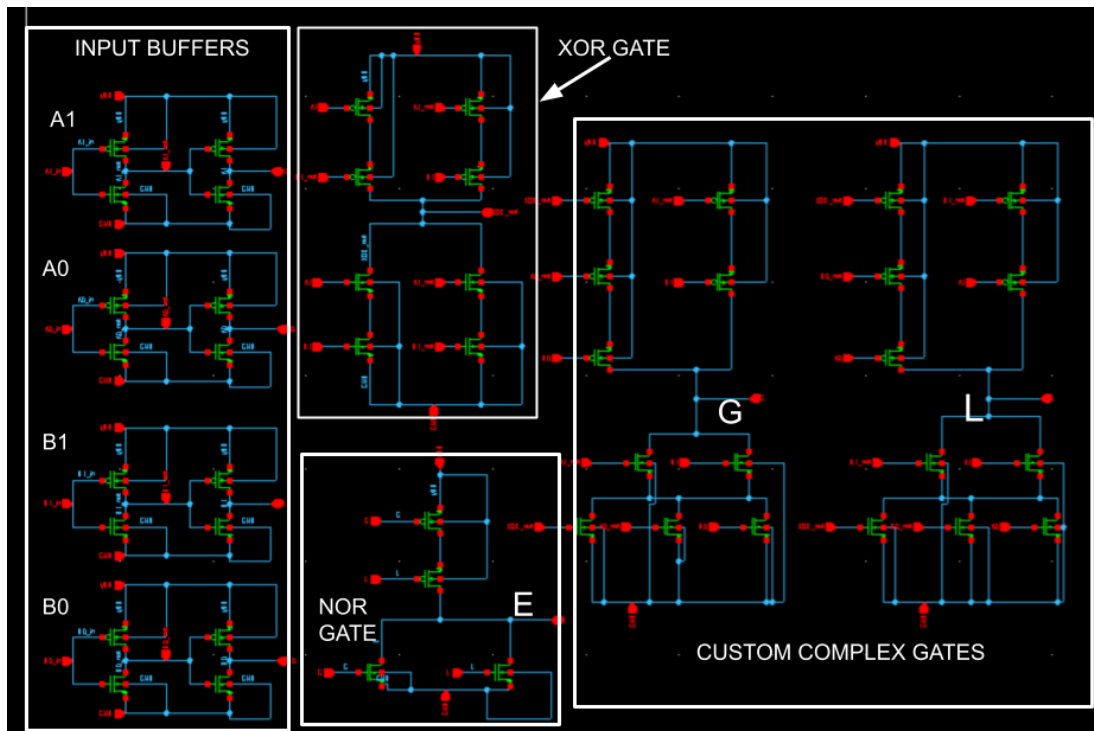


Figure 7: Annotated Synopsys Custom Compiler Circuit Schematic

Verification of Functionality:

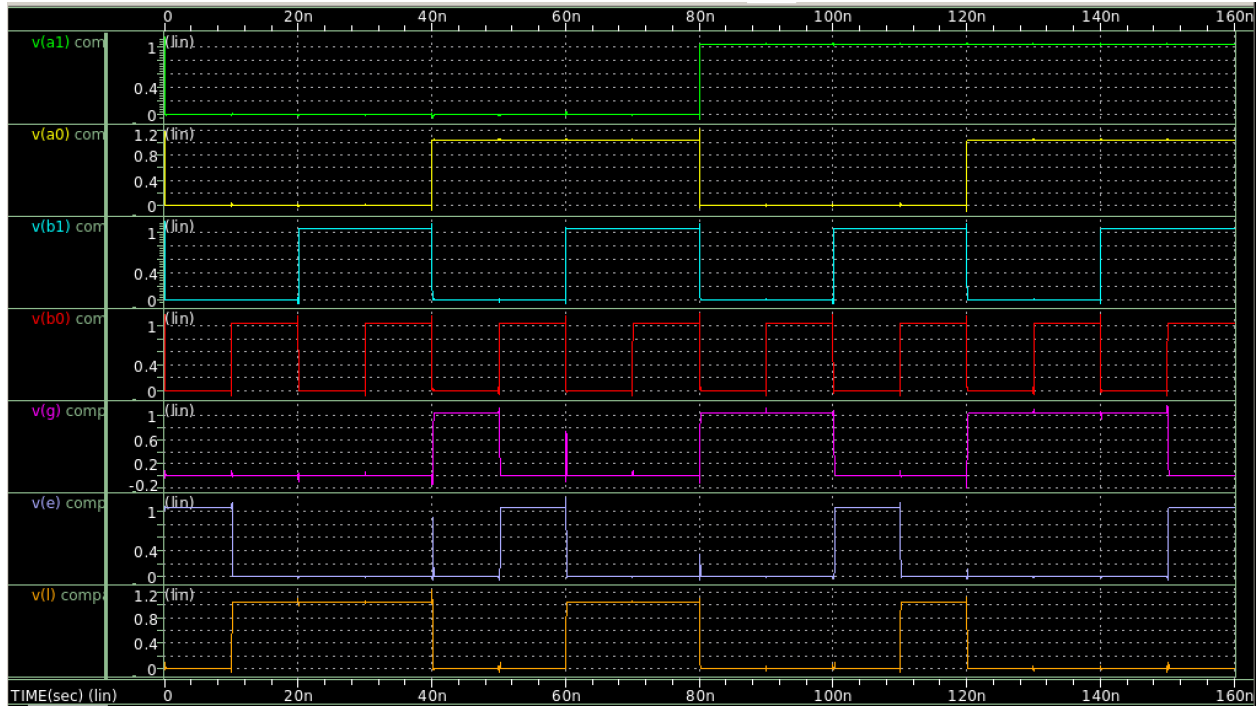


Figure 8: Hspice Output Waveform Verifying the Functionality

Figure 7 matches the waveform results from the VHDL simulation in Figure 3. This verifies the functionality of the circuit.

It is worth noting, that there is some analog noise (which is expected) on the signals when the others are switching. For example, at $t = 40$ ns, the output signal G is on a rising edge, and the output signal L is on a falling edge, this causes noise on the output signal E because it is dependent on these signals and the circuit needs to settle before E reaches a stable value.

