**Executive Summary:**

In this project we designing and implementing a complex logic function . The complex function is broken down to gates and further more into transistors and designed in Electric. Euler's path is used to save resources in laying out the transistors. First a schematic is designed in electric and LT spice simulation is run on the schematic to check the waveform. Then the function layout is implemented on Electric and IRSIM is run on the layout to check if the truth table matches up. This ensures that the schematic and the layout of the complex function implement is correct.

**Introduction**

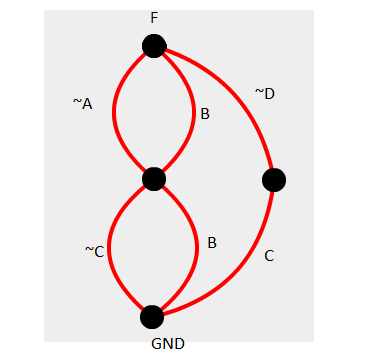
A complex logic function consists of many transistors to for gates and complex interconnections. The function we are implementing in this project is . This complex function requires NAND, NOR and inverter gates to implement. And all in all requires a total 18 transistors. As seen in the Boolean function we are using four NOR gates and two NAND gates. There are three inverter gates used to negate the value of C, A and D.

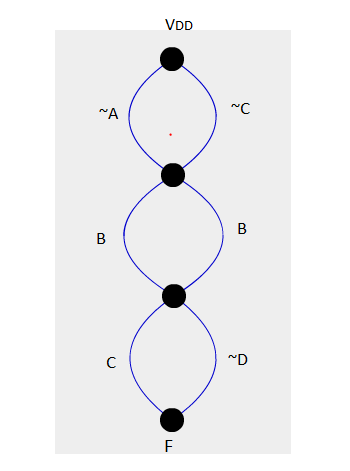
Before we start designing a circuit for this complex logic function, we need to find the optimal method of laying out these transistors, without wasting so much space and resources. We use Euler's path to aid us in setting up the transistors

**Euler's Path**

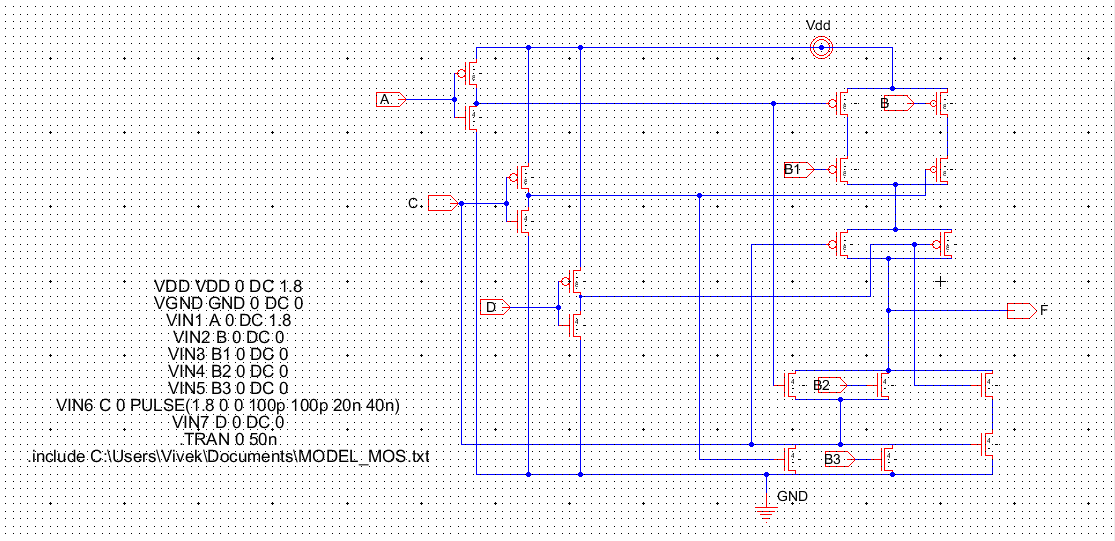
An Euler path is a path that uses every edge of a graph exactly once. An Euler path starts and ends at different vertices. So we have find an Euler's path for the Pull Down Network and Pull Up Network. Pull up network has PMOS transistors only and Pull down network has NMOS transistors only. We not only have to find Euler's path for both networks but also have to find an Euler's path that matches the traversed edges on both networks. The optimal Euler path that I found for this project is **~A-B-C-~D-B-~C**. As shown above the Euler's path traversed for both networks are the same. The Euler path for both network is shown on the figure 1 and 2 below.

