EE 457: Digital Integrated Circuits

Project #1 Report Cover Sheet

Due: 9/30/2023 by 6PM

PROJECT TITLE: 2-Input CMOS Exclusive OR Gate

Student Name: Dylan Kirdahy

Your report should follow the following items in sequence. *Do not change the sequence.*	GRADE
Section 1: Executive Summary	/5
Section 2: Background and Approach (Include a truth table)	/5
Section 3: Electric Schematic	/15
Section 4: LTSPICE for Electric schematic	/10
Section 5: IRSIM for Electric schematic	/10
Section 6: Electric Layout	/25
Section 7: LTSPICE for Electric layout (compare with schematic)	/15
Section 8: IRSIM for Electric layout	/10
Section 9: Conclusions and References	/5
TOTAL	**/100**

Contents

1 Executive Summary **1**

2 Background **1**

3 Approach **1**

1 Executive Summary

The goal of this project is to design an XOR gate CMOS integrated circuit using the design software Electric. The process of designing the chip involves starting with the boolean logic that makes up an XOR gate, turning that logic into a functioning schematic using CMOS logic, and then turning that schematic into a finished layout. Then, both the schematic and the layout are simulated in both LTSpice and IRSIM to confirm that they function properly, and finally measurements can be taken in LTSpice to further validate the design. Ultimately, this project is an introduction to the process of turning requirements into a finished design and then validating the design using the tools we have learned from creating the inverter.

2 Background

Designing an IC that performs the function of an XOR gate begins with an understanding of what an XOR gate does. The operation, denoted as $F = A \oplus B$ is a boolean logic function that takes inputs A and B and outputs a 0 if they are the same and 1 if they are dissimilar. This behavior can be seen in Table 1. The symbol for the XOR gate is shown in Figure 1. We can break down this logic into the following boolean expressions: $F = A \oplus B = A\overline{B} + \overline{A}B = (A + B)(\overline{A} + \overline{B})$

Input A	**Input B**	**Output F**
0	0	0
0	1	1
1	0	1
1	1	0

Table 1: Truth Table for the XOR gate.

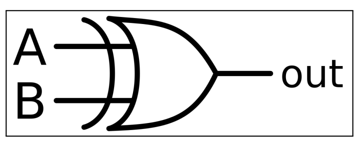

Figure 1: The symbol for the XOR gate.

3 Approach

I chose to start designing the CMOS logic using the function $F = A\overline{B} + \overline{A}B$. The first thing to do is to invert the function, since the output of a CMOS circuit is inverted. Therefore we must invert F and then implement that function in the pull down network. Using De Morgan's Law, we get: $\overline{F} = \overline{A\overline{B} + \overline{A}B} = \overline{A\overline{B}} \cdot \overline{\overline{A}B} = (\overline{A} + B)(A + \overline{B})$ Therefore $(\overline{A} + B)(A + \overline{B})$ will be implemented on the NMOS side. We do this by putting the transistors in parallel for OR and in series for AND. For the pull up network, we put the PMOS transistors in series for OR and in parallel for AND. I sketched out the resulting circuit, as shown in Figure 2.

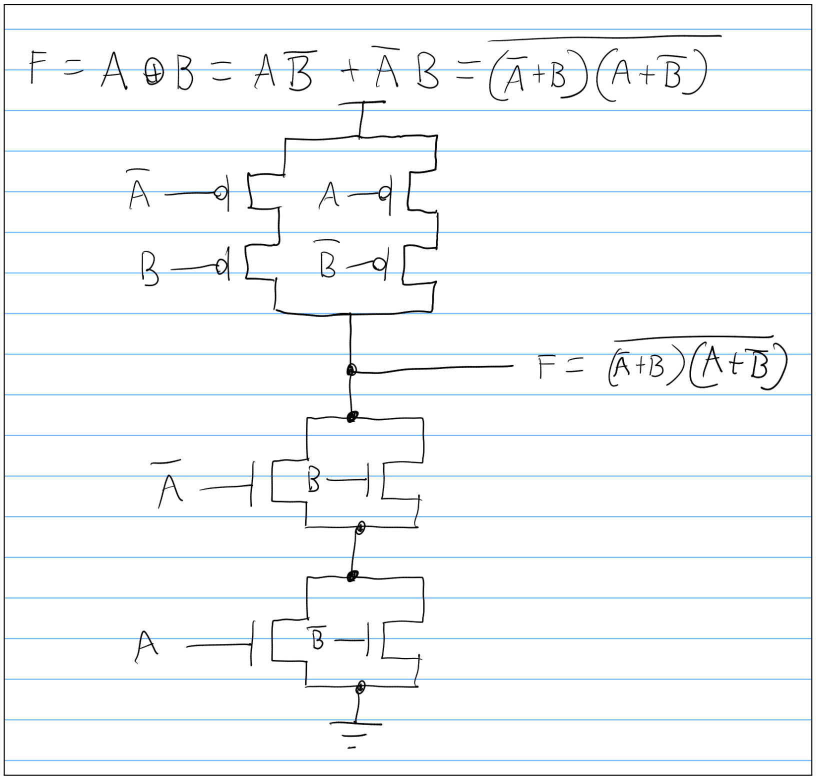

Figure 2: A sketch of the PUN and PDN for implementing the XOR gate.

Since we are only given A and B as inputs and not $\overline{A}$ and $\overline{B}$ as well, we have to invert A and B for the inputs where $\overline{A}$ and $\overline{B}$ are necessary. This requires two inverters at the inputs, as shown in the sketch in Figure 3.

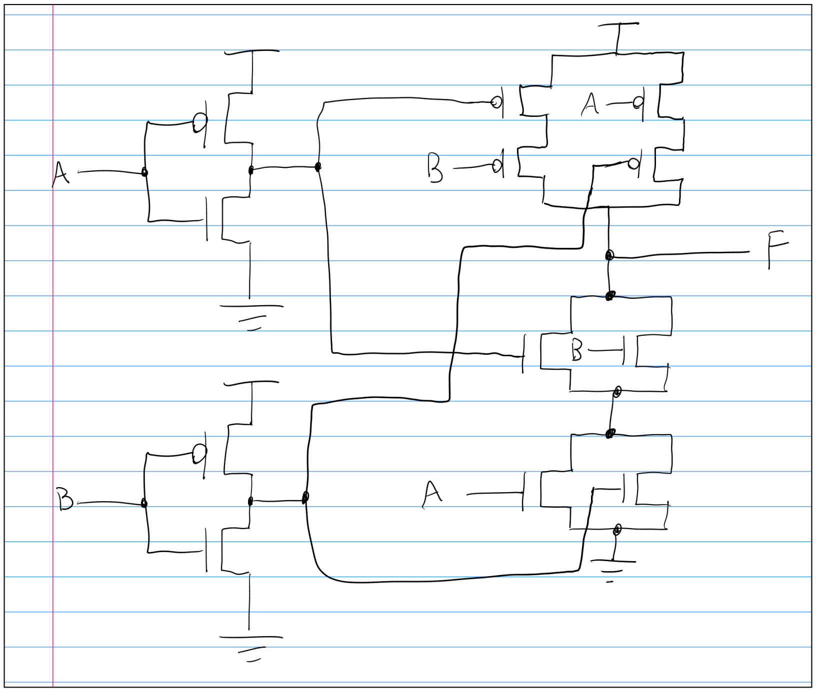

Figure 3: A sketch of the XOR gate in CMOS logic.