Initial Transistor Sizing:

1. Input Buffers:

Since each input buffer is a pair of inverters, they are initially given size 1:

- $W_p = 480$ nm
- $W_n = 300$ nm

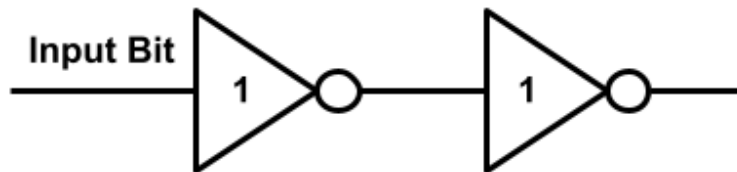


Figure 9: Input Buffer Initial Size

2. XOR Gate:

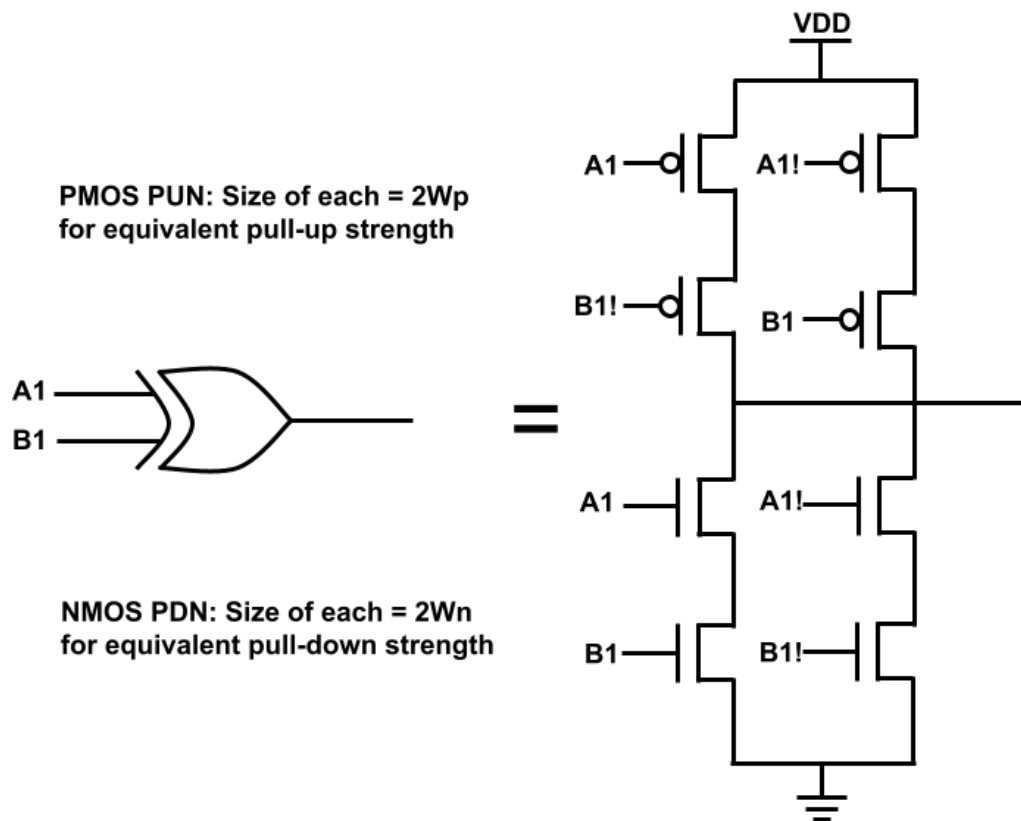


Figure 10: XOR CMOS Circuit and Initial Sizing Parameters

3. NOR Gate:

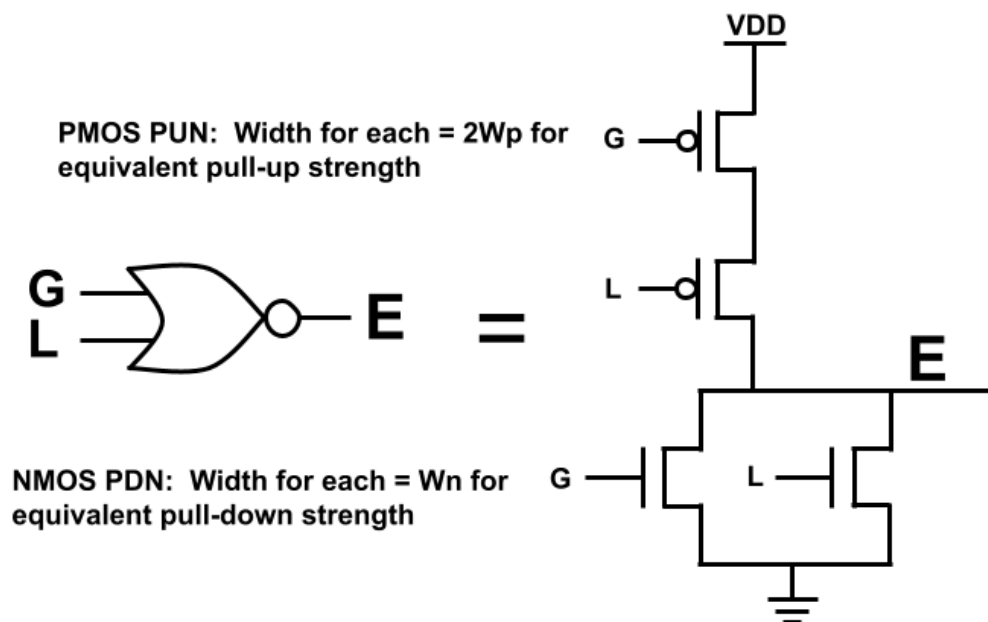


Figure 11: NOR CMOS Circuit and Initial Sizing Parameters

4. Custom Complex Gate:

My custom complex gate (shown in Figure 5) implements the logic function

$$G = X \cdot (A_0 \overline{B_0}) + A_1 \overline{B_1}$$

and,

$$L = X \cdot (B_0 \overline{A_0}) + B_1 \overline{A_1}$$

The topology consists of a PUN with 3 PMOS in series, in parallel with 2 PMOS in series. This gives:

- 3-series PMOS: Size = $3W_p$
- 2-series PMOS: Size = $2W_p$

for equivalent pull-up strength.