**Figure 1: EULER PATH for PULL DOWN NETWORK (PDN)**

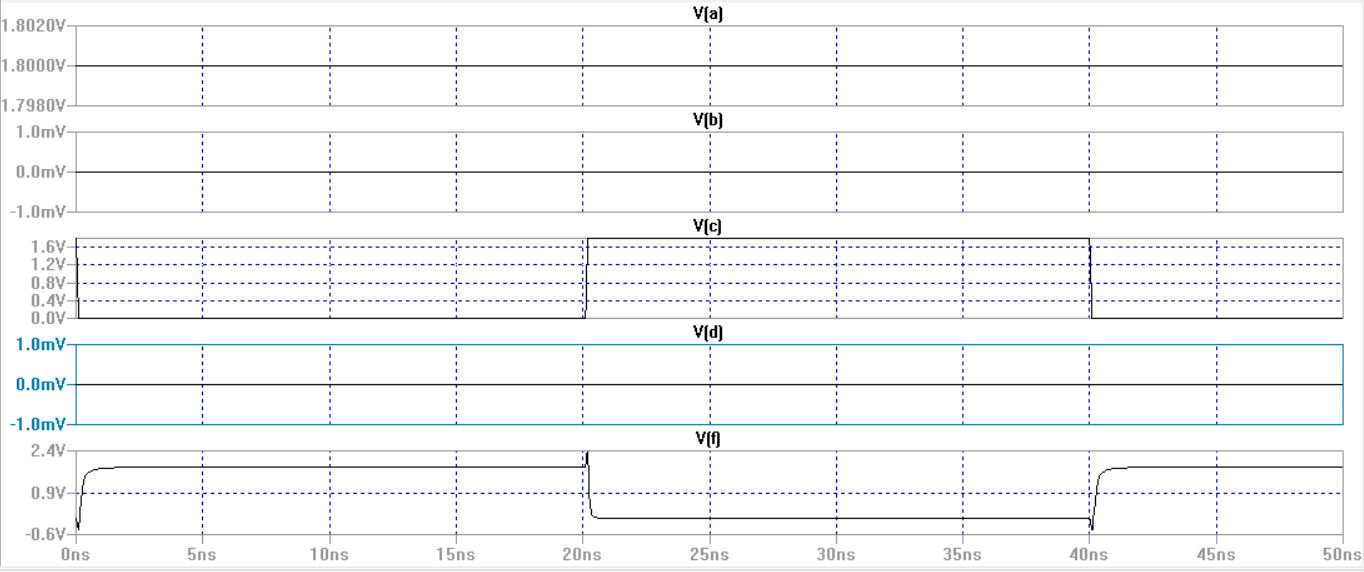
Euler's Path **~A-B-C-~D-B-~C**.

**Figure 2: EULER PATH for PULL DOWN NETWORK (PDN)**  Euler's Path **~A-B-C-~D-B-~C**.

**Figure 3: Schematic of CMOS Complex Logic Function**



**Figure 4: LT spice simulation of the Complex Logic Function Schematic**

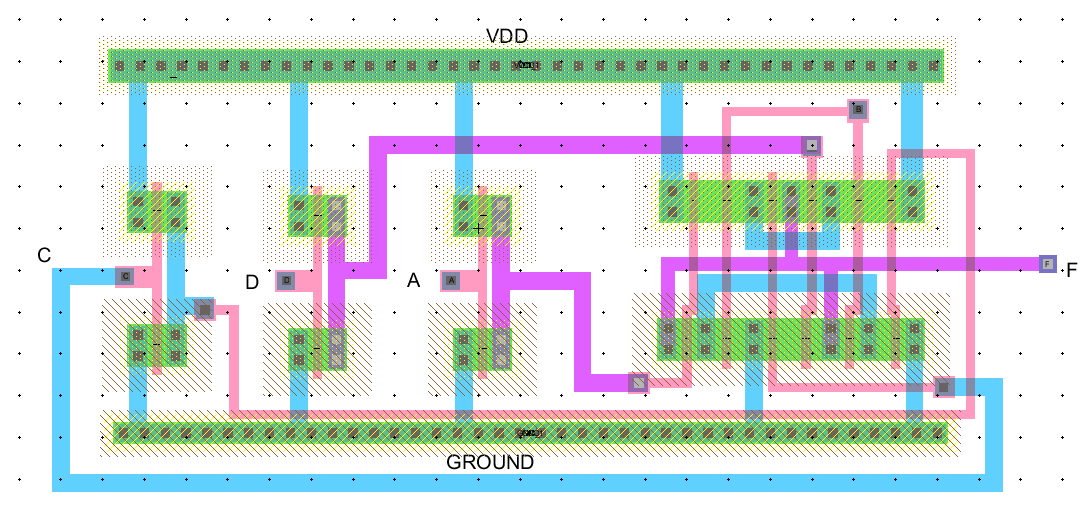


**Truth Table of the Complex Logic Function**

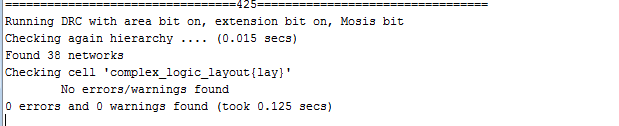
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **output (F)** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| **1** | **0** | **0** | **0** | **1** |
| 1 | 0 | 0 | 1 | 1 |
| **1** | **0** | **1** | **0** | **0** |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