The PDN has 2 NMOS in parallel, in series with 3 NMOS in parallel. The path to GND is always 2 NMOS transistors, so all NMOS transistors in the PDN need to be sized at Size = $2W_n$ for equivalent pull-down strength.

Critical path

Examining the circuit diagram in Figure 6, the critical path will be:

$$A_1/B_1 \rightarrow \text{XOR} \rightarrow \text{Complex Gate} \rightarrow \text{NOR} \rightarrow E$$

There is one branch after the XOR gate, to an equivalent complex gate.

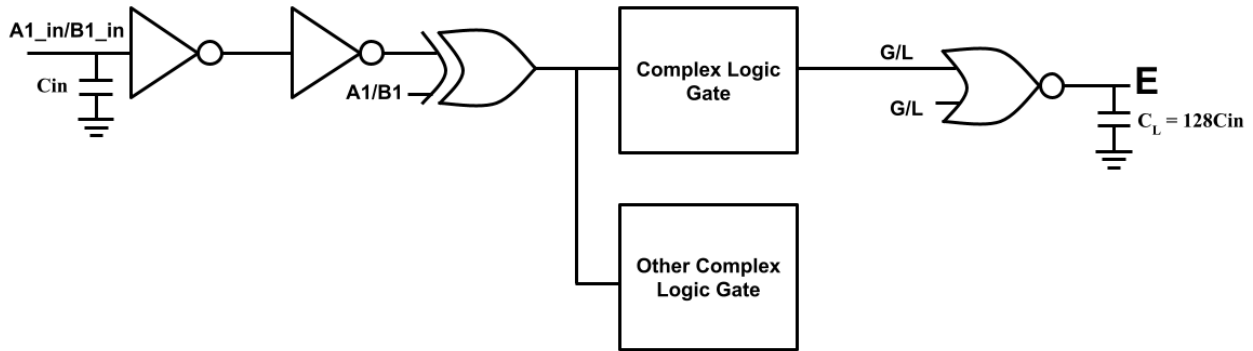


Figure 12: Critical Path Circuit Diagram

Worst Case Transition Vectors:

The worst case transition vectors occur when E changes state given that input A1_in or B1_in is changing. We consider the worst-case delay where only one of the inputs is switching. Based on the truth table in Table 1, and the output waveforms in Figures 3 and 8, E is logic 1 for these four input vectors $\{A_1 A_0 B_1 B_0\}$:

$$\{0000\}; \{0101\}; \{1010\}; \{1111\}$$

Thus, E either needs to go from high-to-low, or low-to-high, given that either $A1_in$ or $B1_in$ changed state. Thus, the worst-case transition vectors are

Case 1/2: $\{0000\} \longleftrightarrow \{1000\}$ or $\{0010\}$

Case 3/4: $\{0101\} \longleftrightarrow \{1101\}$ or $\{0111\}$

Case 5/6: $\{1010\} \longleftrightarrow \{0010\}$ or $\{1000\}$

Case 7/8: $\{1111\} \longleftrightarrow \{0111\}$ or $\{1101\}$

Initial Timing Parameters (no scaling):

To begin the timing analysis, I made sure that the critical path in Figure 11 was correct. To do this a simulation of both G and L was performed:

```
* G = 1 --> 0 --> 1 | 0100 --> 0110 --> 0100
*Valin a1_in 0 0
*Va0in a0_in 0 1
*Vb1in b1_in 0 pulse 0 vdd del trf trf pw per
*Vb0in b0_in 0 0

*.tran 1p 30u
*.measure tran tphl_G_test \
*      TRIG v(b1_in) val='0.5*vdd' rise=2 \
*      TARG v(g)      val='0.5*vdd' fall=2

*.measure tran tplh_G_test \
*      TRIG v(b1_in) val='0.5*vdd' fall=2 \
*      TARG v(g)      val='0.5*vdd' rise=2

*.measure tran tp_G_test param='(tphl_G_test + tplh_G_test)/2'
```

Hspice Script Example for Vector Timing Analysis of G/L

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_g_test= 9.3489n targ= 12.0094u trig= 12.0000u
tplh_g_test= 4.5913n targ= 17.0046u trig= 17.0000u
tp_g_test= 6.9701n

tphl_l_test= 9.3502n targ= 12.0094u trig= 12.0000u
tplh_l_test= 4.6013n targ= 17.0046u trig= 17.0000u
tp_l_test= 6.9757n
```

Figure 13: Initial Delays for G and L | No Scaling

Note that the skew is due to the input switching pattern (p.13-14 VLSI-Combinational Circuit.pdf). This data confirms the critical path.

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0000_1000= 17.2821n targ= 12.0173u trig= 12.0000u
tplh_e_1000_0000= 22.6291n targ= 17.0226u trig= 17.0000u
tp_case_1= 19.9556n
```

Figure 14: Case 1 Input Vector Initial Delay | 0000 \leftrightarrow 1000 | 19.9556ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0000_0010= 17.2626n targ= 12.0173u trig= 12.0000u
tplh_e_0010_0000= 22.8527n targ= 17.0229u trig= 17.0000u
tp_case_2= 20.0576n
```

Figure 15: Case 2 Input Vector Initial Delay | 0000 \leftrightarrow 0010 | 20.0576ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0101_1101= 17.2683n targ= 12.0173u trig= 12.0000u
tplh_e_1101_0101= 22.5631n targ= 17.0226u trig= 17.0000u
tp_case_3= 19.9157n
```

Figure 16: Case 3 Input Vector Initial Delay | 0101 \leftrightarrow 1101 | 19.9157ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0101_0111= 17.2455n targ= 12.0172u trig= 12.0000u
tplh_e_0111_0101= 22.6054n targ= 17.0226u trig= 17.0000u
tp_case_4= 19.9255n
```

Figure 17: Case 4 Input Vector Initial Delay | 0101 \leftrightarrow 0111 | 19.9255ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1010_0010= 17.2773n targ= 12.0173u trig= 12.0000u
tplh_e_0010_1010= 22.9203n targ= 17.0229u trig= 17.0000u
tp_case_5= 20.0988n
```

Figure 18: Case 5 Input Vector Initial Delay | 1010 \leftrightarrow 0010 | 20.0988ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1010_1000= 17.2876n targ= 12.0173u trig= 12.0000u
tplh_e_1000_1010= 22.7153n targ= 17.0227u trig= 17.0000u
tp_case_6= 20.0015n
```

Figure 19: Case 6 Input Vector Initial Delay | 1010 \leftrightarrow 1000 | 20.0015ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1111_0111= 17.2558n targ= 12.0173u trig= 12.0000u
tplh_e_0111_1111= 22.8742n targ= 17.0229u trig= 17.0000u
tp_case_7= 20.0650n
```

Figure 20: Case 7 Input Vector Initial Delay | 1111 \leftrightarrow 0111 | 20.0650ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1111_1101= 17.2651n targ= 12.0173u trig= 12.0000u
tplh_e_1101_1111= 22.6721n targ= 17.0227u trig= 17.0000u
tp_case_8= 19.9686n
```

Figure 21: Case 8 Input Vector Initial Delay | 1111 \leftrightarrow 1101 | 19.9686ns

Minimizing Delay with Path Effort Analysis

The critical path shown in Figure 12 gives a path delay of ~ 20 ns on all the worst-case input vectors shown on the previous page. Now, path effort analysis can be performed on the critical path to minimize the worst case delay.

- $W_p : W_n = 1.6 : 1$ for the unit sized inverter

Logical Effort Values:

The logical effort of each gate can be determined from this information and the topology of each logic gate that was made.

1. XOR Gate (CMOS topology shown in Figure 9 on p.10):

The sizing of each PMOS is 3.2, the sizing of each NMOS is 2. For a given input switching, 2 PMOS, and 2 NMOS will switch. Thus,

$$g_{XOR} = \frac{3.2 + 3.2 + 2 + 2}{2.6} = 4$$

2. NOR Gate:

$$g_{NOR} = \frac{3.2 + 1}{2.6} = \frac{4.2}{2.6}$$

3. Custom Complex Gate:

For the custom complex gate, we consider only the case where either A1 or B1 is switching. Since, A1 and B1 (and their derivatives) are in the two-series PMOS network, both their sizes are 3.2. All NMOS sizes in the PDN are 2. Thus the complex gate logical effort for the vectors to consider on the critical path is,

$$g_{\text{complex}} = \frac{3.2 + 2}{2.6} = 2$$

Intrinsic Delay Values:

For the XOR and NOR gates, the intrinsic delay values will follow the values in the table in the *CMOS-Combination Circuit.pdf* p.32. Units are measured in multiples of $p_{inv} \approx 1$

- XOR: 4
- NOR: 2

For the complex gate the intrinsic delay can be estimated by summing the widths of the complex gate seen at the output node and dividing by 2.6. The connections to the output node are one 4.8 PMOS, one 3.2 PMOS, and three 2 NMOS. Thus,

$$p_{\text{complex}} = \frac{4.8 + 3.2 + 2 + 2 + 2}{2.6} = 5.38$$

Doing an Hspice simulation of the unit inverter and the complex gate yields a similar result. See next page for results.

```

***** transient analysis tnom= 25.000 temp= 25.000 *****
tplh_complex= 67.3601p targ= 12.0001u trig= 12.0000u
tphl_complex= 60.8891p targ= 17.0001u trig= 17.0000u
tp_complex= 64.1246p
tphl_inv= 11.7663p targ= 12.0000u trig= 12.0000u
tplh_inv= 11.3178p targ= 17.0000u trig= 17.0000u
tp_inv= 11.5420p

```

Figure 22: $p_{complex}$ Confirmation | $64.1246/11.5420 \approx 5.55$

Path Effort Analysis Derivation:

The critical path (Figure 11) is an $N = 5$ stage path with one branch consisting of two inverters, one XOR gate (the output here branches to two complex gates), a complex gate, and NOR gate.

$$F = GBH$$

Path Logical Effort G :

$$G = g_{inv} \times g_{inv} \times g_{XOR} \times g_{complex} \times g_{NOR}$$

$$G = 1 \times 1 \times 4 \times 2 \times \frac{4.2}{2.6} = \frac{33.6}{2.6}$$

Branching effort B :

There is only one branch on the path. Let the capacitance of the complex gate be C . Since the branch is two identical gates, they have the same capacitance.

$$B = \frac{C + C}{C} = 2$$

We know $H = 128$ for all paths so,

$$F = \frac{33.6}{2.6} \times 2 \times 128 = \frac{8601.6}{2.6}$$

Delay is smallest when each stage bears same effort:

$$\hat{f} = F^{\frac{1}{N}} = \left(\frac{8601.6}{2.6} \right)^{1/5}$$

$$\hat{f} \approx 5.057$$

Optimal Transistor Sizing (no buffer insertion):

Let the size of the second inverter be a , the size of the XOR gate be X , the size of the complex gate be C , and size of the NOR gate be Y . Using the best stage effort, we can calculate the optimal size of each transistor without buffers for the best delay. Work backwards from the NOR gate (Y) to the initial inverters.

$$C_{in,i} = \frac{g_i C_{out}}{\hat{f}}$$

1. NOR Gate:

$$Y = \frac{128 \times \frac{4.2}{2.6}}{5.057} \approx 40.9$$

$$W_{p, \text{NOR}} = 960 \times 40.9 = 30000 \text{ nm (maxed)}$$

$$W_{n, \text{NOR}} = 300 \times 40.9 = 12270 \text{ nm}$$

2. Custom Complex Gate:

$$C = \frac{40.9 \times 2}{5.057} \approx 16.2$$

$$W_{p, \text{custom, 2 series}} = 960 \times 16.2 = 15552 \text{ nm}; \quad W_{p, \text{custom, 3 series}} = 1440 \times 16.2 = 23328 \text{ nm}$$

$$W_{n, \text{custom}} = 600 \times 16.2 = 9720 \text{ nm}$$

3. XOR Gate:

$$X = \frac{16.2 \times 4}{5.057} \approx 12.8$$

$$W_{p, \text{XOR}} = 960 \times 12.8 = 12288 \text{ nm}$$

$$W_{n, \text{XOR}} = 600 \times 12.8 = 7680 \text{ nm}$$

4. Inverter:

$$a = \frac{12.8 \times 1}{5.057} \approx 2.5$$

$$W_{p, \text{inv2}} = 480 \times 2.5 = 1200 \text{ nm}$$

$$W_{n, \text{inv2}} = 300 \times 2.5 = 750 \text{ nm}$$

Critical Path Delay with New Sizing (No Buffers)

Confirmation of Critical Path:

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_g_test= 1.9024n targ= 12.0019u trig= 12.0000u
tplh_g_test= 1.2374n targ= 17.0012u trig= 17.0000u
tp_g_test= 1.5699n
```