As seen above in highlighted row, the LT spice simulation output verifies the values in truth table and also verifies that the schematic is correct.

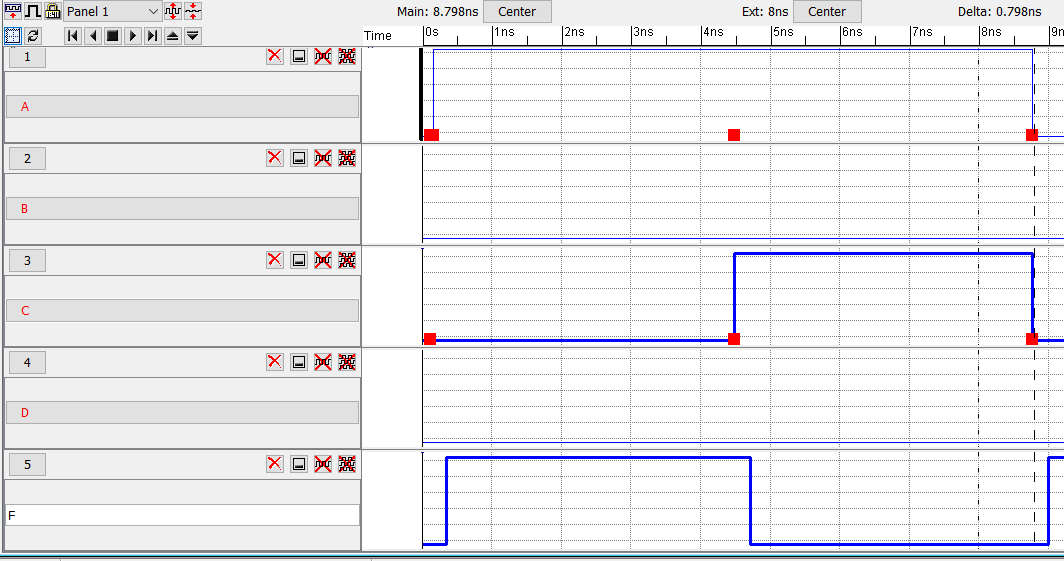
**Figure 5: Layout of CMOS Complex Logic Function**



**Figure 6: DRC and Well Check(This proves that the layout doesn't violate any design rules)**



**Figure 7: IR-SIM of the CMOS Complex Logic Function**



**Truth Table of the Complex Logic Function**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **output (F)** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| **1** | **0** | **0** | **0** | **1** |
| 1 | 0 | 0 | 1 | 1 |
| **1** | **0** | **1** | **0** | **0** |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

**Explanation of Results:**

As seen in the IRSIM graphs, for the values:

* F(A,B,C,D) = (1,0,0,0), F(output) = 1
* F(A,B,C,D) = (1,0,1,0), F(output) = 0

The observation from the IRSIM graph is validated by the truth table of the complex logic function(as seen in the highlighted row). There is a .23nanoseconds delay in the output signal also observed in the graph. This is due to the time taken by the transistors to process the input signal to give a valid output signal.

**Conclusion:**

This project went more in depth about transistors, gates and inverter design. We used Euler's path to help us design a better transistor circuit. Since the number of transistors increase we have to find better ways to design them optimally. Euler's path is a great method that we can use to route the best path to lay down the transistors. One of the things observed in this project is that, as the logic functions get more complicated, there are more and more interconnects between gates. We have to implement multiple layers of metal interconnects to compensate the growing network. We also observed that we need to slow down the frequency of the waveform to observe the results because as the circuit gets more complicated there is more delay in output signal.

**References**

[1] Graph Creator - Illuminations

http://illuminations.nctm.org/Activity.aspx?id=3550

[2] Electric VLSI Tutorials from CMOSedu.com

http://cmosedu.com/videos/electric/electric\_videos.htm

[3] Boolean Algebra - Digital Electronics Course

http://electronics-course.com/boolean-algebra

[4] Spice3 User´s Manual