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_l_test= 1.9340n targ= 12.0019u trig= 12.0000u
tplh_l_test= 1.2377n targ= 17.0012u trig= 17.0000u
tp_l_test= 1.5859n
```

Figure 23: G and L Delays with New Sizing

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0000_1000= 2.2600n targ= 12.0023u trig= 12.0000u
tplh_e_1000_0000= 2.1881n targ= 17.0022u trig= 17.0000u
tp_case_1= 2.2240n
```

Figure 24: Case 1 Input Vector New Delay | 0000 ↔ 1000 | 2.224ns

This confirms that the critical path is still from $A_1/B_1 \rightarrow E$.

Timing Parameters with Scaling:

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0000_0010= 2.2097n targ= 12.0022u trig= 12.0000u
tplh_e_0010_0000= 2.1563n targ= 17.0022u trig= 17.0000u
tp_case_2= 2.1830n
```

Figure 25: Case 2 Input Vector New Delay | 0000 \leftrightarrow 0010 | 2.183ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0101_1101= 2.2501n targ= 12.0023u trig= 12.0000u
tplh_e_1101_0101= 2.1113n targ= 17.0021u trig= 17.0000u
tp_case_3= 2.1807n
```

Figure 26: Case 3 Input Vector New Delay | 0101 \leftrightarrow 1101 | 2.1807ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0101_0111= 2.1974n targ= 12.0022u trig= 12.0000u
tplh_e_0111_0101= 2.0999n targ= 17.0021u trig= 17.0000u
tp_case_4= 2.1487n
```

Figure 27: Case 4 Input Vector New Delay | 0101 \leftrightarrow 0111 | 2.1487ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1010_0010= 2.0712n targ= 12.0021u trig= 12.0000u
tplh_e_0010_1010= 2.8369n targ= 17.0028u trig= 17.0000u
tp_case_5= 2.4541n
```

Figure 28: Case 5 Input Vector New Delay | 1010 \leftrightarrow 0010 | 2.4541ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1010_1000= 1.9802n targ= 12.0020u trig= 12.0000u
tplh_e_1000_1010= 3.0067n targ= 17.0030u trig= 17.0000u
tp_case_6= 2.4935n
```

Figure 29: Case 6 Input Vector New Delay | 1010 \leftrightarrow 1000 | 2.4935ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1111_0111= 2.0562n targ= 12.0021u trig= 12.0000u
tplh_e_0111_1111= 2.7552n targ= 17.0028u trig= 17.0000u
tp_case_7= 2.4057n
```

Figure 30: Case 7 Input Vector New Delay | 1111 \leftrightarrow 0111 | 2.4057ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1111_1101= 1.9631n targ= 12.0020u trig= 12.0000u
tplh_e_1101_1111= 2.9203n targ= 17.0029u trig= 17.0000u
tp_case_8= 2.4417n
```

Figure 31: Case 8 Input Vector New Delay | 1111 \leftrightarrow 1101 | 2.4417ns

The new sizing parameters reduced the delay from ~ 20 ns $\rightarrow \sim 2.1 - 2.5$ ns. Nearly a 90% decrease in delay!

Non-Buffered Transistor Netlist Table:

Name	Type	Function	Size W
xm0	PMOS	A1 first inverter (pull-up)	480 nm
xm1	NMOS	A1 first inverter (pull-down)	300 nm
xm2	PMOS	A0 first inverter (pull-up)	480 nm
xm3	NMOS	A0 first inverter (pull-down)	300 nm
xm4	PMOS	B1 first inverter (pull-up)	480 nm
xm5	NMOS	B1 first inverter (pull-down)	300 nm
xm6	PMOS	B0 first inverter (pull-up)	480 nm
xm7	NMOS	B0 first inverter (pull-down)	300 nm
xm8	PMOS	A1 second inverter (pull-up)	1200 nm
xm9	NMOS	A1 second inverter (pull-down)	750 nm
xm10	PMOS	A0 second inverter (pull-up)	1200 nm
xm11	NMOS	A0 second inverter (pull-down)	750 nm
xm12	PMOS	B1 second inverter (pull-up)	1200 nm
xm13	NMOS	B1 second inverter (pull-down)	750 nm
xm14	PMOS	B0 second inverter (pull-up)	1200 nm
xm15	NMOS	B0 second inverter (pull-down)	750 nm
xm16, 17, 23, 24	PMOS	XOR Gate (pull-up)	12288 nm
xm18, 19, 25, 26	NMOS	XOR Gate (pull-down)	7680 nm
xm28, 29, 30	PMOS	Complex Gate G (pull-up)	23328 nm
xm34, 35	PMOS	Complex Gate G (pull-up)	15552 nm
xm31, 32, 33, 36, 37	NMOS	Complex Gate G (pull-down)	9720 nm
xm38, 39, 40	PMOS	Complex Gate L (pull-up)	23328 nm
xm44, 45	PMOS	Complex Gate L (pull-up)	15552 nm
xm41, 42, 43, 46, 47	NMOS	Complex Gate L (pull-down)	9720 nm
xm21, 22	PMOS	NOR Gate (pull-up)	30000 nm
xm20, 27	NMOS	NOR Gate (pull-down)	12270 nm

Table 3: Non-Buffered Transistor Netlist Summary for 2-bit CMOS Magnitude Comparator

Delay Optimization with Buffer Insertion (optional)

To optimize the delay even further, buffer insertion can be used to find the optimal number of stages with additional inverter buffers. We insert inverter because this does not add much intrinsic delay, and it does not change the path effort of the critical path.

Since adding inverters does not change the path logical effort G , and H and B are still the same, the path effort does not change:

$$F = \frac{8601.6}{2.6}$$

Now, using the optimal delay formula, we can iterate to find the minimum delay and optimum number of buffers to insert.

$$D_{\text{opt}} = NF^{1/N} + P$$

The sum of intrinsic delays through the critical path is

$$P = p_{\text{inv}, 1} + p_{\text{inv}, 2} + p_{\text{XOR}} + p_{\text{complex}} + p_{\text{NOR}} + n_i$$

Where n_i is the additional number of inverters that are added to the path. Figure 21 shows that the intrinsic delay of the complex gate is $\sim 5.55t_{p0}$. Therefore,

$$P = 1 + 1 + 4 + 5.55 + 2 + n_i = 13.55 + n_i$$

Thus, the function to optimize becomes:

$$D_{\text{opt}} = NF^{1/N} + 13.55 + n_i \quad N \geq 5, \quad n_i = N - 5$$

D_{opt} iterations for $N \geq 5$:

$$5. D_{\text{opt}, 5} = 38.83$$

$$6. D_{\text{opt}, 6} = 37.71$$

$$7. D_{\text{opt}, 7} = 37.8$$

$$8. D_{\text{opt}, 8} = 38.58$$

The optimal is around 6 or 7 stages. However, since we are starting with 5 stages, the total number of stages must be odd to maintain signal polarity. So, we cannot choose 6 stages. Therefore, 7 stages, or 2 additional inverters, should be added to decrease the delay.

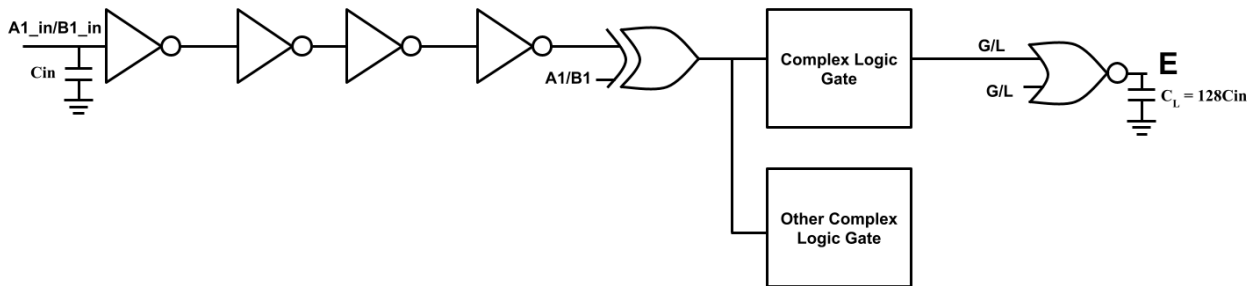


Figure 32: New Critical Path with Buffer Insertion

Let the size of the first inverter be 1, the size of the next three inverters be a, b, c , the size of the XOR be X , the size of the complex gate be C , the size of the NOR gate be Y . Now, work backwards to find the size.

$$\hat{f} = F^{\frac{1}{N}} = \left(\frac{8601.6}{2.6} \right)^{1/7} \approx 3.183$$

1. NOR Gate:

$$Y = \frac{128 \times \frac{4.2}{2.6}}{3.183} \approx 65$$

$$W_{p, \text{NOR, buff}} = 65 \times 960 \rightarrow 30000 \text{ nm (maxed)}$$

$$W_{n, \text{NOR, buff}} = 65 \times 300 = 19500 \text{ nm}$$

2. Complex Gate:

$$C = \frac{65 \times 2}{3.183} = 40.8$$

$$W_{p, \text{complex, 3 series, buff}} = 40.8 \times \frac{2.6}{6.8} \times 1440 = 22,464 \text{ nm}$$

$$W_{p, \text{complex, 2 series, buff}} = 40.8 \times \frac{2.6}{5.2} \times 960 = 19584 \text{ nm}$$

$$W_{n, \text{complex, buff}} = 40.8 \times \frac{2.6}{5.2} \times 600 = 12240 \text{ nm}$$

3. XOR Gate:

$$Y = \frac{40.8 \times 2}{3.183} = 25.65$$

$$W_{p, \text{XOR, buff}} = 25.3 \times \frac{2.6}{5.2} \times 960 = 12312 \text{ nm}$$

$$W_{n, \text{XOR, buff}} = 600 \times 25.65 \times \frac{2.6}{5.2} = 7695 \text{ nm}$$

4. Inverter c :

$$c = \frac{25.65 \times 1}{3.183} = 8$$

$$W_{p, \text{inv}c} = 480 \times 8 = 3868 \text{ nm}$$

$$W_{n, \text{inv}c} = 300 \times 8 = 2400 \text{ nm}$$

5. Inverter b

$$X = \frac{8 \times 1}{3.183} = 2.5$$

$$W_{p, \text{inv}b} = 480 \times 2.5 = 2400 \text{ nm}$$

$$W_{n, \text{inv}b} = 300 \times 2.5 = 1500 \text{ nm}$$

6. Inverter a :

$$a = \frac{5 \times 1}{3.183} = 1.6$$

$$W_{p, \text{inv}, a} = 480 \times 1.6 = 768 \text{ nm}$$

$$W_{n, \text{inv}, a} = 300 \times 1.6 = 480 \text{ nm}$$

Buffer Insertion Timing Contradiction:

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0000_0010= 4.3602n targ= 12.0044u trig= 12.0000u
tplh_e_0010_0000= 7.8275n targ= 17.0078u trig= 17.0000u
tp_case_2= 6.0938n
```

Figure 33: Delay Contradiction of Buffer Insertion

While the math above suggested that these were the optimal sizes and number of stages, the delay of the circuit actually went up about 4 ns, suggesting that changes to the path topology and sizing parameters needed to be made. Additionally, this circuit is not balanced.

The contradiction arises because logical effort assumes ideal parasitics, while the large complex, XOR, and NOR gates introduce diffusion loading and series stacking that LE may not properly model.

Reducing Delay with Manual Simulation and Hspice Optimization

Using the new critical path topology in Figure 31, the Hspice optimization tool was used to give a baseline value for the NOR gate widths. Then, manual simulation and Hspice optimization were done to reduce the delay further.

Hspice Optimization:

```
.param wp = 30.0000u $ 100.0000 33.3333 |
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1111_1101= 3.3051n targ= 12.0033u trig= 12.0000u
tplh_e_1101_1111= 5.6291n targ= 17.0056u trig= 17.0000u
tp_case_8= 4.4671n
```

Figure 34: NOR Gate Wp Size Optimization using Hspice Optimization Tool

The result of the Hspice optimization suggested that the size of NOR gate needs to be very large to compensate for the capacitance and reduce delay.

Furthermore, the Hspice optimization tool and simulation suggested that the XOR gate was oversized based on the logical effort math. Generally, I found the final gate sizes through manual simulation and tinkering with the size until it was faster and more balanced. The path effort values were the starting point. I started with the NOR gate, and then moved backward to optimize the circuit.

Most of the gates were sized well, but slight tweaks to the size enabled a faster circuit. For example, slightly increasing the sizes of the complex gate PMOS widths from 22464nm to 25000nm and 19584nm to 20000nm, while also increasing the NMOS width to 15000nm made the circuit faster and more balanced. Overall, the inverters were generally undersized for optimal timing and balance based on logical effort.

Additionally, when I changed the PMOS:NMOS ratios closer to 2:1 for the buffer-inserted circuit, the circuit performed faster and more balanced.

Manual Simulation and Resizing:

After much resizing (both manually and with the optimization tool), the following sizing parameters in the table below produced the minimum delays on the critical path in Figure 32 above:

Name	Type	Function	Size W
xm0	PMOS	A1 first inverter (pull-up)	480 nm
xm1	NMOS	A1 first inverter (pull-down)	300 nm
xm2	PMOS	A0 first inverter (pull-up)	480 nm
xm3	NMOS	A0 first inverter (pull-down)	300 nm
xm4	PMOS	B1 first inverter (pull-up)	480 nm
xm5	NMOS	B1 first inverter (pull-down)	300 nm
xm6	PMOS	B0 first inverter (pull-up)	480 nm
xm7	NMOS	B0 first inverter (pull-down)	300 nm
xm8	PMOS	A1 second inverter (pull-up)	2000 nm
xm9	NMOS	A1 second inverter (pull-down)	1000 nm
xm10	PMOS	A0 second inverter (pull-up)	5000 nm
xm11	NMOS	A0 second inverter (pull-down)	2500 nm
xm12	PMOS	B1 second inverter (pull-up)	1000 nm
xm13	NMOS	B1 second inverter (pull-down)	500 nm
xm14	PMOS	B0 second inverter (pull-up)	5000 nm
xm15	NMOS	B0 second inverter (pull-down)	2500 nm
xm16, 17, 23, 24	PMOS	XOR Gate (pull-up)	8000 nm
xm18, 19, 25, 26	NMOS	XOR Gate (pull-down)	4000 nm
xm28, 29, 30	PMOS	Complex Gate G (pull-up)	25000 nm
xm34, 35	PMOS	Complex Gate G (pull-up)	20000 nm
xm31, 32, 33, 36, 37	NMOS	Complex Gate G (pull-down)	15000 nm
xm38, 39, 40	PMOS	Complex Gate L (pull-up)	25000 nm
xm44, 45	PMOS	Complex Gate L (pull-up)	20000 nm
xm41, 42, 43, 46, 47	NMOS	Complex Gate L (pull-down)	15000 nm
xm21, 22	PMOS	NOR Gate (pull-up)	30000 nm
xm20, 27	NMOS	NOR Gate (pull-down)	20000 nm
xm48	PMOS	1st Inverter Buffer A1 (pull-up)	2000 nm
xm49	NMOS	1st Inverter Buffer A1 (pull-down)	1000 nm
xm50	PMOS	1st Inverter Buffer B1 (pull-up)	2000 nm
xm51	NMOS	1st Inverter Buffer B1 (pull-down)	1000 nm
xm52	PMOS	2nd Inverter Buffer A1 (pull-up)	4000 nm
xm53	NMOS	2nd Inverter Buffer A1 (pull-down)	2000 nm
xm54	PMOS	2nd Inverter Buffer B1 (pull-up)	4000 nm
xm55	NMOS	2nd Inverter Buffer B1 (pull-down)	2000 nm

Table 4: Buffered Transistor Netlist Summary for 2-bit CMOS Magnitude Comparator

Buffered Netlist Timing Parameters:

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0000_1000= 2.0397n targ= 12.0020u trig= 12.0000u
tplh_e_1000_0000= 1.6796n targ= 17.0017u trig= 17.0000u
tp_case_1= 1.8596n
```

Figure 35: Case 1 Input Vector Final Delay | 0000 \leftrightarrow 1000 | 1.8596ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0000_0010= 1.9546n targ= 12.0020u trig= 12.0000u
tplh_e_0010_0000= 1.6531n targ= 17.0017u trig= 17.0000u
tp_case_2= 1.8039n
```

Figure 36: Case 2 Input Vector Final Delay | 0000 \leftrightarrow 0010 | 1.8039ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0101_1101= 2.0329n targ= 12.0020u trig= 12.0000u
tplh_e_1101_0101= 1.6372n targ= 17.0016u trig= 17.0000u
tp_case_3= 1.8350n
```

Figure 37: Case 3 Input Vector Final Delay | 0101 \leftrightarrow 1101 | 1.8350ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_0101_0111= 1.9355n targ= 12.0019u trig= 12.0000u
tplh_e_0111_0101= 1.6111n targ= 17.0016u trig= 17.0000u
tp_case_4= 1.7733n
```

Figure 38: Case 4 Input Vector Final Delay | 0101 \leftrightarrow 0111 | 1.7733ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1010_0010= 1.6281n targ= 12.0016u trig= 12.0000u
tplh_e_0010_1010= 2.1725n targ= 17.0022u trig= 17.0000u
tp_case_5= 1.9003n
```

Figure 39: Case 5 Input Vector Final Delay | 1010 \leftrightarrow 0010 | 1.9003ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1010_1000= 1.8533n targ= 12.0019u trig= 12.0000u
tplh_e_1000_1010= 2.0408n targ= 17.0020u trig= 17.0000u
tp_case_6= 1.9470n
```

Figure 40: Case 6 Input Vector Final Delay | 1010 \leftrightarrow 1000 | 1.9470ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1111_0111= 1.6117n targ= 12.0016u trig= 12.0000u
tplh_e_0111_1111= 2.1324n targ= 17.0021u trig= 17.0000u
tp_case_7= 1.8721n
```

Figure 41: Case 7 Input Vector Final Delay | 1111 \leftrightarrow 0111 | 1.8721ns

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
tphl_e_1111_1101= 1.8394n targ= 12.0018u trig= 12.0000u
tplh_e_1101_1111= 2.0279n targ= 17.0020u trig= 17.0000u
tp_case_8= 1.9336n
```

Figure 42: Case 8 Input Vector Final Delay | 1111 \leftrightarrow 1101 | 1.9336ns

Conclusions:

This project implemented a CMOS transistor network that performs a 2-bit unsigned magnitude comparator

$$A_1A_0 \geq B_1B_0$$

and optimized it from an initial 66-transistor design down to **48 transistors**. This reduction was achieved through bubble pushing, logic restructuring, and a custom complex logic gate (Figure 5, p.8). By sharing internal circuitry between multiple output signals, minimizing redundant logic, and adjusting design topology, the final transistor netlist met design requirements. This demonstrates the importance of Boolean simplification and logic restructuring in reducing die area, improving timing, and lowering cost in CMOS circuits.

The CMOS comparator was made in *Synopsys Custom Compiler* (Figure 7, p.9), and verified against the ground-truth gate level *VHDL* simulation and *GTKWave* visualization (Figure 3, p.7). Dynamic behavior was confirmed with *Hspice* and *Custom Waveview* (Figure 8, p.10), ensuring correct operation across all 16 input combinations in Table 2 (p.4).

Using path effort analysis, the critical path (Figure 12, p.12) was analyzed to minimize the worst-case delay. Pre-scaling Hspice simulation gave initial delays of the worst-case input vectors (p.13) around 20 ns. After applying the analytically derived sizing factors on p.17, the non-buffered netlist (Table 3, p.19) delays fell to $\sim 2.1 - 2.5$ ns—a nearly **$\sim 90\%$** average decrease in delay on the critical path. This improvement enables the circuit to operate at just under 500 MHz, highlighting the importance of equal stage effort and methodical sizing in CMOS circuits.

To further reduce delay, buffer insertion was explored. The initial Hspice simulation showed that the analytical results contradicted the expected results of a delay decrease (~ 4 ns) for the buffer-inserted circuit (Figure 33, p.22). Using the Hspice optimization tool and rigorous manual simulation and resizing, the finalized sizing parameters in Table 4 (p.23) produced the minimal delays with buffer insertion. The buffered circuit resulted in the worst-case transition vectors being < 2 ns, with an absolute worst-case delay of **1.947 ns**, confirming that the design can be run at **500 MHz**.

While these sizing parameters deviate from the ideal logical-effort predictions, they not only reduced the delay, but also balanced the circuit more based on the results in Figures 35-42. The final size parameters moved closer toward the typical $W_p:W_n$ ratio (2:1), illustrating a key limitation of logical effort in circuits with high-fan-in gates and large parasitic components: LE is an excellent first-order sizing tool, but final optimization in short-channel technologies requires parasitic-aware SPICE tuning that may diverge from the mathematical results derived on paper. By taking the initial sizing results from the path effort analysis as a starting point, the Hspice tool can be used to fine-tune the circuit to achieve optimal results based on simulation in the SAED-32/100 nm process.

This project provided experience into physical design flows for digital IC design using the SAED-32/100 nm CMOS process. By using industry standard EDA tools such as *Synopsys Custom Compiler*, *Hspice*, *Custom Waveview*, and *VHDL*, I gained valuable insight into physical design tradeoffs between theoretical sizing, actual results in SPICE-level validation, and how to optimize digital circuits to meet design requirements.

References

1. Professor Yingjie Lao's Class Slides: *2-VLSI-Transistor.pdf*, *3-VLSI-CMOS.pdf*, *4-VLSI-Combinational Circuit.pdf*, <https://canvas.tufts.edu/courses/67076/files/folder/Slides>
2. N. Weste and D. Harris, "CMOS VLSI Design", Addison-Wesley/Pearson, 4th edition, 2011. ISBN: 0321547748.
3. J. M. Rabaey, "Digital Integrated Circuits: A Design Perspective", Prentice-Hall, 2nd edition, 2003, ISBN: 0130909963.
4. S. Kang and Y. Leblebici, "CMOS Digital Integrated Circuits", McGraw Hill, 4th edition, 2014. ISBN: 0073380628.
5. Various *Electrical Engineering Stack Exchange* Forums: <https://electronics.stackexchange.com/>
6. Various *r/ElectricalEngineering Reddit* Forums: <https://www.reddit.com/r/ElectricalEngineering/>
7. I made all Logic and CMOS diagrams in Google Drawing: <https://drive.google.com/drive/u/0/folders/1rwMiuuLJ3LnRrTESn1KFg6PZsa7fJVQT>